An Agent Architecture for Personal and Group Work Support

Yoshihide ISHIGURO, Hiroyuki TARUMI, Takayoshi ASAKURA, Koji KIDA, Dai KUSUI and Kenji YOSHIFU
Kansai C&C Research Laboratories, NEC Corporation
1-4 24 Shiromi, Chuo-ku
Osaka 540, Japan
ishiguro@obp.cl.nec.co.jp

Abstract

People working in an office concurrently engage in two or more office activities. These activities can be classified into two groups: personal and group work. To increase office productivity with computer software, both personal and group work should be supported. However, the currently available office work support software, such as groupware or a software secretary, focuses on either of these activities. In this paper, a multiagent-based office work support system and its architecture are introduced. In this system, there exist agents for personal and group work, and they are mutually connected in a two-layered agent network. With the support of these agents, people in an office can expect to efficiently perform both their personal and group work.

Introduction

People working in an office generally engage in two or more office activities. For example, a worker may write documents, such as his/her own letters, while participating in other software developing projects. These activities can be classified into two groups: personal and group work. Personal work involves tasks that can be processes at a single worker's discretion, while group work involves tasks that can not be processed at an individual discretion. Figure 1 shows a typical example that involves these two activities. This example shows some tasks in a car dealer office. Tasks included in a workflow process through a car order reception to a car transportation are group work. Each worker allocated to each workflow step is expected to complete his/her task by a given deadline. On the other hand, each worker engage in their personal work independent to his/her group work. For example, order operators register orders, report them to their superiors, and also check date of their car transportation to customers. Mechanics maintain their machine tools and drivers schedule their car transportation to customers. Additionally, workers may be allocated other workflow tasks because many workflows are concurrently running in an office. It is possible for a part of some group work and personal work, having the same deadline, to be given to one worker because group work and personal work are not mutually concerned with convenience of actions. This is an office resource allocation problem. However, the currently available office work support software, such as groupware or a software secretary, focuses on either of these activities. It therefore poses the risk of causing the office resource allocation problem and leads to decrease office productivity. To increase office productivity, both personal and group work should be supported in harmony.

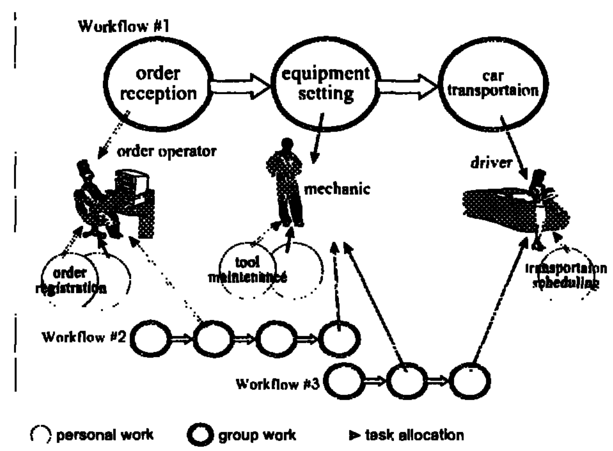

Figure 1: Some examples of group and personal work in a car dealer office.

In this paper, a multiagent-based office work support system and its architecture are introduced. In this system, there exist agents for both group and personal work, and they are mutually connected in a two-layered agent network. With the support of these agents, people in an office can expect to efficiently perform both their group and personal work.

The remainder of this paper is organized as follows. The next section describes requirements of the office work support system and introduces several systems using agents. Then, the architecture that we have introduced to the system, and its implementation and applications based on this architecture are described. Finally, a conclusion is made in the last section.

Agents in the Office

Requirements for the Office Work Support System

Workers in an office concurrently perform group and personal work. To increase the productivity of these tasks, we focus on the productivity of the worker.

From the viewpoint of the productivity of people in an office, office work can be viewed from two different views: *Personal view* and *Group view*.

Personal view This view focuses on the productivity of the individual worker. Each worker personally pays attention on how to process his/her own work, what is the best order to process them, etc. To support such work completely, various tools for eliminating the load of each task, such as a schedule management software, is needed. The key feature for eliminating the load of a task is "automation." With "automation", the worker can leave simple personal work to these tools and perform other work concurrently.

Group view This view focuses on the productivity of the group. A manager, who controls the group, considers the most suitable person for a given task, and how the work is to proceed, but does not pay attention to the details of each work. The manager utilizes a workflow management system to support the work. The key feature of this system is "communications." With this capability, the manager can collect individual information quickly.

