Socialware: Multiagent Systems for Supporting Network Communities*

Fumio Hattori, Takeshi Ohguro, Makoto Yokoo, Shigeo Matsubara, and Sen Yoshida
NTT Communication Science Laboratories,
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02 JAPAN
{hattori,ohguro,yokoo,matsubara,yoshida}@cslab.kecl.ntt.co.jp

As the Internet continues to grow, social activities through the net will become more and more common. Although several systems exist for supporting communities over the net[1], these systems have limitations which will pose problems in the support of forthcoming network communities. To resolve problems, we are developing multiagent systems to assist in various social activities on network communities. We call these systems "*socialware*".

There seems to be three major issues in the support of network communities.

The first issue is how to bring people together, that is, how to link people with others and with communities, that share similar interests [1, 7, 8]. Search engines or directory services for communities can partially help, but more sophisticated systems are needed.

The second issue resolves around support for smooth communications [4], including support for visualizing and sharing common contexts[2], as well as for identifying the flow of conversations/discussions [5]. A lack of context and/or a failure in identifying the flow of a discussion may cause, for example, "flame wars", which unfortunately often happen in network communications. Another example is where a newcomer possibly has difficulty understanding what is going on in a community.

The third issue is finding what relationships people have [3]. This includes how to identify the objectives/roles of communities and individuals. Since network communities are amorphous compared to organizations like companies and schools, such relationships and networks are sometimes hidden or unknown. This causes difficulty in finding the "key" person or people suitable for the current concerns.

These issues also exist in real communities, but the problems are more obvious in network communities since they are not constrained by restrictions of time and place. Providing a new communication environment [6] alone is not sufficient for resolving these problems. Active support using agent technology, which is based on interaction between software agents, and between humans and software agents, seems important and necessary. We believe that socialware is a promising area for multiagent applications.

[1] *CommunityWare* (`www.communityware.com`) or *Firefly* (`www.firefly.net`) can be viewed as typical examples of "community providers".

[2] The term "context" here includes such aspects as the depth of knowledge of individuals, their attitudes towards the subjects, and mutual knowledge and manners (ethics) for the community.

*This is an accepted version. Definitive version will be published on C.ACM 1999 March. Will also be found in ACM digital library.

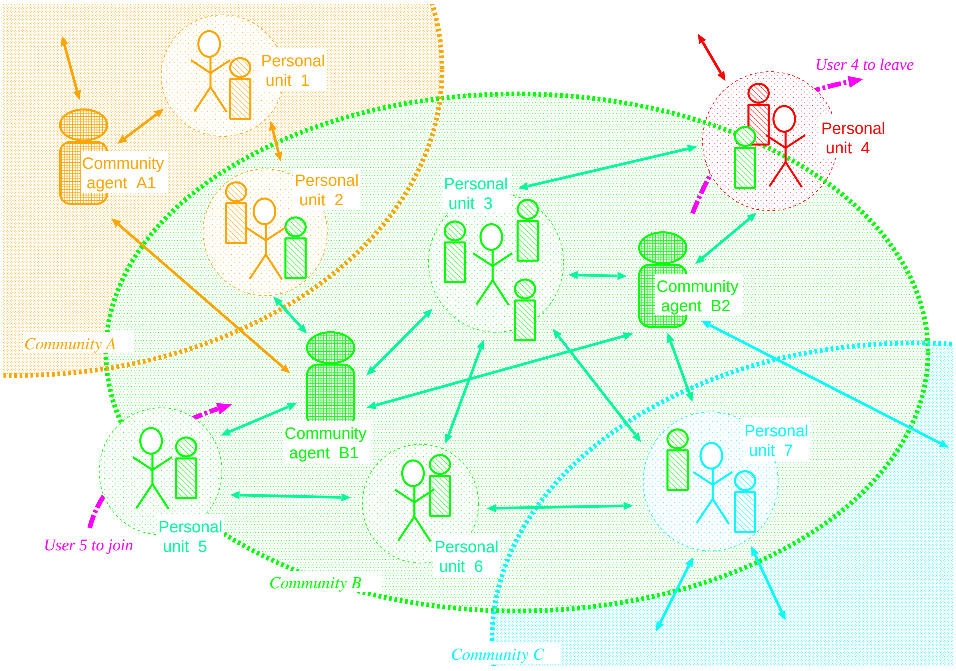

Figure 1: A general architecture of socialware as a multiagent system

Socialware as Multiagent Systems

There are several characteristics specific to network communities, which make a multiagent architecture attractive to use.

