Workshop on Artificial Intelligence

Business Processes Management System based on Mobil Agent Technology *

Jong-Yih Kuo

Department of Computer Science and Information Engineering

Fu Jen Catholic University

510 Chung Cheng Rd., HsinChuang, Taipei 242, Taiwan

jykuo@csie.fju.edu.tw, Tel: 2-29031111-2444, Fax: 2-29023550

Abstract

Due to the development of Internet and the desire of almost all departments of business organizations to be interconnected and to make data accessible at any time and any place, more and more workflow management systems are applied to business process management. In this paper, a mobile, intelligent and document-driven agent framework is proposed to model business process management system. Each mobile agent encapsulates a single document which includes a set of business logic. It can achieve (1) trace ability: a function that enables administrators to monitor document processes easily, (2) document life cycle: a feature using agent life cycle to manage document life cycle and concurrent processing, and (3) dynamic scheduling: a document agent can dynamically schedule its itinerary, and a document control agent can dynamically schedule its services. We also implemented an official document management system explaining our approach by Aglet.

Keywords: Mobile Agent, Business Process Management, Document-driven Approach

*This research is partially supported by the Societas Verrbi Divini and the National Science Council under grant NSC 90-2213-E-030-014.

1 Introduction

Workflow management promises a new solution to a traditional problem: controlling, monitoring, optimizing and supporting business processes. Traditional business process management system utilizes distributed objects management technology [20] to manage and negotiate the heterogeneous and distributed business activity of enterprises. Currently, existing enterprises are expanding from local organizations to international distributed companies where business processes dynamically form temporary alliances, joining their business in order to share their costs, skills and resources in supporting certain activities [27, 6]. Many researches proposed intelligence agent technology to automate process operation integrating the different business systems to reduce the complexity [12, 13, 21]. Several benefits can be obtained by means of agent technology in the business process management [11], including autonomy, social ability, pro-activeness, and responsiveness. Agents perform the majority of their problem solving tasks without the direct intervention of humans or other agents, and they have control over their own actions and their own internal state. Agents interact, when they deem appropriate, with other artificial agents and humans in order to complete their problem solving and to help others with their activities. Agents take the initiative where appropriate and perceive their environment and respond in a timely fashion to changes that occur in it.

Moreover, mobile agent technology offers several advantages over traditional approaches to Internet applications [3, 24, 23]. It can proceed without continuous network connections to save significant bandwidth by moving locally to the resources they need, because interacting entities can be moved to the same site when connections are available and then interact without requiring further network connections. The mobile agent can carry the code to manage remote resources and do not need the remote availability of a specific server, so it has intelligent information gathering to work with mobile computing systems.

As complex business processes rely on intensive information exchange with the company's environment, they are document-driven by nature [1]: employees deal with and react to information and knowledge transferred by and embedded in all kinds of documents, including forms, letters, books, manuals, records, either electronic or paper-based. Documents reflect many results of business processes in the private as well as in the public sector [28]. Most of business processes are centered about the "disposal record", a document that allows a company to dispose of a certain type product or process in the approved way.

Consequently, one would like the business process management system to automatically offer access to relevant knowledge source, or to even directly "pump" information items extracted from incoming documents to the appropriate places in the data models of the actual workflow instance.

In this sense, we attempt to integrate mobile agent technology and document-center concept [1, 28] to provide flexible business process management system. In this paper, we base on our previous works [17, 18] to propose a document-driven agent-based (DAB) framework dealing with dynamic workflow management. The business documents can be packaged and handled by mobile agent over distributed business environment. We also use Aglet [19] to implement the business process management system for our case study explaining our approach. In the sequel, we first review the related researches. The DAB approach is fully discussed in Section 3. The case study is outlined in Section 4. Finally we conclude the paper by outlining the potential benefits of the proposed approach in Section 5.

2 Related Work

Works in a number of fields have made its mark on our research. Technologies for business process management systems are evolving across many different dimensions namely,workflow, intelligent agents and Internet. Our approach has drawn upon several ideas from the document-center concepts, and techniques in handling workflow management.

2.1 Document-oriented business process management

The genesis of workflow management software was probably in automating document-driven business processes [25]. Some of the early products were extensions to the document imaging and management software. The workflow management systems for office automation can support document management, imaging, application launchng, coordination, collaboration, and co-decision [7]. Presently, the systems must support enterprise-wide workflow applications effectively.

