An Agent-based Telecooperation Framework

Albrecht Schmidt, Alexander Specker, Gerhard Partsch,
Michael Weber, Siegfried Höck

Department for Distributed Systems
University of Ulm, Germany
aschmidt@hydra.informatik.uni-ulm.de, aspecker@hydra.informatik.uni-ulm.de,
gpartsch@informatik.uni-ulm.de, weber@informatik.uni-ulm.de,
hoeck@informatik.uni-ulm.de

Abstract. As the amount of information and communication increases dramatically new working environments must provide efficient mechanisms to maximize the benefits of these developments. In this paper a telecollaboration environment based on agent technology is proposed, which could be used as an information infrastructure for cooperative buildings or virtual enterprises.
The number of communication and information services increases rapidly in number and complexity. Therefore mediating components between users and services are required. In the environment being described it is suggested to deploy a personalized agent cluster for each user and network wide directory, broker, and trading services.
The agent cluster acts as a surrogate for the user in the system. In each cluster a variable set of personalized agents is aggregated according to the requirements of the user. Exemplarily the architecture and functionality of a communication agent as one part of an agent cluster is described.

Keywords. telecooperation, Internet, agent technology, personalized agents, computer supported cooperative work (CSCW), multimedia, communication, broker, trader

1 Motivation

Importance of Information and Communication. In the age of proliferating information and communication systems, information is becoming a production factor of great importance. Nearly all private and business processes rely more and more on the timely deliverance of information and data. Without intensive communication and cooperation these processes are hardly manageable. To accomplish this task an effective information, communication and collaboration environment is vital. Such an environment should be the foundation for cooperative buildings or other new forms of work environments, such as virtual enterprises, telecenters or -houses, or virtual (home-)offices.
Quality of Information. As the amount of information delivered to our homes or offices is becoming larger, the time we spend on reprocessing and filtering this extensive information is increasing to a hardly manageable extent. The goal is to extract the

useful and relevant information out of the available data in a cost and time effective manner.

Multiple Media. The media choice is traditionally not made by the receiving side but by the sender. The person that receives the information in this model has no straightforward way of influencing the choice of media and is therefore not able to make an optimal selection of the communication media according to her needs.

Synchronous Communication. Ideally all communication has a synchronous character. Delayed deliverance of messages should not be the result of the underlying transportation system but should be on purpose and adjusted to the users needs. The user wants to receive data not always as fast as possible contrarily he wants to be informed at that point in time that is best suited to help him solving his tasks (Maes 1997b).

Here synchronicity means, that messages are delivered to the user´s agent as fast as possible. If the user is not ready to receive the information, the agent holds messages back. If the agent itself is not reachable, due to network failures for instance, store and forward mechanisms in the framework are exploited.

Addressing the above mentioned problems we present in this paper an architecture and environment utilizing agents for personal communication and information in collaborative processes. These agents residing in the network act as intelligent, personalized and synchronously accessible partners of a real person. The agents optimize the communication and information process both of the individual person and the overall work process. Agents being specialized for different tasks form an agent cluster to represent their user in his various communication and collaboration tasks.

2 System Architecture

The system architecture of the telecooperation framework can be divided into two logical aspects interconnected closely with each other. In this section the first aspect the information exchange and coordination between the system components and the agents will be described. The following sections will explain the design and cooperation of the agents themselves. These agents are used to interact with the user, offer a homogeneous user interface and support the work between different users supported by external service components distributed over the network , see Fig. 1.

The agents can contact these services to use their additionally specialized functionality like teleconference scheduling or workflow management support. External services can offer more easily highly complex and efficient services than the agents are able to implement. With these distributed service components increased functionality needs are easily maintainable by updating and adapting these service components. As the main part of functionality resides inside the service components, the agent-side complexity can be lower and less task specific. Agents just have to be extended to implement an interface to a specific service component.