The first characteristic is that the participants of a network community are widely distributed and the number of potential participants is large. Hence, no solid, centralized, or monolithic system would be adequate: A distributed system would be required, in which personalized agents for each participant cooperate with each other.

The second characteristic is that communities have a dynamic nature. In each community, the active membership will change over time, in addition to the roles of individuals and objectives, and moreover, the community will likely change its aspect. In other words, there exists no fixed organization nor a clear goal for a network community. This characteristic contrasts the area of groupware, which helps people already organized work cooperatively, where the members, their roles, and their objectives are rather clearly defined.

The third characteristic is that the individuality of each member is preserved even in one community. That is, each member can have diverse objectives, even if all members share common interests in general. Furthermore, people can be members of several communities at the same time depending on their various interests. Hence, supports need to be personalized to adapt individual objectives and interests. They also need to adapt to the variations and changes of interests and activities of individuals.

Based on the observations above, we propose a general architecture of socialware (Figure 1).

We view a community as a collection of personal units, community agent(s), and the set of relations between them. A personal unit consists of a user and his/her personal agents. Each personal agent can help the user by, for example, gathering and exchanging information, visualizing contexts, and recommending/assisting the user in making a choice, all in personalized ways. All of the agents cooperate and act as a unit, with the user being the central figure. The community agents have the function of providing shared information, knowledge, and/or contexts within the community and act as mediators for informal communications between people.

Having an architecture where each user has personal agents that communicate with each other, enables the community to spread over the net.

In a personal unit, it is possible for some agents to be domain specific (e.g., an information retrieval agent specialized to financial news), and others more generic (e.g., an interface agent for navigating and reading documents). Adaptation to multiple aspects of a user and the user's changes in interests can be achieved by changing the system dynamically and autonomously. For example, a domain specific agent can clone itself to produce a new agent and make additional communication channels when the user's interest has changed. This architecture also enables the user's individuality to be maintained: When the user belongs to several communities simultaneously, the user may have multiple domain-specific agents for each community.

The whole system needs to be flexible and adaptive so that it can adapt to the dynamic nature of a community where the community members, as well as their relationships, will likely change over time.

Experimental Systems of Socialware

CommunityOrganizer

We have developed a prototype application for the purpose of "linking people", which we call the "*CommunityOrganizer*" [8]. The system consists of personal agents for each user and a community agent. Each personal agent has functions to acquire the user profile and to visualize potential communities around the user. The community agent has functions to collect the user profiles and to maintain the information on potential communities.

Upon a request from a personal agent, the community agent first computes potential communities around the owner of the personal agent, and then sends the necessary data (users in the potential communities and relevances among them) to the personal agent. Next the personal agent locates users (indicated by icons) in a two-dimensional space, where the distances among the users reflect the relevances (degrees of common interest) among them. In any view-space, the owner of the personal agent is located at the center.

