行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※ ※
※ 電子商務之程序程式之研究 ※
※ ※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：☑個別型計畫　　□整合型計畫
計畫編號：NSC 89－2213－E－009－022
執行期間：88 年 8 月 1 日至 89 年 7 月 31 日

計畫主持人：陳振炎 教授
共同主持人：

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：交通大學資訊工程系

中　華　民　國　89 年　9 月　6 日

行政院國家科學委員會專題研究計畫成果報告
電子商務之程序程式之研究
Process Programming in Electronic Commerce
計畫編號：NSC 89-2213-E-009-022
執行期限：88 年 8 月 1 日至 89 年 7 月 31 日
主持人：陳振炎 教授　交通大學資訊工程系

1. Abstract

Electronic commerce process traditionally is enacted by centralized work flow management system which is inflexible and unreliable. Contrarily, the research presents a decentralized architecture composed of collaborating agents to solve the problem. Also, mobile agent is used to reduce communication burden on the web.

Moreover, both process modeling language and agent communication message are modeled by extensible markup language (XML) which is regarded as the universal format on the web. This greatly improves interoperability which is very important, as E-commerce process normally involves multiple systems with different process modeling languages.

This process environment is currently implemented in JAVA, and is based on the IBM aglets software development kit (ASDK).

Keywords: Agent-based System, electronic comer process

中文摘要

傳統上電子商務程序是集中式管理，這樣不靈活且不可靠，本研究提出代理人結合之分散式管理環境來克服之，而且，使用行動代理人來降低網路交通負擔。

進一步，程序模塑語言及代理人溝通訊息皆用 XML **描述，可提昇異系統之互通性，電子商務常需此事。**

本環境用 JAVA **製作，使用** IBM **之代理人程式集（**Aglet Software Development Kit**）。**

關鍵詞：代理人系統，電子商務程序

2. Objective

Business process, especially electronic commerce process, is very dynamic and unpredictable, making it difficult to define its activities and work flow (process). While work flow management system (WFMS) [Alo97b] [Dii95] [Sema] [Wod96] provides definition, generation, management, and enactment of work flow, WFMS can thus be used in e-commerce process. However, WFMS lacks the capabilities of evolution meaning that changing flow during enactment, and scalability meaning that new resources can be easily included. On the other hand, agent [Hay99] is a high level abstraction with intelligence to automatically execute an activity without explicitly specifying control flow. Furthermore, multiple agents can collaborate to perform a task. Therefore, using agent seems to be a good choice in enacting e-commerce process.

One of the weakness of current WFMS is its lack of interoperability due to incompatibility between underlying process modeling language (PML) and data. To solve that, we use extensible markup language (XML) [XML] to model both PML and data. XML provides universal format for data on the web. Also, XML can be extended to define a markup language for a given domain. Initially, XML was used to enhance HTML (hypertext markup language) to better describe data. Later, it is used in areas beyond that, such as e-commerce.

This agent-based process environment (APE) uses agents to exchange XML-based messages to enact e-commerce process. The process is defined by an XML-based process modeling language.

3. Agents

There are currently two kinds of agents in APE 1) client representation agent and 2) process agent.

Client Representation Agent (CRA) is a stationary agent. It assumes two roles, client representative and service manager. As client representative, CRA communicates with process agent (PA) to request PA to execute XML-based process program for client, or to accept and manage the work item assigned by PA. In addition, it provides editor to edit process program, and monitoring tool to watch process enactment. It also possesses intelligence to relieve client from handling some routines. For example, CRA analyzes the content of the work assigned to locate the tools needed, and then automatically invoke the tools at the right time. As service manager, CRA executes the work assigned by PA through the external application invocation system, and reports the result of execution. The service refers to an executable file or a remote object providing

CORBA services.

Process agent (PA) is a mobile agent which enact process. Its functionality includes 1) process enactment, 2) process management and 3) process evolution support. 1) PA gets the XML-based process program from CRA, parses the process program and transforms it to internal process model, and then interprets the model to enact activities. 2) process management provides functions to control enacting process, such as execute, pause, stop, or resume process. 3) PA supports process evolution intelligence which will be covered elsewhere.

APE communicates with PA through control message. When PA receives a control message, it will decide what to do based on the type of message:

1) Process enactment message: This message requests PA to execute process program. The message contains process program and input data.
2) Process migration message: This message instructs PA to migrate to designated place to continue enacting process. The message contains URL of the place.

4. XML-based Process Modeling Language

Process modeling language (PML) is used to define the work (activity) and the work flow (process). APE adopts XML-based PML. Its benefits are:

1)simple, clear, and easy to understand.
XML uses tags to describe data. By reading tags, it is very easy for others to understand the meaning of data. Also, agents can precisely enact the process by following the tags.
2)interoperability.
When WFMS enacts a process, it may need to interact with another WFMS such as requesting it to enact a sub-process. However, if the two WFMS uses incompatible PML, the two WFMS cannot communicate with each other. The nice thing about using XML is that one PML can be easily translated to another through extensible style sheet language (XSL).