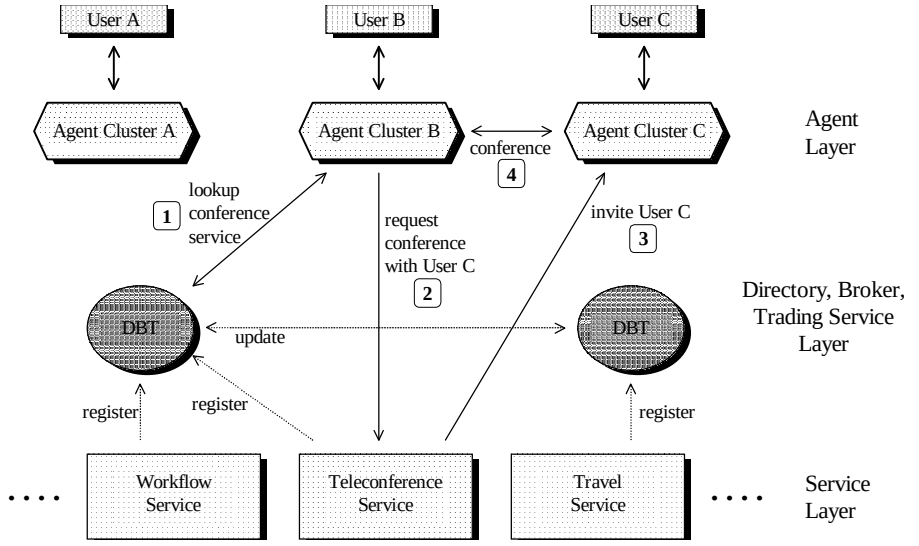

Fig. 1: System Architecture

Inside the service layer of the system all task specific knowledge is concentrated, this makes an efficient support of multiple user agents possible. Services such as Teleconferencing Service, Workflow Service, Meeting and Conference Scheduling Service, Database Service, Gateway Service and Electronic Marketplace Service can be offered in the network by third party service providers or by the IT-department of the user's company. Depending on the task the service can be provided by humans or intellignet systems. The system components in the telecooperation framework are designed flexible and open to integrate additional services at any time.

The directory service is used to deliver static information on service components and the location and addresses of agents, brokers and traders. The deployment of broker and trading services in a telecooperative environment as shown by Höck et al. (1997) is used here. The broker and trading functionality together with the above mentioned directory service serves as the location, negotiation and binding foundation of the presented telecooperation environment.

3 Agent Cluster and Functionality

To implement an optimal participation of each user in the cooperative process the agent paradigm is deployed. Each user is represented by an agent cluster in the system. In the cluster a variable set of agents is aggregated, see Fig. 2.

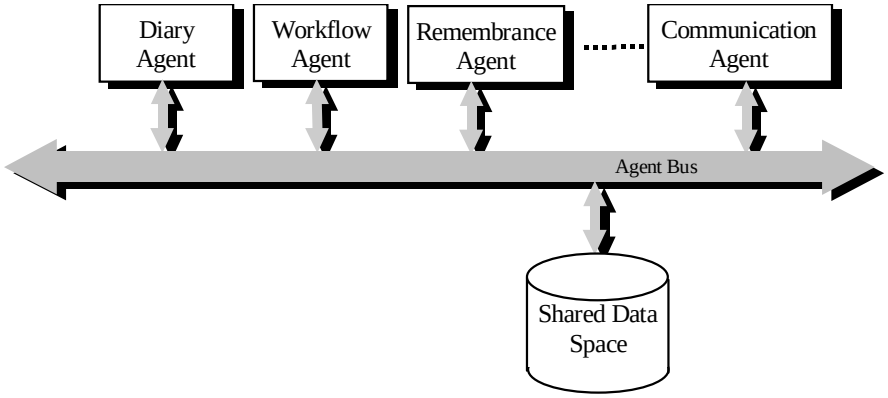

Fig. 2: Agent Cluster

Each agent has its own special task. E.g. the diary agent (DA) is keeping track of all appointments and makes the scheduling for the user. The workflow agent (WA) coordinates the interaction with other participants taking part in a workflow as well as with the workflow service. The remembrance agent (RA) is an active notebook to remind the user. A communication agent (CA) handles all communication on behalf of the user.

The main design goals for the agent cluster are described below. These features are of major importance to gain an improved productivity by using agents.

Personalization. The agents are personalized to the needs of the user. Agents have a certain knowledge about their users such as *she does not like meetings before 11am, he can not stand video conferences*, or *she prefers travelling by train*. The fact that agents are personalized give users the feeling that the agent is intelligent and caring.

Ability to learn about the user. To improve personalization it is necessary that the agent can learn likes and dislikes of the user by monitoring their behavior.

Synchronicity. Due to the fact that the agents are the surrogate of the user within the system, they are always accessible and therefore synchronous communication between agents of different users is always possible even if the user is not available.

Privacy. To provide privacy, and that is a very important issue when using personalized agents with extensive knowledge about their users, protocols are deployed that can determine who the partner is you are talking to and which information this partner should be able to see and which not.