For flexible control and data flow modeling, Joeris introduced an approach to support cooperation on the workflow level which takes versioning and different forms of data interchange between activities into account [15]. They concentrate on both document level and workflow level within a comprehensive process

management system. A document may be decomposed into sub-documents and may depend on other documents as well. A task consists of some sub-tasks and atomic activities. Tasks and activities are linked by control flow dependencies which form a partial order. The backbone of the integration of tasks and documents is the input/output relationship which is defined by the parameter specification of a task.

The approval of hazardous-waste disposal is a highly regulated, interorganizational business process. Wewers *et al.* present a system that supports this process and embed it in a framework for interorganizational, document-oriented workflow [28]. Due to the document-orientation of the disposal record process, it can suitably be supported by workflow management systems that are based on document management system. In order to establish a workflow among several organizations, four aspects are identified: the organization, document and workflow control data, communication channels, and naming/directory services.

In [27], Aalst describes a Petri nets technique to model and analyze a workflow process. The generation of a workflow process is based on a bill-of-materials (BOM). A BOM specifies which materials are needed to manufacture a product. The well-known BOM document can be used to describe the product that is manufactured using a workflow management system. It allows workflow designers to think in terms of the end-product instead of the internal process and constitute a basis for the automatic configuration of a workflow management system on the basis of a BOM. The BOM provides product-centric view and the Petri-net provides a process-centric view.

The idea of coupling a user observation/task management system with a sophisticated information retrieval tool for proactive and context-sensitive [1],

2.2 Agent-based business process management

Usually, agents in workflow management systems act as personal assistants, performing actions on behalf of the workflow participants, continuing watching for information and responding to it when it meets certain specified criteria. In multi-agent system, agents have control over the tasks that they may perform, the resources available to them and how they coordinate their activities with other agents. To really exploit the capability of the web as the computing infrastructure other than merely the wonderful user interface, mobile agents, that can be dispatched to different locations to carry out various computations, can be

implemented in a workflow system [9].

A "WorkWeb System" [26] is an expanded workflow system, that is able to management, control and coordinate office resources, especially human resources. A business process tactics (BPT) agent in the system autonomously manages each workflow process instance, trying to acquire the necessary resources to complete it in time. The system also provides interface to manage office goals and several workflow re-planning algorithms to handle exceptional cases. The WorkWeb system also solves the resource conflict problem among workflow process instances and personal schedules.

In [8], Gou *et al.* treat agents as autonomous entities with abilities to solve problems independently. They propose an agent-based approach for workflow management, aining at achieving flexible and dynamic workflow management in the distributed environment. In their agent hierarchy, there are three kinds of agents at three neighboring levels: activity agents, role agents and actor agents. For organizing and collaborating the three kinds of agents at three levels, they provide two collaboration patterns inside the agent hierarchy: activity-role-actor assignment and feedback patterns. The proposed agent model consists of three sub-agents: a message-receiving sub-agent, a decision-making sub-agent, and a message-sending sub-agent. Connected by two message queues, these three collaboratively operating sub-agents can complete functions of an agent effectively.

ADEPT (Advanced Decision Environment for Process Tasks) [14, 11, 13] adapts multi-agent architecture composed of a number of autonomous agencies to provide a well structure for dealing with business process management. An agency contains a single responsible agent, a possibly empty set of subsidiary agencies and a set of tasks that are under the direct management of the responsible agent. The recursive definition of an agency allows a nested agent system to be constructed in which a responsible agent realizes its function through the responsible agents of lower level agencies. All ADEPT agents have the same basic internal architecture, illustrated by the responsible agent of agency. An agent has six functional components: communication module, interaction management module, situation assessment module, service execution module, acquaintance model, and self model. Although ADEPT has been attracting widespread interest in business process management, several developmental obstacles remain. All of agents are stationary in the environment, so the major weakness of this approach is the lack of monitoring, trace ability, and mobility for business activity.