An office work support system should have both personal and group work support functions, that is, both personal and group views should be available.

To provide both personal and group views to the office work support system, a multiagent architecture is introduced to the system. In this system, there are several types of agents. Some agents act as a load-reducing tools to support personal work, such as information filtering, schedule management, etc. Other agents provide group work support functions, such as workflow management. Personal support is formed by a group of the former agents. A worker can use these agents like office tools. Group support is formed by a group of the latter agents.

Agent Systems for the Office Workers

There have been some research results that support office work using agents, and there is no research result that has an architecture to support both personal and group work simultaneously.

Maes and Kozierok's program mainly focuses on personal work (Maes & Kozierok 1993). This program, called the interface agent, observes its user's actions and imitates them by learning. For example, the agent observes its user's interactions with a calendar application. When it can assume the user's response to a certain appointment query, it may offer a suggestion, or even automate the action if its "confidence" is high enough.

Bocionek's program, called CAP II, also focuses on personal work (Bocionek 1995). It also deals with the automation of planning meeting appointments. Like Microsoft's Schedule+, the CAP II agent uses human-readable electronic mail that sends/receives meeting information. A user who possesses no CAP II agent can reply to a meeting request from CAP II.

Additionally, Kautz et al. focus on personal work (Kauts *et al.* 1994). The key feature of their framework is the use of personalized agents called "userbots." These userbots mediate communications between users and task-specific agents, called "taskbots." In their example, a "visitorbot", which is a kind of taskbot, arranges meetings between a visitor and several researchers in a laboratory, by interacting with the researchers through their userbots. This approach is similar to ours, but the userbot does not represent its user; instead, it provides only communication methods to taskbots and other users. That is, when a visitorbot asks about a schedule preference to a userbot, the user of the userbot is obliged to decide and make a reply immediately.

Yamaki et al.'s program, called "Socia", mainly focuses on group work, but also eliminates individual loads in the task of planning a meeting appointment (Yamaki *et al.* 1995; 1996). This system automates the scheduling of a desktop conference. An agent, which exists on a user's desktop computer, obtains the presence of its user from a video camera and replies to the agent of the meeting arranger as to whether or not the user can attend the meeting.

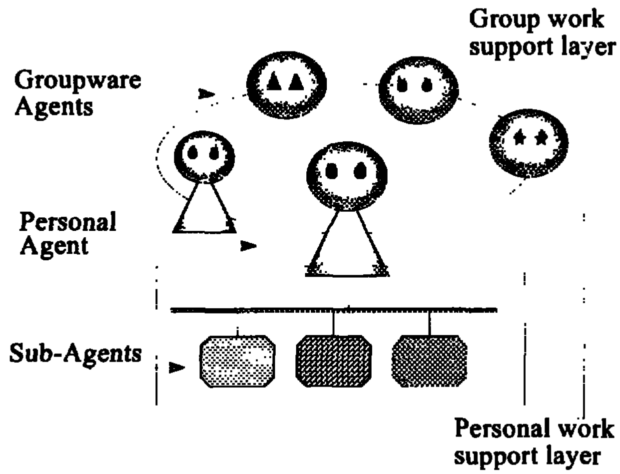

Figure 2: Two-layered agent architecture.

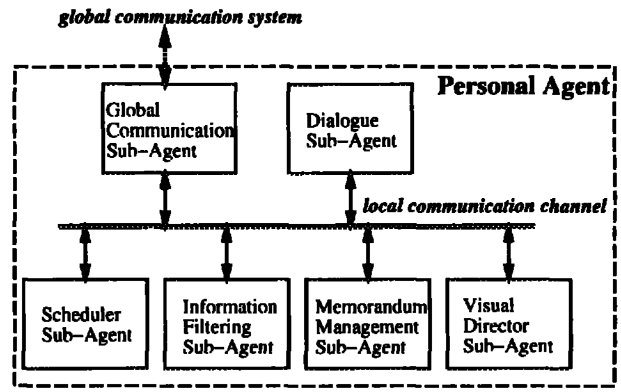

Figure 3: Configuration of a personal agent.

INA/LI Architecture

The Layered Architecture

To provide two different views to the system, a two-layered architecture, which we call the *INA/LI*[1] *architecture*, is introduced. Figure 2 shows the configuration of this layered architecture. A group of agents on the two layers give a worker the global and personal views.

