Agent Grid Based Enterprise Application Integration

Maoguang Wang\textsuperscript{1,2,3}, Zhongzhi Shi\textsuperscript{2}, Li Zeng\textsuperscript{2,3} and Fen Lin\textsuperscript{2,3}

\textsuperscript{1}School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221226, China
\textsuperscript{2}Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
\textsuperscript{3}Graduate University of Chinese Academy of Sciences, Beijing 100039, China
\{wangmg,shizz,zengl\inf\}@ics.ict.ac.cn

Abstract

Due to the increasing software complexity, how to integrate the legacy data resources and systems is a big challenge for the enterprise. This paper makes full use of agent technologies to establish an agent grid intelligent platform for enterprise heterogeneous system integration. The agent grid intelligent platform adopts a hierarchical integration model and middleware to provide efficient integration capability for different application systems. Finally, this paper introduces the grid based city emergency interaction system to illustrate the AGrIP practical system integration.

1. Introduction

How to integrate various heterogeneous systems is becoming a big challenge for various enterprises due to the considerable diversity, heterogeneity and geographical distribution. So far, engineers have endeavored to propose various solutions to solve this problem, including CORBA, RMI, Web Services, Grid technologies and so on. Grid technologies are expected to share a large scale of resources and provide service collaboration \cite{1}. During integration process reuse is very important because it improves the reliability and efficiency and decreases software development time and cost.

In the past decade attempts to apply intelligent agents in realizing the grid vision have been made by academic researchers. The most interesting work in the literature might be the Agent Grid concept proposed under the DARPA ISO’s Control of Agent-Based Systems (CoABS) program. As Ian Foster says, the grid and agent communities both develop concepts and mechanisms for open distributed systems, albeit from different perspectives \cite{2}\cite{3}.

Sentil Selliah, Ramana Reddy, \textit{et al}.\cite{4} present a framework called Eksarva which supports the formal modeling of collaboration as a function of behavior and workflow rules. They use this modeling approach to program agents that execute the integrated activities of knowledge management necessary for efficient group collaboration.

Martin Bichler, Arie Segev, \textit{et al}.\cite{5} discuss the future trends to use intelligent agents as the gluing components within a component framework. Munehiro Fukuda, Koichi Kashiwagi, \textit{et al}.\cite{6} propose a grid-computing middleware system AgentTeamwork which dispatches a collection of mobile agents to coordinate a user job over remote computers in a decentralized manner.

We looked agent grid as a number of interacting agent-based autonomic nodes, which adjusts its behaviors automatically and collaborates with each other to provide efficient intelligent application services. Our research goal is to integrate the systems based on agent grid technologies. Furthermore, we developed an agent grid intelligent platform called AGrIP to test the integration solution.

The rest paper is organized as follows: Section 2 describes the hierarchical integration model and middleware. Section 3 introduces agent grid intelligent platform AGrIP and its application in GEIS(Grid-based Emergency Interaction System). Section 4 brings a summary of this paper.

2. Integration method

2.1. Hierarchical integration model

The software integration based on agent is a clear hierarchical structure. We adopt the integration model by encapsulating the legacy application to agent directly or reuse software components as behaviors. The generating behaviors can be added to the agent. Furthermore, the relevant encapsulated agents construct an agent society \cite{7}. Figure 1 illustrates the hierarchical integration model.
Figure 1. Agent based integration model

The bottom layer provides connector and reusable plenty of algorithm packages for generating behaviors. The behaviors describe the agent capabilities which model the agent running task. The agents are physical autonomous entities located in the respective node. And the related agents become a virtual agent society, which can collaborate and communicate to resolve the appointed complex tasks.

The multi-agent platform MAGE plays an underlying role in the system integration because the multi-agent system is responsible for the managing and integrating diverse agents. Figure 2 illustrates MAGE platform.

Figure 2. MAGE platform architecture

MAGE has core modules as follows:

AMS (Agent management system) is a mandatory component of MAGE. It maintains a directory of Agent Identifiers, which contains transport addresses for agents registered in MAGE and offers white pages services to other agents.

DF (Directory facilitator) is an indispensable component, which provides yellow page services to other agents. Agents may register their services with the DF or query the DF to look up which services are offered by other agents.

MTS (Message Transport System) is the default communication mechanism between agents on different FIPA-Compliant agent platforms. It uses FIPA ACL as the standard communication language.

Compared with the distributed object technologies such as DCOM, CORBA and RMI so on, agent based hierarchical integration model does not need the programmers to master the technology details because MAGE provides various templates to design agents. Also it offers the user with behavior library, agent library for reusability.