Intra-cluster agent communication. The agents within a cluster can communicate with each other in an unrestricted way. Thus the user can be supplied with the information and communication at the right time and in the appropriate form. Through agent cooperation within the cluster the information load on the user can be reduced significantly. An agent bus (a software bus) and a shared data space, as shown in Fig. 2 is used to enable intra-cluster communication.
Inter-cluster agent communication. To perform their tasks agents must have the ability to communicate with other agents that belong to other users. As a general communication protocol the knowledge query and manipulation language, KQML described by Finin et al. (1993), is suggested for this purpose.

4 Agent Architecture and Functionality

The whole functionality to support the user is split up in separate entities each being supported by an agent. This way the complexity of a single agent is reduced. The architecture allows each user to compose his own set of agents in his cluster. The functionality of one agent results from the mechanisms provided by the agent, the interaction between agents, and from communication between agents and services.

4.1 The Communication Agent

To make it more concrete the communication agent is discussed in more detail in this section. This agent is deployed to handle all communication on behalf of a user. In Fig. 3 the architecture of the CA is depicted.

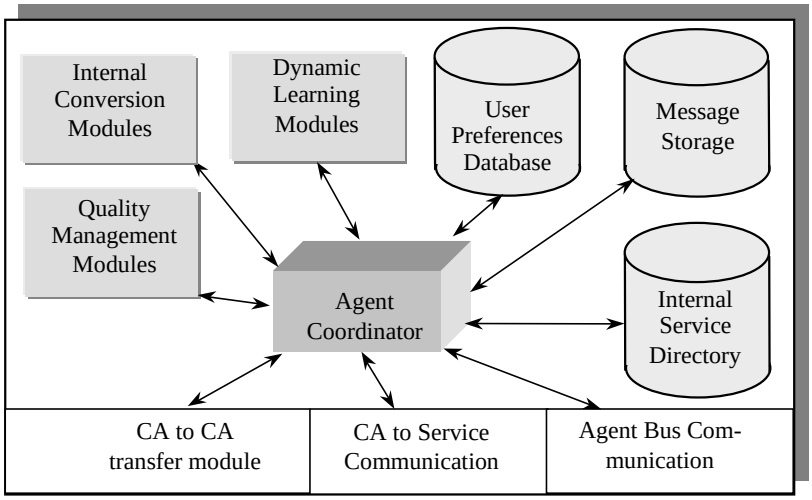

Fig. 3: Agent Architecture

The knowledge about the user is stored in the *User Preferences Database*. The *Dynamic Learning Module* is monitoring the behavior of the user and improves the personalization of the agent. Media conversion can be provided by the *Internal Conversion Modules* or by external services. Addresses and parameters of often used services are cached in the *Internal Service Directory*. Incoming and outgoing messages that could not be delivered yet are collected in the *Message Storage*. In the *Quality Management Modules* the logging and acknowledging of the communication streams is performed. The agent also provides different interfaces for communications to other agents and services.

The following features are of major importance to the developed communication agent.

Choice of media. In the proposed architecture the freedom to chose the communication media is given to the sender as well as to the receiver. This has the advantage that the user can always take the most appropriate way to communicate according to his situation.

Active agent. The agents are active components. The interaction can be triggered by the agent as well as by the user. Due to the fact that the CA has knowledge about the user and also communicates with other agents in the cluster the support for the user can be improved significantly.

Quality management. By using a synchronous CA it is possible to provide different levels of acknowledgement protocols. If the messages and the acknowledgements are authenticated by the senders the logging of the communication can be used for quality and history management or for legal issues.

Freshness of messages. To reduce the information load on the user a *best before* date is introduced into the messages.

Ownership of messages. Messages are owned either by the sender or by the receiver; they are never owned by the transport system. The system provides two general way to send messages. First the traditional way: the whole message is transmitted to the receiver and here the message is owned by the receiver after sending. The second way is to send only a link to the message. Then the message is still owned by the sender and can be modified or deleted till the receiver fetches the message.

5 Related Work

Agent technology and its usage in the Internet environment is quite popular in present research. It is beyond the scope of this paper to give a complete survey of agent technology, but the following selection provides some fundamentals of agent technology, describes a choice of research projects using agents in the web, and by this mentions work that inspired the described system.

At MIT Media Lab the software agent group is investigating agent technology in different projects (Rhodes 1996, Lashkari et al. 1994). Pattie Maes (1997b) states the following features of agents that are relevant for our work, too. “Software agents differ from current-day software in that they are

(1) proactive (taking the initiative to help the user by making suggestions and/or automating the more mundane tasks the user normally would have to perform),
(2) adaptive (learning the user's preferences, habits and interests as they change over time),
(3) personalized (customizing their assistance according to what they learned about the user), and
(4) autonomous (operating with minimal supervision on behalf of the users)".