The first layer is a *Group work support layer*. This layer provides a communication channel among *Groupware Agents* and *Personal Agents*. The groupware agents are related to group work such as workflow management, office resource management, etc. The personal agent represents an individual worker, which maintains personal information such as a personal schedule, the progress of tasks, etc. This communication capability among personal and group agents realizes the group view, that is, a manager can quickly collect information on other workers, the reservation status of meeting rooms, etc.

The second layer, called *Personal work support layer*, exists inside the personal agent. This layer provides a communication channel among *Sub-Agents*. These agents act as software tools for the individual worker, and together provide the personal view to the worker.

The personal agent is described in the next section and description of groupware agents follows this section.

Personal Agent

A personal agent has two main functions:

(1) To support its user's group work by representing the user, such as collecting information on the schedules of members when the user is scheduling appointments, or answering questions on the user's schedule when other agents ask.

(2) To carry out simple personal work of the user, like a human secretary would.

When a user performs personal work, the personal agent is expected to offer some load-reducing tools to the user. As for group work, on the other hand, such as scheduling appointment, the personal agent should be considered as only one representative of the user. This is because a user would not want to search for sub-agents of another user who could answer his/her questions but would simply prefer to ask only one representative of the user about the user's schedule.

These two functions, realized with the two-layered agent architecture described above, provide global and personal views to the user, and consequently support group and personal work simultaneously.

Figure 3 shows the configuration of a personal agent. The personal agent consists of several sub-agents. These sub-agents communicate with each other through the local communication channel to achieve cooperation. Each sub-agent basically provides a simple function and together they work as the user's secretary. In the current implementation, there are six kinds of sub-agents, but new sub-agents can be developed and added to this system.

These sub-agents have the following functions.

Scheduler sub-agent Manages the appointment table and to-do items for the user. It also gives human resource information to the other agents through the global communication system.

[1] INA/LI stands for Intelligent Agent/Lite.

Information Filtering sub-agent Maintains information on the user's sphere of interests and automatically sorts incoming electronic information such as e-mail, Internet news, and the World Wide Web (WWW) pages. This sub-agent helps the user find the most important information from the vast amount of information.

Memorandum Management sub-agent Manages the user's electronic memoranda which are linked to some electronic information such as e-mail, news, and WWW pages. On a memorandum, the user's comments can also be added. These comments are collected and sent to the information filtering agent as a source of the user's interests. This agent also helps the user find information on the computer. By finding a certain memorandum, the user can easily find any e-mail, news, or WWW page linked to that memorandum.

Visual Director sub-agent Provides a visual user interface for the data management agent, and allows the user to view data.

Global Communication sub-agent Provides global communications to the other sub-agents. All incoming/outgoing global messages pass through this agent. Messages to global agents[2] are re-composed and sent to the receiver, and messages from global agents are sent to the appropriate sub-agent, using a function of the dialogue sub-agent.

Dialogue sub-agent Provides a goal-oriented user interface. This sub-agent is a kind of message distributor or *facilitator* (Cutkosky *et al.* 1993). Each command from the user, which is a sequence of key-words, is composed in the agent message format and sent to the appropriate sub-agents. It also distributes messages from global-agents, using the same mechanism. This agent helps the user interact with other agents and sub-agents.

The minimum set of the personal agent consists of the global communication sub-agent and dialogue sub-agent. With these two sub-agents, the other sub-agents can interact with global agents, and other global agents can regard the set of sub-agents as one global agent, that is, a personal agent.

The personal work support layer enables the cooperation of sub-agents. For example, if the memorandum management sub-agent has a link to a certain e-mail message, and is given some comments from the user, it sends the comments to the filtering sub-agent. The filtering sub-agent, which receives this message, can give a higher priority to the processed information so that the user can easily get related information.

[2]The term "global agent" is used to refer to an agent that has a global communication capability, that is, the personal agent and the groupware agent.

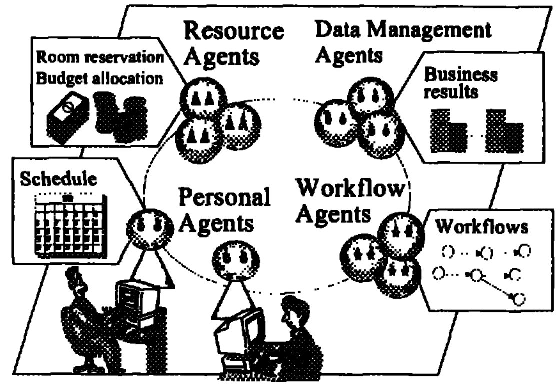

Figure 4: Agents in the WorkWeb System.

