

MESSAGE SERVICE SYSTEM FOR INTEGRATION PLATFORM BASED ON THE DISTRIBUTED AGENT TECHNOLOGY

Bao Zhenning, Fan Yushun, Xue Sixin

Department of Automation, Tsinghua University, Beijing, P.R. China

Key Words: Distributed Agent Technology, Message Service System, Integration Platform

ABSTRACT

Message Service System (MSS) is a part of CIMS application integration platform (IP), which help users connect with each other directly. Based on distributed agent technology, direct user-oriented messaging service is realized. The integration of platform messaging system and the Internet mail system is also completed. In this paper, the messaging mechanism is introduced, the system architecture, system functions and the implementation methods are presented in detail.

1.BACKGROUND

CIMS application integration platform (IP) is a software platform, which supports development, integration of CIMS applications and system operations. It provides transparent and consistent information accesses and interaction methods under heterogeneous distributed environments. It also manages CIMS applications, and supports their integration^[1]. On integration platform, it is a frequently used function for transferring managing information between users from different departments. So an effective means of contact among users is necessary. Besides, as a basic part of process integration, workflow model needs a messaging method to realize the communication between its modules in the run-time. In order to support the information sharing of users as well as the collaboration of applications, a Message Service is implemented that serves as a basic service in 'Operation Administration sub-platform' (OAP) of CIMS application IP.

2.MESSAGING MECHANISM OF THE IP BASED ON DISTRIBUTED AGENTS

Message Service System (MSS) is a module of OAP built on Communication Services (CS) and Global Information System (GIS) of the IP. It provides uniform management, monitor and statistic for different areas and resources, such as users, data, applications and computer hardware resources integrated in the IP. In order to cope with the problems caused by the heterogeneous distributed environment, advanced software agent technology is used to realize the MSS.

Software agent is a software entity, which can realize functions continuously and autonomously in specific circumstance, and contact with associated agents and process^[2]. It is a new technology formed on a multi discipline basis, such as software engineering, artificial intelligence, and communication technologies. Through agents, Programs can provide intelligence to some extent. That is, they can forecast, adapt and find approaches to support user's demands. So it can effectively support the operation of complex and distributed system. Distributed agent system, composed by a set of autonomous and coordinated agents, can provide much more intelligence and flexibility to large-scale systems under heterogeneous distributed environment.

2.1 Distributed Agents Model of OAP

Distributed agents on OAP map the inter-operation between applications into collaboration of agents. It provides unified interactions between agents instead of previously developing many one-to-one interfaces between different applications. Fig.1 and Fig.2 [3] describe the system model and agent unit model of the system. It can be seen that distributed agents realize the unification of loose-coupled system structure and close-coupled unit logic.

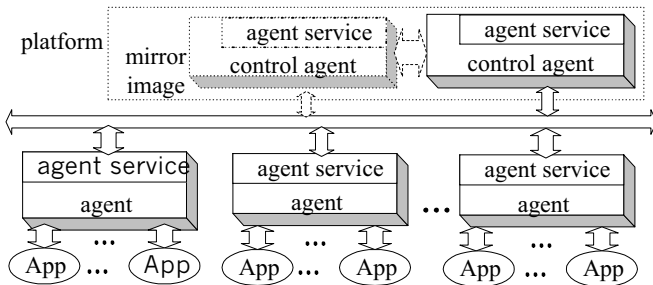
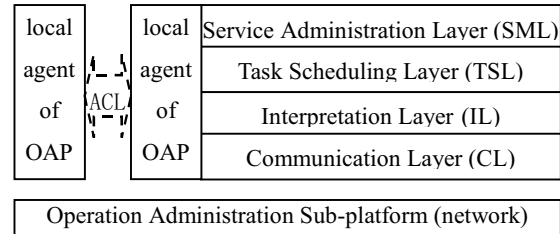


Fig.1. System model of OAP



ACL: Agent Communication Language

Fig.2. Agent Unit Model of OAP

The agents are equal to each other in static functions. Each of them can provide corresponding down-level communication according to user's request. When the request of 'demand-serve' happens, the related agents are linked in C/S structure dynamically. The control agent is used to manage the global information so as to ensure necessary consistency.

Internal agent is layered. The independence between different layers conduces to good openness. Lowest CL (referring to Fig.2) is the interface to exchange data between agent and external. IL interprets requests from CL, which come from local applications or remote agents, and transfers the result to upper layer. TSL is core part for realizing intelligence of an agent. It analyzes the data from TSL according to its own 'knowledge', and directs SML to provide right service.