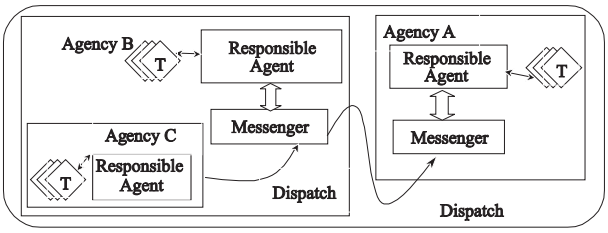

Figure 1: The external architecture of DAB framework

Table 1: Responsible Agent V.S. Messenger agent

	Responsible agent	Messenger agent
GUI	Yes	No
Communication	User, messenger	Responsible, messenger agent
Service execution	Maintain services, Schedule services	Maintain documents, Schedule ininerary
Mobility	No	Yes
Knowledge-based handler	Acquaintance	Agent locker pattern

3 Document-driven Agent-based Framework

We proposed the DAB, a document-driven approach that applied the intelligence agent and mobility technology to encapsulate a document as a mobile agent that handles the business activity. The mobile agent can trace and monitor the business document, so that the business process may be better managed.

3.1 The external architecture

The external architecture represents the structure of the multi-agent system as a whole, and the role of an agent within that system. The architecture is broadly based on the ADEPT model. The architecture is composed of a number of autonomous agencies (see Figure 1). Agency A and B are peers, and agency C is the subsidiary of agency B. The responsible agent of agency C generates a messenger agent that moves locally to agency B in order to interact with each other. The architecture can model the organizations of enterprises, such as structure of hierarchical or flat organizations, or a mixture of the two, through the concepts of agents and agencies. It also provides a design and implement pattern for multi-agent business process management systems.

In the proposed architecture, two agent types are provided: the responsible agent and the messenger

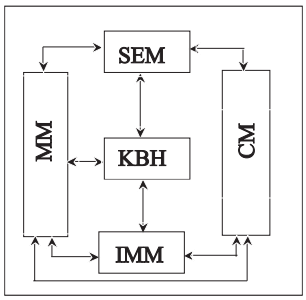

Figure 2: The internal architecture of DAB framework

agent. The responsible agent is a stationary agent that creates and dispatch the messenger agent into the right server for some missions. The user can control it through the user interface. The messenger agent is a mobile agent with no user interface, so a user cannot directly communicate with the messenger agent. A messenger agent can pack a document up and run the business process described in the document. The concepts between mobile agent and stationary agent differ in terms of the communication message, the life cycle of agent, and the cognition of context. In order to communicate with remote agents, the responsible agent created a messenger agent that moves itself to the remote content. The comparison between the two types of the agents is summarized in Table 1.

3.2 The internal architecture

All DAB agents have the same basic internal architecture (Figure 2). This involves an agent which is responsible for managing the agent's activities and interacting with peers and agency. The agent has a number of functional components responsible for each of it's main activities - communication, mobility, service execution, interaction management, and knowledge-based handler. The communication module (CM) routes messages between an agent and both its agency and peers. The mobility module (MM) is designed for the mobility ability of agent, including the mechanisms of dispatch, retract, dispose, clone, activate, deactivate, and create. The service execution module (SEM) is responsible for managing services throughout their execution. The knowledge-base handler (KBH) is a knowledge storage for the reasoning mechanism where the agent is committed, describing the services the agent can provided, running time application/service specific information and generic domain information. The interaction management

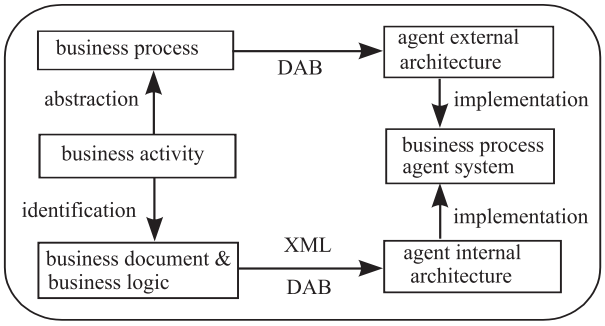

Figure 3: The framework of agent-based document-driven approach

module (IMM) provisions services through negotiation. The internal architecture is designed to ensure maximum flexibility to adapt as a business process changes.

By using mobile agent technology, there are three advantages described as follow: (1) Better communication: The messenger agent can move to any level of the agency to communicate with one another. By doing that, the network loading is reduced and the communication between agencies is more efficient and flexible, (2) Higher trace ability: An administrator can easily monitor the business processes through the messenger agent, and (3) Greater environmental sensitivity: We design the mobility module of the agent to capture the mobility and the knowledge-base handler to store the knowledge about the environment. Our design enables an agent to perceive its environment and respond in a timely fashion to the change occurred. Thus, the agent is able to schedule its itinerary and services dynamically.