Agents to Support Group Work

Figure 4 shows the global agents in our system based on the INA/LI architecture, called the "WorkWeb system " (Tarumi *et al.* 1996). Four types of agents, including the personal agent, are defined to support both group and personal work: resource agents, data management agents, workflow agents, and personal agents. The first three types of agents are concerned with group work support and called groupware agents.

Resource agent Manages non-human resources: budget allocations, meeting rooms, shared tools or facilities, etc.

Data management agent Does not manage any resources, but collects group sharable data from the agents according to some given themes and gives back the collected information to the agents per their requests. This agent is called the "GIM[3] agent" in our system.

Workflow agent Manages each workflow process instance and tries to complete it within deadline or as early as possible. This agent is called the "BPT[4] agent" in our system.

Personal agent Mainly supports personal work, as already described in previous section.

[3]GIM: Group Information Management

[4]BPT: Business Process Tactics

These agents communicate with each other using the global communication system, which exists in the global communication layer. For example, a user who wants to plan a meeting appointment, asks his personal agent to collect the schedule information of other members. The personal agent sequentially communicates with each member's personal agent and gets their schedule information. Then when the meeting appointment is fixed, the personal agent communicates with the resource agent to reserve a meeting room.

Communication Protocol

In the INA/LI architecture, there are two communication protocols for group and personal work support layers described before.

Global Communication A global communication protocol is used for peer-to-peer communication among global agents. Each global agent is given a unique address represented by a Uniform Resource Locators (URL) format. For example, the personal agent for Mr. Tanaka has the following URL address.

inalip://obp.cl.nec.co.jp/people/tanaka

Here, *inalip* is the original protocol name, "obp.cl.nec.co.jp" is the domain name, and "people" is a keyword that indicates the address of a personal agent. All personal agents belonging to this domain are collected into this "people" directory and have a unique name that corresponds to a certain user's name such as "tanaka."

A workflow agent has the following URL address.

inalip://obp.cl.nec.co.jp/bpt/1

This kind of URL format is used for a dynamically generated address. "Bpt" is the subdirectory with workflow agents (BPT agents) and "1" is a serial number generated by the agent management process. This process generates an agent address by applying a predefined generation rule and registers it to the agent table or erases it when the agent finishes execution.

These mechanisms are implemented with OMG's CORBA[5] (OMG 1993). Implementation of the global communication system is described later in implementation section.

The global communication message format is defined based on the KQML (Finin & Weber 1993) format. Two types of performatives are used from the KQML: "evaluate" and "reply". Global messages are always used with a pair of "evaluate" and "reply". The following is an example of an "evaluate" global message.

[5] Common Object Request Broker Architecture

```
(evaluate
  :content (MessageToBeSent)
  :language inali
  :ontology inali
  :reply-with ID
  :sender  inalip://HostA/people/ishiguro
  :receiver inalip://HostB/people/tarumi  )
```

Here, the language and the ontology are currently fixed to "inali" only.

Local Communication The local communication protocol is used for sub-agent communications. Unlike the global communication, all local communication messages are processed by one main process: the local communication channel. Each sub-agent is always connected to the local communication channel and exchanges messages through this channel. Each sub-agent's address is represented by its name because local communication is limited inside the same machine.

In this local communication protocol, there are three kinds of messages and two types of addressing methods.

The three message types are "Request", "Answer", and "Order". "Request" and "Answer" form a pair of messages for synchronous communication. These messages correspond to the global messages' "evaluate" and "reply". "Order" is a message for asynchronous communication without ACK.

The two addressing method types are "Private" and "Broadcast". "Private" addressing is normally used to send a message to one sub-agent. "Broadcast" addressing is used to send a message to all sub-agents currently connected to the local communication channel. It is useful for collecting different solutions from different agents at the same time.

Linking between Local and Global Agents In the INA/LI architecture, global agents and sub-agents must be able to communicate with each other. Accessing a global agent from a sub-agent or vice versa is achieved by the global communication sub-agent and dialogue sub-agent. Figure 5 illustrates the message handling process between personal and group work support layers.