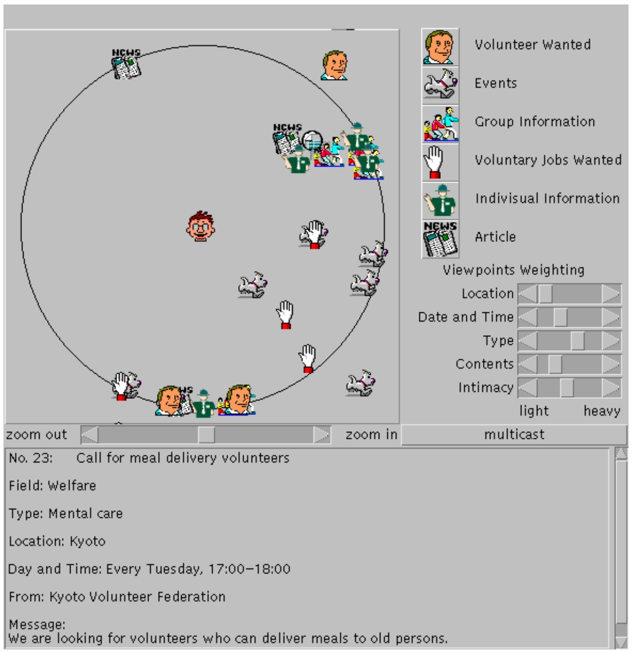

Figure 2: A screen-shot of the CommunityOrganizer through a personal agent

The relevances between users are calculated by the community agent from the users' profile data. These profiles can be obtained either from each user's input, from archives of mailing lists using keyword extraction techniques, or from information on the users published on the Web using information retrieval techniques (The system acquires profile data automatically when the URL for a user's "home page" is given).

Each personal agent has slidebars, which adjust the weightings of the viewpoints temporarily, since the degree of common interest consists of multiple aspects. These include such aspects as location, date and time, type, content, and intimacy, in the domain of voluntary activities for example (Figure 2). When the user changes the weightings of the viewpoints using the slidebars, the personal agent automatically re-calculates the appropriate locations of the other users using a weighted cosine measure, and re-display them.

For further personalization of the view according to the user's interests, the personal agent can learn the weightings from user feedback. When the user finds that a person is located at an inappropriate position (i.e., too close to or too far from the user), the user can indicate this fact to his/her personal agent. Then, the agent can modify the weights of keywords permanently to adjust the relevance value between the indicated person and the user.

When people or potential communities with similar interests are found, the user can multicast a message to them. The multicast area is basically within a certain distance from the user, but can be adjusted by the user. Furthermore, users can exchange their viewpoints (weightings) through the community agent. If a viewpoint received is attractive, a user can incorporate it into his/her own weightings. This function is useful in forming a new community, since the viewpoints exchanged can make the individuals' interests more clear, and can lead to common interests shared among the users.

In this way, the system enables users to find potential communities around them, to simulate several possibilities of potential communities, and to form new communities.

CommunityBoard

We have developed a prototype application for visualizing the structures of discussions, called the "*CommunityBoard*" [5]. The system consists of personal agents for each user and a community agent, all of which are totally written in Java. Each personal agent displays the structures of discussions according to the user's interests (Figure 3). The community agent classifies messages according to several criteria like topic, time, reputation, and so on, and provides a shared information-base in the community.

Three main dimensions are chosen to express the structure of a discussion: person, topic, and importance. In an integrated view-space of the personal agent, each message is represented as an icon. Person, topic, and importance are represented by the type, position, and shading of the icon, respectively.

Person. Each person has his/her own icon

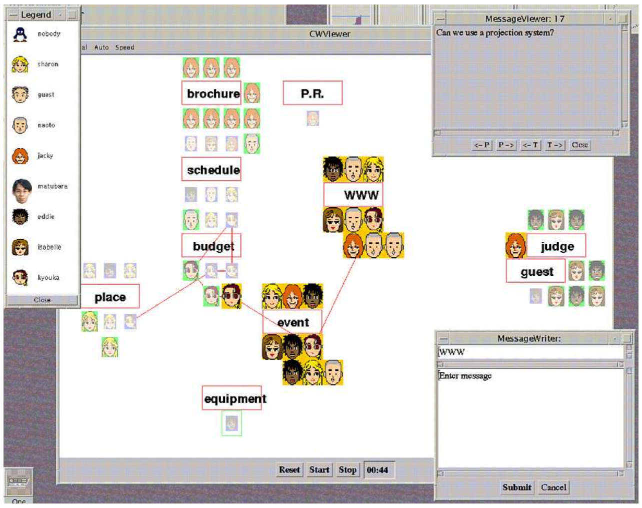

Figure 3: A screen-shot of the viewspace for the CommunityBoard

for messages.