3.3 Agent-based document-driven approach

Most workflow management systems follow an activity-centered approach and have serveral limitations with respect to support for cooperative work and data-centered processes. Further, they lack the ability of dynamic workflow modifications and flexible control and data flow modeling. Document-driven aspects are not only covered by cooperation support. Rather, we have to take document-driven processes like object life cycle models or object migration models into account. Furthermore, the document structure and dependencies may determine task decomposition and ordering. The evolution of the document content or structure and dependencies will cause evolution of the process instance which cannot be defined a priori. As suggested in Figure 3, we advocate an integrated approach to designing a DAB business process

management system as following three phases.

Analysis phase In this phase, we analyze the organizations of enterprise, the business documents and related activities from various industrial and commercial aspects. The output is the use case model. The steps of analysis are described as follow.

- To extract the key business documents: First of all, the key business documents that the involved organizations execute together. In general, business documents reflect many results of business processes in the intra-organization as well as inter-organizations.
- To identity the structures of organizations ans the roles of divisions: For each key document, we view the related organization as a collection of roles, that stans in certain relationships to one another, and an division as an actor, that deals with business activities based on the business document.
- To abstract the common the characteristic of business processes: The business process are defined across different organizations that serve to complete a task which is the goal of an business document. A scenario is an instance of a business process that expresses a specific occurrence of the business process with a specific employee operating at a specific time and using specific document.

Modeling phase In this phase, there are two major goals. According to the structures of organizations, we build the external architecture of multi-agent system. To extract the documents and its related business logic from the business process, we establish the internal architecture of agents. Both the architectures models can be first described in their high-level abstract forms, which can be further refined into more detailed specifications in the next phase.

- The external architecture: Each organization may have many divisions, then we construct one agency and some subsidiary agencies in the system. To defined an agency, it suffices to define the responsible agents of divisions in the organization, how these agents relate to one another, and how a agent can interact with other agents.
- The internal architecture: A business document which includes business logic can be packaged and handled by a messenger agent over distributed business environment. The messenger agent also manages the life cycle of the document. The responsible agent is a stationary agent that creates and

dispatch the messenger agent into the right server for some business activities. So the messenger agent has the mobility module, and the responsible agent has the agent management module.

Design phase The business logic comprises process descriptions, business rules, and the document related information. The dynamic properties of the agent system are characterized by (1) using the sequence diagram or activity diagram to explicitly describe what the sequence of a business process is and the interactions between organizations or divisions, (2) specifying the business rules and business policy using the ECA (Event-Condition-Action) rules of active database [4], and (3) The document related information may be the representation mechanism of documents (e.g. XML or XSLT [31]) or the business forms.

- To specify the process descriptions: For execution of the business activities of document, the business document is routed from one employee of division to the other. A sequence diagram shows the sequence of messages exchanged of the business document by the set of agents performing a certain business process. For business automation, we then formally specify the routing sequence using task state expressions [32].

- To formulize the business rules: ECA rules automatically perform actions in response to events provided stated conditions hold. They are used in data warehouses for incremental maintenance of materialized views, for validation and cleansing of the input data streams, and for maintaining audit trails of the data. An event can be message from an agent (e.g. a new document arrival), a property change (e.g. updating document) or a state transition (e.g. document rejected). Conditions are predicates that are used to decide what actions should be executed and are applied to the event or events that precede it and return true or false (e.g. trade price is within acceptable limits). Actions represent things to do, and are typically built-in functions (sending a message to the journaling database recording partial trade settlement) or any business strategy.

- To represent the business document: XML is becoming a dominate standard for exchanging and storing information in e-commerce or data warehousing area. The role of XML in electronic applications is to promote open and highly reliable exchange of document between many unspecified private companies by describing the document rules in formal way. With regard to one XML document, there are DTD (document type definition) [5] and XML schema [29, 30] as the existing frameworks

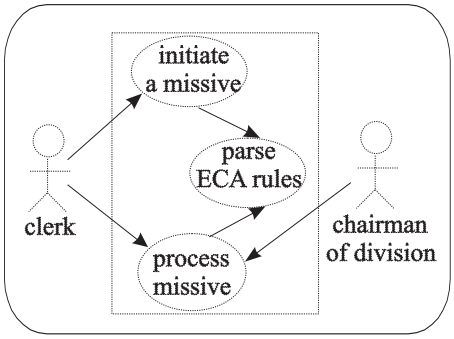

Figure 4: The use case diagram of official missives management system

to describe document rules. We use XML to represent the business documents and business logic. We also use XSLT to control the access rights of the business documents.