Furthermore, MAGE support the protocols including RMI, SOAP, HTTP, etc. Thus, the user can build the distributed application effectively.

Software reusability is the key problem for integration. MAGE supports three basic kinds of reusability.

Embedding: MAGE can call the corresponding methods directly from external objects by inheriting the classes or creating instances of the classes.

Executable program: MAGE is capable of intercepting the standard I/O stream and executes the programs directly.

DLL: MAGE preloads the related .dll files into the memory and calls the corresponding methods.

The three reusability methods provide more flexibility for integrating legacy applications.

Moreover, MAGE provides additional mobility capability once the agent is deployed on MAGE platform. When the agent moves from one node to another, MAGE platform will decide whether it transfers the codes or states to the destination. When the agent moves to the destination, MAGE will check the transformation state through the agent ID and attributes. After that, MAGE will just transfer the agent states if the destination has the needed agent. Otherwise it will oblige the agent to transfer all of the codes and states. After the agent moves or clones itself to another node, it will operate automatically according to its roles and rights. Once the agent obtains the satisfactory result, it will feed back them to the source program. Through the definition of the priorities, time and location, the agent can restrict itself to run on the correct node at the specified time. Therefore, by encapsulating the legacy systems to agent it not only reuses the applications efficiently but also provides additional agent capability to the systems.

2.2. Integration middleware

AGrIP is developed by our laboratory on the basis of MAGE. Our research goal[9] is to make full use of agent and grid technologies to build an agent grid environment for the industry applications. Considering the software complexity and heterogeneity, we have been focusing on building a middleware for complex system integration. The middleware is divided into two parts logically: ontology components middleware and application software middleware.
To disambiguate the semantics, ontology technologies are used to eliminate the misunderstanding of the information. AGrIP builds ontology components middleware to enable message mutual understanding for communication and collaboration.

Likewise, the ontology middleware also plays an important role in the AGrIP technical solution. It supports well-defined application programming interfaces used for accessing knowledge, deals with such matters as meta-information and ontology lookup, as well as concepts and properties definition. It provides a general mechanism for message interaction and knowledge management such as KMSphere [8]. Thus, after the application systems are encapsulated to agents they can collaborate and exchange the message based on ontology. For instance, we integrate a GIS system into the AGrIP middleware layer by inheriting the core class and adding the communicating behaviors successfully.

Moreover, web services are fast emerging as the dominant means for connecting remotely executing programs via well established internet protocols and commonly used machine readable representations. Software agents are now increasingly used in commercial applications to solve complex engineering problems, and these applications often expose or make use of web services. Hence, we build the gateway architecture for connecting software agents and web services in a transparent manner with fully automatic operation. This gateway allows web services to invoke agent services and vice versa by translating message encodings and service descriptions between the two technologies. Thus AGrIP can access application systems via a number of protocols: HTTP, RMI, CORBA, and SOAP.

AGrIP provides the abstract level to separate the complex integrations with the fundamental facilities. Once the legacy systems are integrated to AGrIP, the related agents construct a virtual agent society, which can collaborate to resolve the appointed complex tasks. The agent societies can further construct a complex agent grid.

When integrating the legacy systems, AGrIP can deploy some MAGE servers to construct a complex topology. This structure inherits the advantages of dynamic grow-up, manageability, maintainability and high efficiency. AGrIP provides simple, fast and agent-based integration. Also it provides a perfect development pattern to help client users encapsulate the legacy systems to agent services. Compared with the traditional application integration, AGrIP supports the data transformation and application integration. Ultimately, the external applications, such as e-government, e-commerce, information retrieval etc., can obtain the needed knowledge via the middleware levels.

3. AGrIP platform

3.1. AGrIP architecture

AGrIP is an agent grid intelligent platform for the agent grid development and deployment. It adopts the agent-oriented software engineering methods to develop complex systems. It uses AUMP to model the system, VAStudio to support the hierarchical system development, and use MAGE to deploy the developed system. It adopts the hierarchical integration pattern mentioned above to encapsulate the legacy systems and reusable components. Figure 3 shows the AGrIP architecture.

AGrIP is an agent grid intelligent platform for the agent grid development and deployment. It adopts the agent-oriented software engineering methods to develop complex systems. It uses AUMP to model the system, VAStudio to support the hierarchical system development, and use MAGE to deploy the developed system. It adopts the hierarchical integration pattern mentioned above to encapsulate the legacy systems and reusable components. Figure 3 shows the AGrIP architecture.

