VASTUDIO –A GENERIC TOOLKIT FOR MULTI-AGENT DEVELOPMENT

Jiewen Luo, Zhongzhi Shi, Fen Lin, Maoguang Wang

Institute of Computing Technology, Chinese Academy of Sciences, 100080 Beijing, China
{luojw, linf, wangmg, shzz}@ics.ict.ac.cn

Abstract: In this paper, we propose a hierarchical agent development toolkit-VASudio. It is composed with legacy component layer, behavior layer, agent layer and society layer. Through the structure, it can facilitate software reuse and rapidly develop multi-agent system. We use it as the main developing toolkit for various multi-agent applications, which include the agent based, distributed data mining and flood decision support system etc.

Key words: Agent, MAGE, AUML, Toolkit, VASudio

1. INTRODUCTION

Intelligent agent technology has been raised as the primary technology in the intelligent Internet and mobile computing. Agent is the rational entity with big-granularity, high-intelligence and autonomy characteristics, and multi-agent system (MAS) is composed of a number of agents capable of communicating, coordinating, cooperating and even competing with each other. Up to now, it has been applied in some industry domains such as transportation control, emergency management. However, there still exist many challenges to construct complex agent-based industry applications because of lacking uniform methods and proper integrated development platform. Most of current platforms focus on the multi-agent run-time environment and neglect the development environment. In order to conquer this problem, we develop a toolkit named VASudio to provide an integrated, user-friendly environment for system developers.

The rest paper is organized as follows. Section 2 introduces the multi-agent development process. Section 3 describes the hierarchical architecture of VASudio. Section 4 demonstrates its GUI and its main functions. Section 5 presents two application developed by VASudio. Section 6 summarizes the paper and gives a conclusion.

2. DEVELOPMENT PROCESS

Our development process is as follows. When facing a problem and thinking it beneficial from the multi-agent paradigm, we first analyze the problem and model it using the AUML [4]. After that, we implement the model through the toolkit VASudio. And then, the implemented agent programs run on the multi-agent environment MAGE to concrete applications. The process is as figure 1. The next we briefly introduce VASudio’s companion systems: AUML and MAGE.

Figure 1. Multi-Agent Development process

2.1 AUML

Multi-agent Systems are complex software. Building such complex systems software requires using adequate engineering methods. In the UML standard published by OMG, the abstract syntax are expressed by a class diagram that represents the relationships between the model elements, and the static semantics are described by constraints in OCL. Diagrams are very clear and easy to understand by the UML users, but they are hard to process by computers. We combine the attributed graph and the Meta representation to formalize the abstract syntax of AUML. This formal abstract syntax and static semantics meets the requirements of AUML supporting tool designing and model checking.

2.2 MAGE

MAGE [1] is a multi-agent run-time environment. From Fig. 4 we can see that it consists of four main components, that is, Agent Management System (AMS), Directory Facilitator (DF), Agent, and Message Transport System (MTS). The AMS maintains a directory of AIDs (agent identifiers), which contain transport addresses (amongst other things) for agents registered in MAGE. The DF provides yellow pages services to other agents. MTS is the default communication method between agents on different agent platforms. Agent is the fundamental actor in MAGE, which combines one or more service capabilities into a unified and integrated execution model that may include access to external software, human users and communications facilities.

Figure.2 Runtime platform-MAGE

3. HIERARCHICAL STRUCTURE OF VASTUDIO

After introduce the development process, this section will describe the hierarchical architecture of the VASudio in detail. From software engineering perspective, the reusable classes are main components of new software and programmers prefer to integrate the existing modules for rapid development. Hence, we adopt a hierarchical design pattern in VASudio development phase. Figure 3 shows VASudio's structure, which is composed with four different layers: the legacy component layer, behavior layer, agent layer and the society layer. The lower layers supply the fundamental materials for upper ones to construct more abstract and complex instances.

Legacy component layer includes connectors and entity components. The entity components are independent existence and can be applied directly. For

example: procedures, functions, DLL and java Beans etc. Connectors are used for guaranteeing connection, which includes pipe, cache, and message transmission etc. We can describe them as follows.

Figure. 2 Legacy component descriptions

According to specification, behaviors are the basic components to build agents. In VASstudio, we apply the template to build basic framework of every behavior, which has one or more actions in the program bodies. Moreover, the base Behavior Class that provides common interfaces for subclasses has been implemented and stored in the template library. It can be conveniently inherited by every child Behavior to implement various applications.