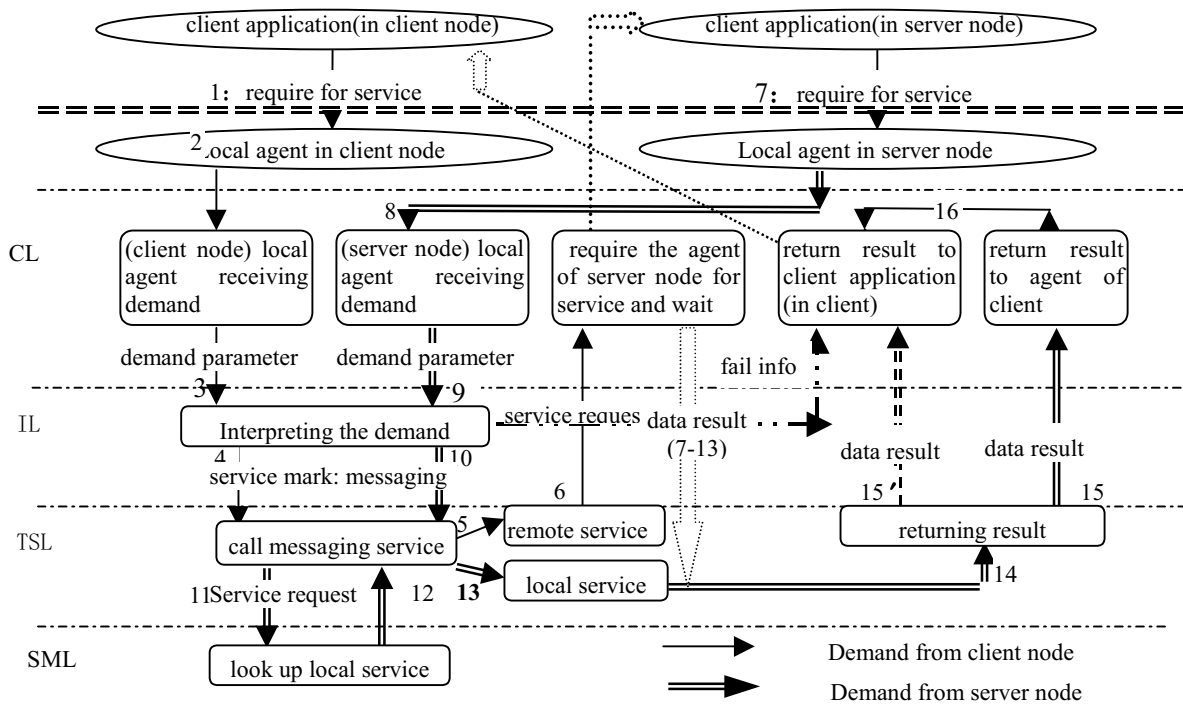


Fig. 3 Internal Communication of MSS

2.2 Messaging Mechanism of MSS

MSS realizes its inner communication by applying event-driven mechanism. After sending out request, the client keeps waiting until result returns. Distributed agents can be considered as middle stages between client and server. Firstly, local agent receives client's request, then it checks the 'local ability list'. If the current node can provide messaging service, the local agent will activate the service. Otherwise, it will inquire the control agent about the node address of the remote server. Then it sends request to the agent of the server. For an agent, the request from remote agent is the same as that from local application. So the server provides corresponding service analogously. Fig.3 illuminates the communication process (the control agent is omitted) made by agents between the client and server (suppose the two are in different location).

3.MESSAGING SERVICE SYSTEM OF IP

3.1 System Structure and Functions

The basic task of MSS is the messaging between IP users and related functions. The both sides involved are inside the IP. By applying C/S structure and distributed agents, it is viable to accomplish low-level communication between application and server, or different IP nodes.

It is very useful to integrate messaging system of the IP with public Internet mail system, then the users on the IP can easily connect with Internet users. This direct contact enterprise also broads the range of information sharing. Therefore it is significative to add an interface to public mail system to MSS. For the senders and receivers of public mail outside IP, the communication mechanism is different: user-oriented client application is still the same as that of internal messaging; but public mail server here acts as the server program, and the down-level communication is prescribed by common network protocol. Fig. 4 shows the structure of MSS.