There are two personal agents X and Y. Agent X contains a sub-agent A and a global communication sub-agent (called "GBComAg" for short) and Y contains a dialogue sub-agent, a sub-agent B, and GBComAg.

Message transfer between agents is achieved by following steps.

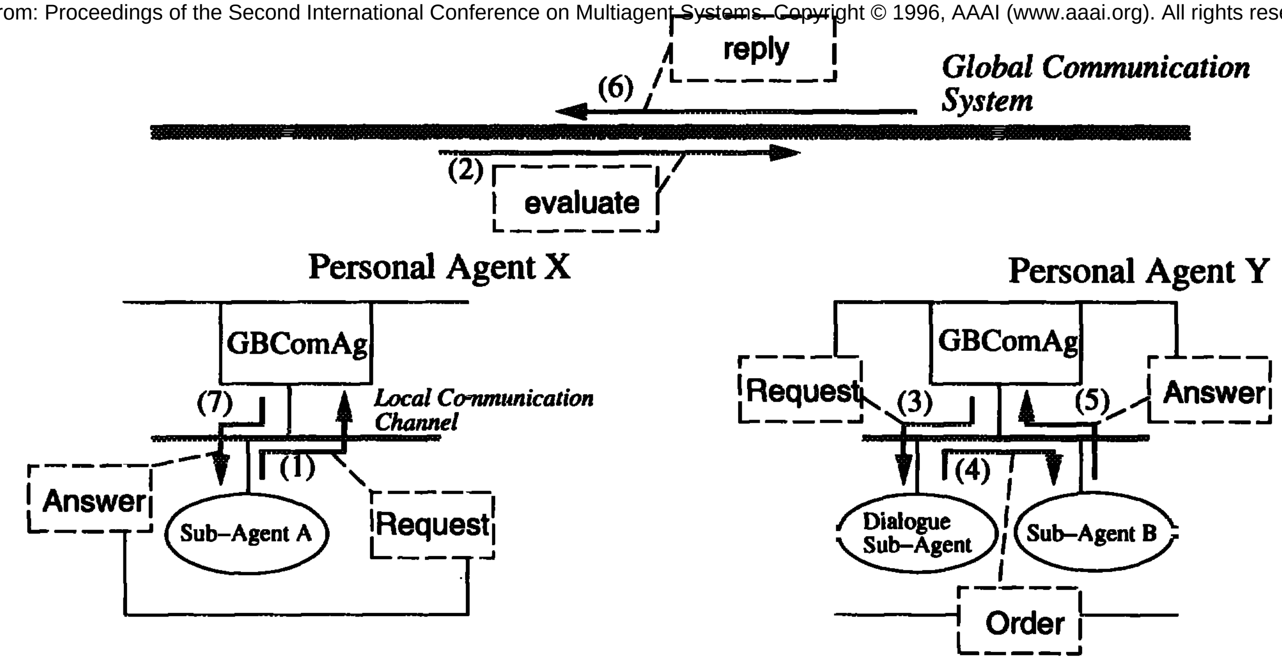

Figure 5: Message handling process between local and global communication system.

(1) When sub-agent A sends a message to personal agent Y, A sends a request to GBComAg of personal agent X.

(2) GBComAg translates this message into an "evaluate" message and sends it to personal agent Y. GBComAg also registers the ID of the message to the outgoing message table in order to decide on, at a later point (process (7)), the receiver sub-agent of the "reply" message.

(3) The message, received by GBComAg of agent Y, is translated into a local communication message and sent to the dialogue sub-agent.

(4) The dialogue sub-agent analyzes the received message and decides on a sub-agent who can reply to this message, and sends it to that agent (in this case sub-agent B).

(5) Sub-agent B makes a reply for received message and sends an "Answer" message to GBComAg.

(6) Upon receiving the "Answer" message, GBComAg translates it into a "reply" message and sends it to personal agent X.

(7) GBComAg of personal agent X receives the "reply" message and translates it into an "Answer" message, and sends it to sub-agent A. At this point, GBComAg knows the receiver of the answer message by checking the message ID against one in the outgoing message table.

Thus, the exchanging of messages between global agents and sub-agents is achieved. The most important feature is that the sub-agent can access a sub-agent in another personal agent only by specifying the personal agent's address. Therefore, the sender of a message need not know the details of the receiver agent. This is one of the advantages of the INA/LI architecture.