Figure 3. Hierarchical Structure of VASstudio

Although behaviors are components having concrete actions, they cannot execute as independent entities. They are used by autonomous, intelligent and mobile agents. In VASTudio, agents that wrap concrete behaviors can be moved to distributed data sites for independent run and can communicate and collaborate with their peers for complex multi-agent application.

The highest layer of the structure is agent society that provides a platform for controlling multi-agent interaction. In this layer, we can monitor the message transmission, agent collaboration process.

4. VASUTIDO ENVIRONMENT

Figure 4. Graphical user interface of VASTudio

VASTudio is a multi-agent development platform. We summarize its main function and characteristics as follows.

- (1) Software reusable. VASTudio can support software reuse including functions, procedures, classes and java Bean etc.
- (2) ADL Description. VASTudio provides an effective description method and uses the ADL (Agent Description Language) for dynamic deployment.
- (3) Powerful tool sets. It provides many useful tools for users to develop applications, which include code editor, ADL analyzer and ontology editor etc.
- (4) GUI. VASTudio uses the visual Wizard to help builders step by step, which makes users feel easier to develop.
- (5) Java. VASTudio is developed completely by Java language, which has good ability to cross-heterogeneous platforms.

As figure 4 shows, GUI is composed with the five main parts: the visual coding area, basic library, debugs frame, working bench and menu-tool bar. Visual coding area offers an ideal window for java-oriented programmers to implement agent-based applications. Considering some programmers not familiar with agent theory, we provide the Behavior, Agent and Society building Wizard like most business software to help them quickly master multi-agent development process. Working bench record the recent classes and packages. It is composed by three overlap frames, which includes the recent behaviors, agent and society. The menu and tool bar provide the basic support for various manipulations. And the debug frame records the bugs of the recent programs and give suggest how to correct them. One of the most important components is basic library. It stores the basic legacy components which support system builders to rapidly implement their applications. From the figure, we also show the ontology editor, it provides a wizard to visually develop agent ontology, which is much important for multi-agent interaction.

5. APPLICATIONS

5.1 Agent-Based Distributed data mining

Figure. 5 Agent based distributed data mining

Recent years, many researchers are investigating to apply the multi-agent paradigm for distributed data mining. Although many models and techniques are introduced such as meta leaning and wrapping-agent, there is few mature development environments for this kind of application. In order to narrow

the gap of theory and practice, we build the data-mining algorithms and store them in the basic library. These algorithms include classifiers and clusters and some of them are imported from Weka [5]. Through the visual agent development environment, user can easily to wrap the algorithms to agents and dispatch them to remote hosts for distributed data mining.

5.2 Flood decision support system

We use sense network to monitor rivers, which transmit the real-time information to mobile agents. After collecting flood information, the agents sent the message to the central control office where has flood information analysis software wrapped by agents. At the same time, we use the agent spiders to collect former cases and store them in the case bases. Combining with the expert system software OKPS and data mining software MSMiner [2], we can acquire the proper decisions.

Figure 6. Agent-based flood decision support system

6. CONCLUSIONS

In this paper, we describe VASudio, a generic and user-friendly tool for developing multi-agent applications. It has a hierarchical structure and visual programming environment. In order to support rapidly developing various applications, we build a basic library, which includes legacy components.

Though the application, we can see it an idea development environment. In the next step, we will continuously enrich the basic library and provide more useful and mature components for user developing multi-agent applications.

ACKNOWLEDGMENTS

This work is supported by the National High-Program of China (Grant No. 2003AA115220) and the National Basic Research and Development Plan of China (Grant No. 2003CB317000).

REFERENCES

- [1] Zhongzhi Shi, Haijun Zhang, Yong Cheng, Yuncheng Jiang, Qiuqian Sheng, Zhikung Zhao: MAGE: An Agent-Oriented Programming Environment. IEEE ICCI 2004: 250-257
- [2] Zhongzhi Shi, Youping Huang, Qing He, Lida Xu, Shaohui Liu, Liangxi Qin, Ziyang Jia, Jiayou Li. MSMiner-A Developing Platform for OLAP. Decision Support Systems, 2005
- [3] Zhongzhi Shi, Qing He, Ziyang Jia and Jiayou Li. Intelligence Chinese Document Semantic Indexing System. International Journal of Information Technology and Decision Making, Vol.2, No.3, 2003: 407-424.
- [4] Zhikung Zhao, Research on Agent Unified Modeling Language, Ph.D. Thesis, Institute of Computing Technology, CAS, 2003
- [5] <http://www.cs.waikato.ac.nz/ml/weka/>
- [6] Intelligent science website www.intsci.ac.cn