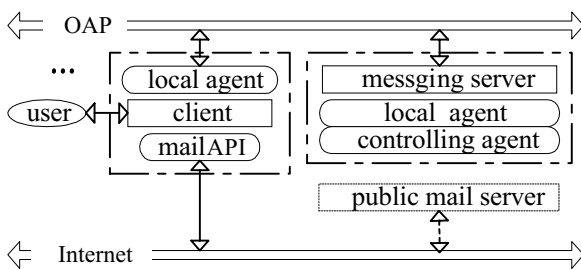


Fig. 4 System Structure of MSS

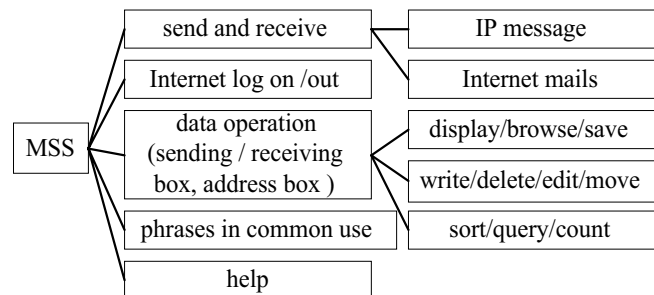


Fig. 5 MSS Functions

The main functions of MSS are showed in Fig.5. It has features as follows:

- Real time: This is most important, for the operation message has close connection with time.
- Safety: Strict authority management is absolutely necessary for the safety of the enterprise.
- Agility: e.g., users can send message to one user, user group or their combination, or broadcast.
- Easy manipulation: The user interface is very amicable.

3.2 Key Implementation Method

There is only one Messaging Server (MSVR) on OAP, installed at the same node as the control agent (because MSVR needs information of users continually). It receives and responds to user request, e.g. sending/receiving message, handling data, etc. It stores users' data congruously, keeping the data independent to data repository. Users' data is stored (encrypted automatically) as

files of a defined format under a certain directory in MSVR. So read-write of data is quite fast, and data is independent to GIS and database. When message is sending, MSVR receives it firstly, then add it to the 'receiving boxes' of all receivers. But MSVR does not realize public mailing function.

There is another server program in each nodes on OAP—Message Telling (MT). In fact, it can be considered as the extension of MSVR in client. After the MSVR receives the message, MT of every receiver who had logged on is started up, telling user to receive new message. Furthermore, it can be started up by the 'checking user's message' module of MSVR, and the module can be activated when starting up local agent if the current user has message unread. The users can receive message as early as possible thanks to the utilization of MT.

Client program provides user interface of all user-oriented functions of MSS, and submits user request. The interface is designed using standard Windows style.

Client gets user's data from the server just after start-up and save it locally. Before closing, it updates the data in the MSVR with new data. In this way, most of the manipulation of data will be localized, so as to improve operation efficiency and alleviate the burden of network.

The client also provides the interface to public mail system. According to current application demands, it is enough to accomplish public mailing function by using some existing API. We choose Common Message Calls (CMC). It can support various messaging systems (e.g. Exchange for Windows95 is a typical CMC application). CMC API can connect to the mail server and account that user has appointed, and then download the received mails to the user's mail box established by CMC or send user's sending mail to that mail server. Its user interface is the same as that of IP messaging, the internal difference is hidden by program. The sending boxes and address books of public mail are maintained as IP data. However, the received mails are stored in IP file form in local node, considering most of them have nothing to do with the IP. In this way, the same data operation can be kept, and the burden of MSVR is greatly alleviated as well. In fact, after receiving a mail, the client removes it from the CMC box and saves it under the IP directory. The 'server' is the common mail server assigned by user, which does not need extra development.

4.CONCLUSION

The MSS is the unified and unique messaging system implement of OAP. It provides direct connect between IP and Internet users. It supports the collaboration of workflow applications also. Distributed agents constitute the foundation of the MSS. They connect the MSS server and client. Agents, client and server cooperate with each other, accomplish whole functions related with messaging. Now the first version of MSS product has entered the market. Of course it is inevitable that there is still many problem to solve. Further research will improve performance of it.

REFERENCES

1. Fan Yushun, Wu Cheng, Yu Panxiang, CIMS Application Integration Platform For Manufacturing Enterprises, Journal of Tsinghua University (Science and Technology), Vol.38, No.3, 1998
2. Jeffrey M. Brandshaw, Software Agents, First edition, New York: The MIT Press,1997
3. Cao Junwei, Fan Yushun, The Research and Design of Operation Administration Agents, Computer Integrated Manufacturing System, Vol.5, No.3, Jun.,1999