The XML-based process modeling language (XPML) in APE models a process in four basic components, service, data, activity, and transition. The entire process is modeled by process template.

Example 1 shows the structure of XPML. <ProcessDefinition> tag is root element, consisting of four components just mentioned, service, data, activity, and transition.

```
<ProcessDefinition Name = "TestCase2"
EntryActivity="CheckCredit">
    <OrganizationList>
      <Organization ... > ... </Organization>
    </OrganizationList>
    <ServiceList>
      <Service ... > ... </Service >
    </ServiceList>
    <DataList>
      <Data ... > ... </Data >
    </DataList>
    <ActivityList>
      <Activity ... > ... </Activity>
    </ActivityList>
    <TransitionList>
      <Transition ... > ... </Transition >
    </TransitionList>
  </ProcessDefinition>
```

Example1：XPML Structure

5. XML-based Agent Communication Message

APE agents communicate with each other through agent communication messages. Currently, the messages consists of data synchronization message and evolution message. The format of the messages are represented by extensible markup language (XML).

A generic agent communication message (see Example 2) is represented by <Message> tag. <Message> tag contains two attributes **Type** and **Sender**. **Type** designates type of message. **Sender** records agent id that sends the message. The content of message is stored in <Content>. Its format depends on message type.

```
<Message Type = “Message Type” Sender =
“sender id”>
    <Content>
      …
    </Content>
  </Message>
```

Example 2：A generic agent communication message

6. APE Architecture

The APE architecture provides services and agent communication framework. Its modules are described below:

- **Process model** is a high level abstraction of process program. It models PML constructs such as activity and transition. Process management sub-system operates on the process model.

- **Communication management.** Like human, agents needs to communicate continuously with other agents to accomplish a task. Communication management subsystem

provides a mechanism for agents to send and receive agent communication messages. Its implementation is based on Java Messaging System (JMS). JMS provides a uniform messaging API which drives messaging systems by different vendors.

- **Parser** parse the process program to produce process model. Its implementation is based on the XML parser by Xerces. First, input process program is parsed to produce DOM tree. Then, by using Element Parser, ServiceList, DataList, ActivityList and TransitionList are extracted from DOM tree.

- **Process Management** is the kernel module to execute and manage process. It handles process management service and agent communication message. Details will be given shortly.

- **Agent** implements APE agent, including client representation agent (CRA) and process agent (PA) that enacts process. The implementation of CRA and PA is based on IBM Aglets Software Development Kit (ASDK) [Lan98].

The current APE implementation consists of about 4400 lines of JAVA code in 54 JAVA classes. Among them, 15 in Process Management subsystem which is the system kernel; 6 in agent subsystem; 9 in Process Model subsystem; 11 in Parser; 6 in Communication Management subsystem; and 7 in utility and user interface.

7. Conclusions

This paper presents an Agent-based Process Environment (APE). APE uses agents to implement decentralized work flow management system (WFMS). Mobile agents enacts XML-based process program. Agents communicate through XML-based agent communication messages to coordinate process enactment. This APE appears to have the following benefits:

1) Reduced communication overhead.
   Since mobile agent carries process program and data to designated place to execute activity, data transfer between server and agent is no longer needed. The communication overhead is thus reduced, especially when the amount of data is large.

2) Increased Reliability.
   Process enactment is shared by multiple nodes on the net, instead of one single node as in a centralized work flow management system. Therefore, if one node is down, it will not affect enactment of other process instances. The system reliability is thus increased.

3) Increased Interoperability.
   Using XML as the formats for both process modeling language and agent communication message enhance system flexibility and interoperability. By using extensible style sheet language (XSL), the process program can be easily translated to the process program of another system. The XML-based message not only eases defining new messages, but also provides a common format among agents. Thus, the system interoperability is increased.

References

[Alo97b] G. Alonso, C. Hagen, H.-J. Schek, and M. Tresch. Towards a platform for distributed application development. In A. Dogac et al.,editor, Advances in Workflow Management Systems and Interoperability, NATO Advanced Study Institute, pages 195 – 221. NATO, Istambul, Turkey, August 1997. To be published by Springer-Verlag.

[Dii95] Diimitrios Georgakopoulos, Mark Hornick, Amit Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infratstructre. Distributed and Parallel Databases, 3, 11-153, 1995.

[Hay99] Caroline C. Hayers. Agents in a Nutshell – A Very Brief Introduction. IEEE Transactions on Knowledge and Data Engineering, vol. 11, no. 1, January/February 1999.

[Lan98] Danny B. Lange, Mitsuru Oshima. Programming and Deploying Java™ Mobile Agents with Aglets™. Addison Wesley Longman, Inc. 1998.

[Sema] Sema Group, Madrid, Spain. http://www.sema.es/projects/WDIE.

[Wod96] D. Wodtke, J. Weissenfels, G. Weikum, and A. K. Dittrich. The MENTOR project: Steps towards enterprise-wide workflow management. In Proceedings of IEEE International Conference on Data Engineering, New Orleans, Louisiana, 1996.

[XML] Extensible Markup Language. http://www.w3.org/XML/.