Finally, we can use an agent development kit to implement the agent system (e.g. Aglets [19], Voyager [22] or JATLite [10]).

4 A Case study: Official Missives Management

This section briefly illustrates how the method can be applied, through a case study of the analysis and design of an document-driven agent-based system for managing an official missive process. First, we use the use case diagram and the activity diagram to analyze the management system. Second, once the system architecture is established, the internal and external architectures of agent system are designed to fit the management system. Finally, the official missives management system is implemented by Aglet [19].

4.1 The requirement analysis of document-driven agent system

The process is initiated by An office clerk of a department at the university. The clerk writes a missive and sends it to the chairman of his department. The chairman of department gives some comments and then sends the missive to the chairman of the college. The chairman of the college gives some more comments and then sends the missive to the office of Academic Affairs, Student Affairs, General Affairs, Accounting, and Personnel. The directors of these offices give their comments, and then send the missive

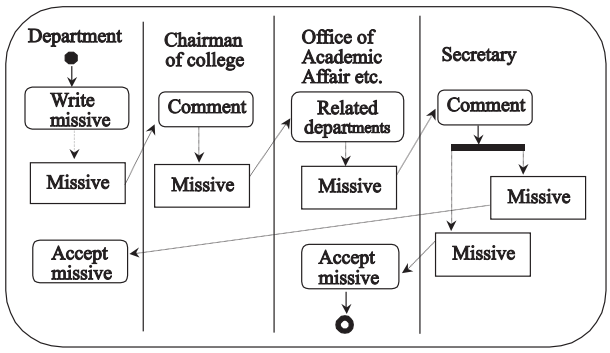

Figure 5: The activity diagram of official missives management system

to the secretariat. As soon as the head of secretariat receives the missive, she/he sends it to the president of school. Finally, the missive is sent back to the department clerk.

We identify two actors: clerk and chairman of division. The use case diagram is shown as Figure 4. Then the scenario of the agent-based management system is shown as follow:

- Each user of department of college first activates their document agent system, and then the agent system triggers the web browser. The user can manage their document (messenger) agent through the web browser.
- When the user wants to write a missive, he will have to choose the type of missive and the degree of emergency. The ECA Parser will check the business logic of the missive and the roles participated this process.
- If there is any error occurred in the process of parser, the system will demand the user to make some specific modification. Otherwise, the document control agent will create a document (messenger) agent to finish the process of missive.
- The document agent carries the missive to the destination office and queues the schedule along with other missives from another department. If the missive is an emergency missive, the document agent will create a new window and notify the user to deal with the missive in time.
- The missive sender can trace where the missive is and what the situations are. The situations of missive contain the arrival time, the finish time, the waiting time, and any exception event.

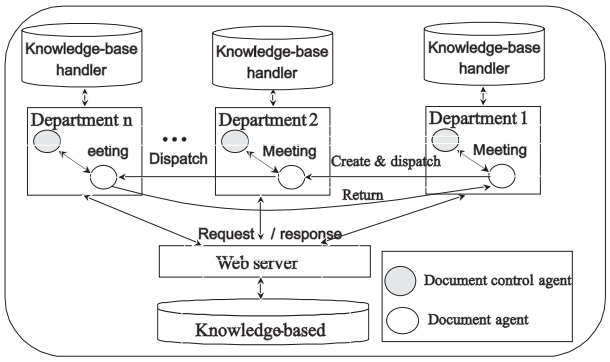

Figure 6: External Architecture of Agent System

- The missive receiver can click the review function. If there are too many unread missives coming in to the receiver, the document agent will give the receiver a warning. If the receiver completes the review, the agent system will continue to route the missive based on the business logic and the knowledge of agent.
- If the user has any problem, he can send message to the system by message agent.
- Finally, the document agent takes the missive and the results back to the original sender.