Advantages of the INA/LI Architecture

This section summarizes some advantages of the INA/LI architecture. The INA/LI architecture has advantages for both office workers and agent software developers.

(1) Advantages for office workers.

Agents on the INA/LI architecture support both group and personal work. Office workers can directly access group sharable information through data management agents or receive notifications for upcoming tasks from workflow agents. And if a worker wants to know another worker's related information, he/she simply asks another worker's personal agent. He/she does not need to specify nor to know an actual sub-agent name such as scheduler sub-agent. This leads to reduce the amount of time for the workers to get information. This is the same style as workers do in their actual office. For example, if a worker wants to know about another's schedule, he/she does not explicitly specify information source like a schedule book but only asks about schedule.

(2) Advantages for software developers.

The INA/LI architecture also gives advantages to agent software developers. Sub-agents can be added to or even removed from the system without considering other personal agents. This is because sub-agent's name is not specified in the global agent communication processes and a sub-agent that should receive the message is decided by the dialogue sub-agent in the personal agent. If dialogue sub-agent can not find an appropriate destination sub-agent, it forwards the message to the user in order to make a reply message. This simple mechanism guarantees to avoid the communication failure.

Consequently, a system designer can change the size of this system according to user request.

Finally, compared with the federation architecture such as PACT (Cutkosky *et al.* 1993), the INA/LI architecture has following features.

- Groupware agent acts like a meta-facilitator.

  Facilitators in the federation architecture correspond to dialogue sub-agents and global communication sub-agents in the INA/LI architecture. These agents manage sub-agent's group and deliver messages to appropriate sub-agents. However, there is no correspondence agent in the federation architecture to the groupware agent. This kind of agent manages group of personal agents which include several sub-agents. For example, a workflow agent dynamically allocate workers according to workflow definition. Consequently, the workflow agent make a group of personal agents and manages it.

- Users are included in the INA/LI architecture as sub-agents.

  Considering a user as one of sub-agents, a dialogue sub-agent will send messages that can not be processed by any other sub-agents to the user.

Implementation

The agents based on the INA/LI architecture are implemented in the Windows NT environment using the C++, BASIC, and Pascal languages.

The local communication channel is implemented using a DDE[6] function of Windows. It consists of one program and libraries: a local communication program

[6]Dynamic Data Exchange

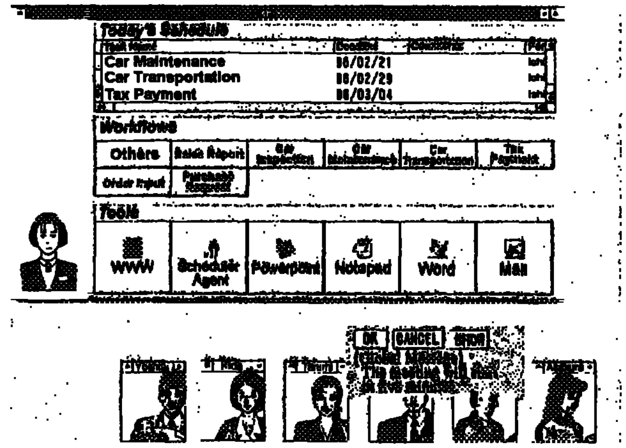

Figure 6: A screen image of a personal agent. The personal agent window consists of information field from scheduler sub-agent, buttons to invoke workflows, and buttons to invoke applications or sub-agents. Small windows represent the other workers' personal agents. There is a notification message from one personal agent.

and three kinds (Visual C++, Visual Basic, and DELPHI) of libraries for connecting sub-agents with a local communication program.

The global communication system is implemented using CORBA. It consists of two programs and one library: a global communication agent, a directory agent, and a global communication library. The directory agent manages an agent name table. It also generates a global agent's address with another agents' request according to predefined naming rules. The global communication library enables a program to act as a client or a server at anytime. Using this library global agents can exchange messages with each other.

Both local and global communication software are provided in an agent development kit.

Some Applications based on the INA/LI Architecture

Some applications based on the INA/LI architecture, which are part of the WorkWeb system, have been developed. In this section, we briefly describe how the WorkWeb agents work in cooperation.

Figure 6 shows a screen image of each worker's terminal[7]. On this screen, there are several personal agents' windows including his/her own personal agent's win-

[7]The original system has a Japanese GUI. Several texts on this screen are translated into English.

dow. A worker interacts with the other agents through this window.