Topic. Each message is located according to its topic, with similar topics located nearby. The topic for a message is currently given by the writer of the message as a keyword (We plan to allow multiple keywords or to extract keywords automatically from messages). The relevances between topics are calculated using a database on the community agent. Each personal agent decides the locations of the topics from the relevances using a multidimensional scaling method. The locations are recalculated whenever a message with a new topic arrives.

Importance. Icons for messages become dimmer and smaller as the importance decreases. The importance is a function of time, reputation, and interest for topics of messages. Interests for topics as well as the importance function itself can be adjusted by the user. Reputations of messages are calculated by the community agent using collaborative filtering techniques.

Users can take in the various combinations of these three aspects at a glance, without changing the viewspace itself.

Such information can be used in several ways. A user can decide whether to join a discussion, or can guess to whom a question relating to a topic should be asked, from the information about persons and topics. Users can also grasp the relations between topics, what topics have been discussed (or not discussed), and what topics are currently important, from the information about topics and importance. Such knowledge helps the users learn the current context of the discussion and helps them anticipate responses beforehand, e.g., before they join the discussion. Furthermore, the users can better understand the transitions between topics from the information about persons and importance. This helps them understand other users and the causal relations between their messages.

In this way, users can obtain and share the contexts of a community. This helps smooth communications between people, helps in revealing relations between people, and helps in finding the role of each person.

Future Direction

Socialware is a new concept for network community support [2]. We believe that it opens up new areas of application for a multiagent framework. These areas include

- modeling the behaviors of communities,
- structuring communities,
- enabling flexible communication channels,
- learning and collaboration of users and agents, and
- mediating public information spaces and personal spaces, including balancing between privacy and open opportunity.

There are several technical challenges in implementing socialware as a multiagent system. One is how to make a coalition of agents. Another challenge is how humans and agents can effectively collaborate as a system.

References

1. Foner, L.N. A multi-agent referral system for matchmaking. In *Proceedings of PAAM'96* (Apr. 22-24, London,UK). The Practical Application Company, Lancashire, 1996, pp. 245–261.
2. Ishida, T. Towards CommunityWare. In *Proceedings of PAAM'97* (Apr. 21-23, London,UK). The Practical Application Company, Lancashire, 1997, pp. 7–21.
3. Kautz, H., Selman, B. and Shah, M. Referral Web: Combining social networks and collaborative filtering. *Commun. ACM* 40,3 (Mar. 1997), 63–65.
4. LaLiberte, D. and Woolley, D. Presentation Features of Text-based Conferencing Systems on the WWW. *Computer-Mediated Communication Magazine* 4,5 (May 1997).
5. Matsubara, S., Ohguro, T. and Hattori, F. Community Board: Social Meeting System able to Visualize the Structure of Discussions. In *Proceedings of Knowledge-based Intelligent Electronic Systems (KES'98)* (Apr. 21-23, Adelaide,Australia). IEEE, Piscataway, pp. 423–428.
6. Nakanishi, H., Yoshida, C., Nishimura, T. and Ishida., T. FreeWalk: Supporting casual meetings in a network. In *Proceedings of CSCW'96*, (Nov. 16-20, Boston,Mass.). ACM, New York, 1996, pp. 308–314.
7. Nishimura, T., Yamaki, H., Komura, T., Itoh, N., Gotoh, T. and Ishida, T. Community Viewer: Visualizing community formation on personal digital assistants. In *Proceedings of IJCAI'97 Workshop on Social Interaction and Communityware* (Aug. 23-29, Nagoya,Japan). Morgan-Kaufmann, San Francisco, 1997, pp. 25–30.
8. Yoshida, S., Kamei, K., Yokoo, M. Ohguro, T., Funakoshi, K. and Hattori, F. Community Visualizing Agent. In *Proceedings of PAAM'98* (Mar. 23-25, London,UK). The Practical Application Company, Lancashire, 1998, pp. 643–644.