Finally, We use the activity diagram to model the system behavior as figure 5.

4.2 The architecture design of the agent system

The official document agent system is a web-based system, so users can use the web browser to interact with each other. Each department has an Aglet agent environment that is viewed as an agency of the agent system. Each agency encompasses the document control agent, the document agent, the message agent, and the knowledge-base handler (see figure 6).

The document control agent that is the extend of the responsible agent is responsible for managing the document agent and the message agent to provide resources and services for agency. The document control agent dispatches a document agent to other agencies in order to request or supply a service. A user can control the document control agent by web browser. The document control agent monitors the missive through managing the document agent. The document agent extended the messenger agent is charged with the process of missive. It packs and processes the missive over the Internet. The process order of

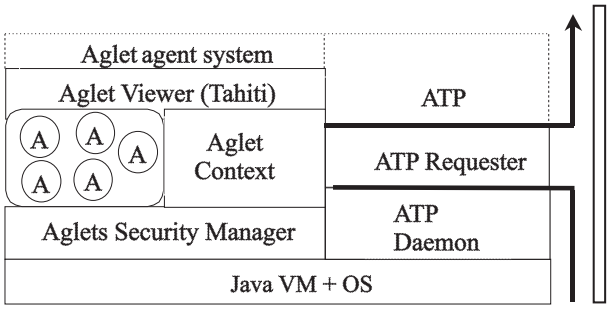

Figure 7: Aglet environment

the missive depends on its degree of emergency. If it is a normal missive, the document agent must make a meeting with others prior to make decision about the process order. If it is an emergency missive, the user can deal with the missive directly in time. The message agent inherited from the messenger agent provided the communication mechanism. The user of the business department can send messages through the message agent. The knowledge-base handler is the knowledge storage for reasoning mechanism in which the document control agent is committed, describing the services the agent provided, running time application/service, and the specific information.

A user can first use web browser to write a missive, and the agent system can later apply an ECA Parser to parse the business logic of the missive. If the missive passes the parser, the document control agent may get a notification from the agent system and create a document agent to complete the mission of missive. The schedule of missive is determined through a meeting where the document control agent and document agent negotiate with one another. The chairman of the negotiation is the document control agent, and all of the document agents in the same context are the members of meeting. Based on the information regarding waiting time, degree of emergency, and type of missive, the meeting will choose the next document agent to deal with its missive. The negotiation mechanism is implemented by the multicasting technology of Aglet.

4.3 The implementation of the agent system

Aglet [16, 19] is designed for development the mobile Java agent system that supports the concepts of autonomous execution and dynamic routing on its itinerary. In the aglet object model, a mobile agent is

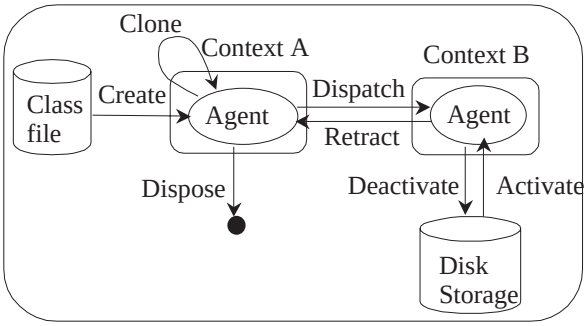

Figure 8: Aglet life-cycle model

a mobile object that has its own thread of control, is event-driven, and communicates by message passing. This model defines a set of abstractions and the behavior needs to leverage mobile agent technology in Internet-like, open wide-area networks. The Agent server provides an environment for aglets to execute in, and the Java virtual machine and the Aglet security manager make it safe to receive and host aglets. Aglets framework use ATP (Agent Transfer Protocol) as a network protocol for dispatching aglets. The ATP daemon is responsible for receiving messages and giving safe authorization (see Figure 7).

There are two ways to bring an aglet to life: either it is instantiated from scratch (creation) or it is copied from an existing aglet (cloning). To control the population of aglets you can destroy them (disposal). Aglets are mobile in two different ways: (1) actively: An aglet pushes itself from its current host to a remote host (dispatch); and (2) passively: A remote host pulls an aglet away from its current host (retract).When aglets are running, they take up resources. To reduce their resource consumption, aglets can go to sleep temporarily, releasing their resources (deactivation) and later can be brought back into running mode (activation). Finally, multiple aglets can exchange information to accomplish a given task (messaging). The aglet life cycle is shown as figure 8.