Common resources: it consists of various resources distributed in grid environment, such as databases, various services, web resources, or others, which run on Unix, NT and other operating systems.

Multi-Agent Environment MAGE: it is the kernel of agent grid computing which is responsible for directory service, integration service, security service, policy, management, communication and so on.

Middleware including ontology and developing toolkits: ontology middleware supports resource sharing and knowledge management. Developing software toolkits provide development environment including collaborative toolkits, information retrieval, distributed data mining and the like, to let users effectively use grid resources.

Application service: it organizes certain agents automatically for specific application purposes, such as e-science, e-business, decision support and bio-information.

Figure 3. Agent grid intelligent platform AGrIP
The integration can be looked as an evolutionary process. The agent grid intelligent platform supports the agent based integration, and supports complex collaboration and self-management. AGrIP support requirement analysis and design including role model, goal model and agent library and function component library for reusability. AGrIP makes use of software toolkits to integrate external applications, thus becoming a robust, friendly-used and stable integration platform. The policy plug-ins support various templates design and support policy-driven self-management. Besides, AGrIP inherits the characteristics of agent and has the capability of collaboration, coordination and reasoning etc. The users can expand the agent capability, adding new functionality and policy, and then add them to AGrIP. Therefore, AGrIP is a user friendly fundamental integration platform.

Once the legacy systems are integrated to AGrIP, the related agents construct a virtual agent society, which can collaborate to resolve the appointed tasks according to their roles and policies. The agent societies can further construct a complex agent grid. Figure 4 illustrates the agent grid based integration.

AGrIP can deploy some MAGE servers to construct a complex topology. This integration inherits the advantages of dynamic grow-up, manageability, maintainability and high efficiency. AGrIP provides simple, fast and agent-based integration. Also it provides an effective integration pattern to help client users encapsulate the legacy systems to agent services. Compared with the traditional application integration, AGrIP supports the data transformation and application integration efficiently.

3.2. AGrIP application

AGrIP has been applied successfully in a complex, dynamic, multi-agent domain, such as GEIS(Grid-based Emergency Interactive System)[8]. GEIS is expected to integrate various resources and systems from different departments including government, police, traffic, sanitation and so on. It achieved the goals of integrating resources for further group decisions and value-added services. Figure 5 illustrates the GEIS framework.

GEIS based on AGrIP comprises the platforms: MAGE fundamental integration platform, GIS for geography location, Decision Support System OKPS, Knowledge management platform KMSphere and so on[7]. The related departments such as police, constructs a police agent society. We deployed agents on various departments and collect information by monitoring and analyzing the real-time information. Once the emergency accident occurs, the agent will feed back the information timely. Then the DSS tool will query the history databases based on Case-Based Reasoning methods and make decisions for scheduling resources. Finally it deploys the solution on the GIS platform.

To sum up, AGrIP proved to have the following characteristics:

Flexible. AGrIP provides easy and fast heterogeneous system integration. The enterprise application integration framework allow the policy engine to configure itself dynamically. We can change a policy configure file to mix and match different application components.

Open. AGrIP supports the integration for popular web service applications, business intelligence and external applications seamlessly.

Distributed. It integrates heterogeneous data sources and deals with different documents stored in the distributed environment.

Self-adaptive. It adapts itself to the front users needs according to the specified preferences. AGrIP shows
the relevant functionalities when the authorized customers use it.

Manageable. It provides an easy-used and universal console to manage the performance, metadata, ontology library and so on.

AGrIP is capable of encapsulating database system, application system and legacy system components. It supports grid node collaboration to the integrated systems. Every grid node can monitor the network resource and local hardware utility including CPU, memory, diskette and network resource to make decisions.

The GEIS system integrates the heterogeneous systems including the Government Control Center that provides the service security and emergency decision supports, Anti-terror and Criminal System that charges for the anti-terror and criminal events, Fire control system, and traffic security system so on. It supports the call center, e-mail, telephone, and fax etc. to access the system.

4. Conclusions

Based on the hierarchical integration model and middleware, AGrIP is capable of integrating the existing legacy systems. It provides a promising approach to some of the challenges in agent grid initiative to meet the needs of sharing resource, system integration, and service collaboration. It has been applied into the industry projects that proved it scalable and efficient.

Acknowledgment

This work is supported by the National Science Foundation of China (No. 60435010, 90604017, 60675010), 863 National High-Tech Program (No.2006AA01Z128), National Basic Research Priorities Programme (No. 2003CB317004) and the Nature Science Foundation of Beijing (No. 4052025).

5. References