Falchuk et al. (1997) at the University of Ottawa are working on mobile and intelligent agents targeting the problem of information retrieval and computer-aided instruction. A news gathering agent and a telelearning agent are introduced. They also address problems emerging with mobile computing such as address migration, heterogeneous networks, or portability (Ford et al. 1997). An agent based news filter is implemented as a prototype. Their work is mainly concentrated on resources located in the Internet. Due to the focus on mobile agents most of the work concentrates on asynchronous tasks.

The support of mobile users is targeted by researchers at ECRC. Chevalier et al. (1997) have developed a technology called Mobile Service Agents. They developed the idea to use an agent as a local representative while the user is not connected to the network. In this way interactive working can be done even when the user is temporarily disconnected.

Petrie (1996) analyzed the conflict between the client/server WWW paradigm and the peer-to-peer agent model. The usage of KQML-like agents and their compatibility with the web is discussed.

The concept of a software bus called InfoBus for JavaBeans is in the process of specification at Sun Microsystems (1997). The idea is to define a small interface which allows different components (JavaBeans) to exchange information in a structured way. The bus is proposed as an asynchronous and symmetric communication system between components without the need for a master component. The protocol is the controlling component. Corba also represents a middleware technology based on the software bus concept; Siegel (1996) shows an implementation of several interworking ORBs.

6 Conclusion

The described system supports the work of users by adaptive system functionality through the use of a triple layered service and agent infrastructure. In this paper the primary focus was drawn on the reduction of communication and information load. This is achieved by analyzing, extracting and optimizing the data flow in accordance with the situation and environment the user is in.

Users can access different services in a uniform way by employing their personal agents. These agents can interact with other agents or services. To locate and access services and agents in a heterogeneous environment brokers and traders are used. Due to this architecture agent-side complexity can be fairly low. Dynamic changes of

services and users, such as new access protocols, changing locations, and additional functionality are transparent.
The exemplarily demonstrated communication agent acts as a synchronously accessible surrogate of the user in the network. It provides the mechanisms for the conversion of different communication forms and therefore enables the sender to contact the receiver always in an appropriate manner, without imposing a common communication medium.

References

1. Höck, S., Weber, M.: *A Telecooperation Framework Based on Broker and Trading Services.* Proceedings of the 1st Workshop on Enterprise Distributed Object Computing, EDOC'97, Brisbane (1997)
2. Maes, P.: MIT Media Lab, Software Agents Group: Research (1997a) http://agents.www.media.mit.edu:80/groups/agents/research.html
3. Rhodes, B.: *Remembrance Agent: A continuously running automated information retrieval system.* Proceedings of the 1st International Conference on the Practical Application of Intelligent Agents and Multi Agent Technology, London, pp. 487-495 (1996)
4. Lashkari, Y., Metral, M., Maes, P.: *Collaborative Interface Agents.* Proceedings of AAAI '94 Conference, Seattle (1994)
5. Falchuk, B., Karmouch, A.: *A Mobile Agent Prototype for Autonomous Multimedia Information Access, Interaction and Retrieval.* Proceedings of Multimedia Modeling '97, Singapore (1997)
6. Ford, B., Karmouch, A.: *An Architectural Model for Mobile Agent-based Multimedia Applications.* Proceedings of Canadian Conference on Broadband Research '97, Ottawa (1997)
7. Chevalier, P.-Y., Thomsen, B.: *ECRC - Mobile Service Agents.* (1997) http://www.ecrc.de/research/dc/msa/
8. Petrie, C. J.: *Agent-Based Engineering, the Web, and Intelligence.* IEEE Expert. (Dec. 1996) http://cdr.stanford.edu/NextLink/Expert.html
9. Sun Microsystems. JavaBeans InfoBus draft specification (v0.04A, Aug. 21 1997) http://java.sun.com/beans/infobus/ibspec.html
10. Finin, T., Weber, J., et al.: Specification of KQML Agent Communication Language. DARPA Knowledge Sharing Initiative (1993)
11. Interview with Pattie Maes: Pattie Maes on Software Agents: *Humanizing the Global Computer*. IEEE Internet Computing July-August (1997b)
12. Siegel, J.: *CORBA: Fundamentals & Programming*, Wiley (1996)