4.4 Discussion

Using the mobile agent technology, a business document can be packed by a document agent. Thus, the document-driven agent system provides three properties: (1) trace ability: a function enables an administrator to monitor document processes easily, (2) document life cycle: a feature using mobile agent life cycle to manage document life cycle and concurrent processing, and (3) dynamic scheduling: this

function contain two components: a document agent that can dynamically schedule its itinerary, and a document control agent that can dynamically schedule its services.

Trace ability For missive sender, it is importance to know where the missive is and what the situations are. The situations of missive contain the arrival time, the finish time, the waiting time, and any exceptional event in a destination department. The document agent routinely analyzes the information in order to plan the next route. We use mobility listener of Aglet implementing the document agent to achieve the trace ability of agent system. The mobility listener can capture the event triggered by mobility reason. The types of event contain dispatching, retraction, activation, and deactivation. We also apply the locker pattern [2], an agent design pattern, to store the private information of the document agent, such as the beginning time, arrival time, and completing time of missive. The document control agent monitors where the missive is and what degree the missive completes through querying the document agent.

Document lifecycle The lifecycle of a document agent is similar to the life cycle of Aglet. It includes create missive, copy missive, suspend missive, resume missive, and dispose missive. Each document agent encapsulates a document and manages its lifecycle. The lifecycle of a document agent is managed by the lifecycle of Aglet.

Dynamic scheduling We use mobility technology to improve the internal architecture that enables agents to obtain the environment sensibility. The document control agent can dynamically change the schedule by perceiving the services, types, and amount of document agent. If many missives gathered in a department or a department is suspended, the document control will change its regular route to another department and come back to the previous department later. The document agent perceives their environment and responds in a timely fashion to any change that has occurred. For example, if the context of a department is restarting or shutting down, the document agent will be deactivated or move to another context.

5 Conclusion

In this paper, we have proposed an agent system approach, which applied the document-center concept and the mobile agent technology to enable the construction of business process management. The DAB approach can build a document-driven agent system, which encapsulates a single document per agent. In

terms of communication, the mobility affects the external architecture of agent system in order to reduce the network loading and to make the communication between agents more flexible. The agent life cycle and the environment sensibility influence the internal architecture of agent system that enables the dynamic schedule.

The DAB provided an architecture and process to develop the document-driven business process management system. It also supplied a mechanism for communication between enterprises or divisions of enterprise. Its strengths include: (1) trace ability: a function that enables administrators to monitor document processes easily, (2) document life cycle: a feature using agent life cycle to manage document life cycle and concurrent processing, and (3) dynamic scheduling: a document agent can dynamically schedule its itinerary, and a document control agent can dynamically schedule its services.

References

[1] A. Abecker, A. Bernardi, H. Maus, M. Sintek, and C. Wenzel. Information supply for business process: Coupling workflow with document analysis and information retrieval. *Knowledge-Based Systems*, 13:271–284, 2000.

[2] Y. Aridor and D.B. Lange. Agent design patterns: Elements of agent application design. In *Agents'98*, pages 108–115, 1998.

[3] G. Cabri, L. Leonardi, and F. Zambonelli. Mobile-agent coordination models for internet applications. *IEEE Computer*, pages 82–89, Feb. 2000.

[4] U. Dayal, A.P. Buchmann, and D.R. McCarthy. Rules are objects too: A knowledge model for an active object-oriented database management system. In K.R. Dittrich, editor, *Advances in Object-Oriented Database Systems*, pages 129–143. Springer, Berlin, 1988.

[5] Extensible Markup Language (XML) 1.0 c. *http://www.w3.org/TR/ 1998/REC-xml-19980210.*

[6] D. Georgakopolus, H. Schuster, A. Cichocki, and D. Baker. Manageing process and service fusion in virtual enterprises. *Information Systems*, 24(6):429–456, 1999.

[7] D. Georgakopoulos and M. Hornick. An overview of workflow management: From process modeling to workflow automation. *Distributed and Parallel Databases*, 3:119–153, 1995.

[8] H. Gou, B. Huang, W. Liu, S. Ren, and Y. Li. An agent-based approach for workflow management. In *Proceedings of IEEE International Conference on System, Man, and Cybernetics.*, volume 1, pages 292–271, 2000.