A BPT agent is a workflow management agent. This agent is automatically generated when a workflow is invoked, that is, when a certain manager executes a workflow. The BPT agent negotiates with scheduler sub-agents of workers' personal agents, and accesses GIM, which collects data from other agents. For the BPT agent, GIM collects data on the workflow achievement and holds a "target value", which represents how many workflows should be completed in a certain period. The BPT agent, by getting the target number and completed number from GIM, calculates the priority for the workflow. If the difference between target and completed number is large, the priority for the workflow is rated high. Then the BPT agent schedules a plan to complete the workflow and distributes it to the appropriate personal agents. The scheduler sub-agent, that exists inside the personal agent, receives this plan and decides if the worker can achieve this task by the given deadline. If all of the answers from the scheduler sub-agents are "YES", the BPT starts executing workflow and proceeds step by step. If some worker can not achieve his/her task, the BPT agent re-plans the workflow. This is a dynamic workflow control function, but the details are not described here due to space limitations. Thus, with the cooperation of BPT, GIM, and scheduler sub-agents, group work represented with workflow is achieved.

Conclusion

In this paper, we have presented an office work support system with multiagents. Because people in an office concurrently engage in personal and group work, the system provides both personal and group work support functions. To support both personal and group work, a two-layered agent architecture, called the INA/LI architecture, was introduced. The first layer is the group work support layer, on which agents for group work, such as BPT agents, exist. The second layer is the personal work support layer, which exists inside the personal agent. The personal agent represents one human user and consists of several sub-agents. The sub-agents serve the user as software tool, such as an information filtering agent, to eliminate the work load of the user. A worker can expect to increase his/her productivity in both personal and group work with the WorkWeb system.

After coping with some problems, such as designing portable devices to enabling wearable computers, wireless communication infrastructures enabling location transparent agent communications, etc., we expect this architecture to be utilized to realize portable personal assistance systems like intimate computers(Weiser 1991; Lamming & Flynn 1994).

Acknowledgements

The authors would like to thank Mr. Masao Managaki, Mr. Hitoshi Miyai, and Mr. Toshiaki Miyashita for their helpful comments and discussions.

Windows NT, Visual C++, and Visual Basic are trademarks of Microsoft Corp. DELPHI is a trademark of Borland International, Inc. INA/LI and WorkWeb are trademarks of NEC Corp.

References

Bocionek, S. R. 1995. Agent systems that negotiate and learn. *International Journal of Human-Computer Studies* 42(3):265–288.

Cutkosky, M. R.; Engelmore, R. S.; Fikes, R. E.; Genesereth, M. R.; Gruber, T. R.; Mark, W. S.; Tenenbaum, J. M.; and Weber, J. C. 1993. Pact: An experiment in integrating concurrent engineering systems. *IEEE Computer* 26(1):28–38.

Finin, T., and Weber, J. 1993. *DRAFT Specification of the KQML Agent-Communication Language.*

Kauts, H.; Bart, S.; Coen, M.; Ketchapel, S.; and Ramming, C. 1994. An experiment in the design of software agents. In *Proceedings of the AAAI-94*, 438–443.

Lamming, M., and Flynn, M. 1994. "forget-me-not" intimate computing in support of human memory. Rank Xerox Europarc Technical Report EPC-94-103, Rank Xerox Research Centre.

Maes, P., and Kozierok, R. 1993. Learning interface agents. In *Proceedings of the 11th National Conference on Artificial Intelligence*, 459–465.

Object Management Group. 1993. *The Common Object Request Broker: Architecture and Specification.*

Tarumi, H.; Ishiguro, Y.; Tabuchi, A.; Yoshifu, K.; Kida, K.; and Asakura, T. 1996. An implementation of the workweb system. IPSJ SIG Notes, SIG-GW-15-22. (in Japanese)

Weiser, M. 1991. The computer for the 21st century. *Scientific American* September:66–75.

Yamaki, H.; Kajihara, M.; Nishimura, T.; and Ishida, T. 1995. Socia – an agent-based meeting support system. JSAI SIG Notes, SIG-FAI-9502-4.

Yamaki, H.; Kajihara, M.; Tanaka, G.; Nishimura, T.; Ishiguro, H.; and Ishida, T. 1996. Socia: Non-committed meeting scheduling with desktop vision. In *Proceedings of PAAM96*, 727–742.