[9] M.N. Huhns and M.P. Singh. Workflow agents. *IEEE Internet Computing*, 2(4):94–96, 1998.

[10] JATLite. *http://java.stanford.edu/java_agent/html/.*

[11] N. R. Jennings, T. J. Norman, and P. Faratin. Adept: An agent-based approach to business process management. *ACM SIGMOD*, 27(4):32–39, 1998.

[12] N.R. Jennings, P. Faratin, M.J. Johnson, T.J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 5(2,3):105–130, 1996.

[13] N.R. Jennings, P. Faratin, T.J. Norman, P. O'Brien, and B Odgers. Autonomous agents for business process management. *International Journal of Applied Artificial Intelligence*, 14(2):145–189, 2000.

[14] N.R. Jennings, P. Faratin, T.J. Norman, P. O'Brien, M.E. Wiegand, C. Voudouris, J.L. Alty, T. Miah, and E.H. Mamdani. Adept: Managing business processes using intelligent agents. In *BCS Expert Systems 96 Conference (ISIP Track)*, pages 5–23, Cambridge, UK, 1996.

[15] G. Joeris. Cooperative and integrated workflow and document management for engineering applications. In *Proceedings of 8th International Workshop on Database and Expert Systems Applications*, pages 68–73, 1997.

[16] G. Karjoth, D.B. Lange, and M. Oshima. A security model for aglets. *IEEE Internet Computing*, 1(4):68–77, 1997.

[17] J.Y. Kuo. Mobile agent technology in business processes management. *Fu Jen Studies, Science and Engineering*, 35:53–72, 2001.

[18] J.Y. Kuo and J. Lee. Xml-based virtual enterprise model: A document-driven agent-based approach. In *Twelfth Workshop on Object-oriented Technology and Application*, pages 103–111, Tainan, 2001.

[19] D.B. Lange and M. Oshima. *Programming and Deploying Java Mobile Agents with Aglets*. Addison-Wesley, 1998.

[20] F. Manola, S. Heiler, D. Georgakopoulos, M. Hornick, and M. Brodie. Distributed object management. *International Journal of Intelligent and Cooperative Information Systems*, 1(1), Mar, 1992.

[21] T.J. Norman, N.R. Jennings, P. Faratin, and E.H. Mamdani. Designing and implementing a multi-agent architecture for business process management. In *the ECAI-96 Workshop on Agent Theories, Architectures, and Languages (ATAL-96)*, pages 149–161, 1996.

[22] ObjectSpace's Voyager. *http://www.recursionsw.com/products/voyager/ voyager.asp*.

[23] T. Papaioannou and J. Edwards. Mobile agent technology in support of sales order processing in the virtual enterprise. In *The 3rd IEEE/IFIP International Conference of On Information Technology for Balanced Automation System in Manufacturing (BASYS'98)*, pages 23–32, 1998.

[24] T. Papaioannou and J. Edwards. Using mobile agents to improve the alignment between manufacturing and its it support systems. *Journal of Robotics and Autonomous Systems*, 27:45–57, 1999.

[25] T. Smith. The future of work flow software. *INFORM*, April 1993.

[26] H. Tarumi, K. Kida, Y. Ishiguro, K. Yoshifu, and T. Asahura. Workweb system - multi-workflow managament with a multi-agent system. In *Proceedings of the ACM SIGGROUP Conference*, pages 299–308, 1997.

[27] W.M.P. van der Aalst. On the automatic generation of workflow processes based on product structures. *Computers in Industry*, 39:97–111, 1999.

[28] T. Wewers and C. Wargitsch. Four dimensions of interorganizational, document-oriented workflow: A case study of the approval of hazardous-waste disposal. In *The 31th Hawaii International Conference*, volume 4, pages 332–341, 1998.

[29] XML Schema Part 1: Structures. *http://www.w3.org/TR/xmlschema-1*.

[30] XML Schema Part 2: Datatypes. *http://www.w3.org/TR/xmlschema-2.*

[31] XSL Transformations (XSLT) Version 1.0. *http://www.w3.org/TR/xslt.*

[32] J. Yen and J. Lee. A task-based methodology for specifying expert systems. *IEEE Expert*, 8(1):8–15, Feb. 1993.