RC 21895 (98513) 30 November 2000 Computer Science

IBM Research Report

Smart Distance and WWWaware - A Multi-Agent Approach

Yiming Ye, Stephen Boies, Paul Huang
IBM Research Division
Thomas J. Watson Research Center
P. O. Box 704
Yorktown Heights, NY 10598

John K. Tsotsos
Dept. of Computer Science
York University
Toronto, Ontario
Canada M3J 1P3

Research Division
Almaden - Austin - Beijing - Haifa - T. J. Watson - Tokyo - Zurich

Smart Distance and WWWaware – A Multi-agent Approach

Yiming Ye, Stephen Boies, Paul Huang
IBM T.J. Watson Research Center
PO Box 704
Yorktown Heights, NY 10598
1 (914) 784-7455
yiming@watson.ibm.com

John K. Tsotsos
Department of Computer Science
York University
Toronto Ontario Canada M3J 1P3
1 (416) 736-2100
tsotsos@cs.yorku.ca

ABSTRACT

In this paper, we propose the concepts of a smart distance and an awareness network in a distributed collaborative environment. We illustrate the architecture of an Agent Mediated Collaborative system - the Agent-Buddy system that can create a sense of group presence and at the same time preserve the privacy of each user. Virtual springs systems are used to model the awareness degrees among team members. Each agent makes decisions by considering multiple factors. The goal of the multiagent team is to minimize the global awareness frustrations with respect to different kinds of tasks. Empirical studies have been conducted to analyze the influence of individual behavior on global performance for various kinds of tasks.

Keywords

Multiagent System, Awareness Network, CSCW, User Interface

1. INTRODUCTION

The worldwide nature of today's market has forced many companies and institutions to de-centralize their organizational structures. Furthermore, more and more people will be working from home. With ubiquitous connectivity on the horizon, collaborative computing promises to become one of this century's core applications. People will be more and more involved in collaborative computing because of the pressure from companies to improve their product-development and decision making process and because of the convenience brought by the information super-highway.

There are four modes conceptualized by researchers in Computer Supported Collaborative Work (CSCW) on how people work [10] in a collaborative environment. Synchronous mode refers to the situation in which activities occur at the same time and in the same place; distributed synchronous mode refers to the situation in which activities occur at the same time but at different places; asynchronous mode refers to the situation in which activities occur at different times in the same place; and distributed asynchronous mode refers to the situation in which activities occur at different times and places. This paper concentrates on the application of agent and multi-agent technologies to group work in a distributed synchronous nature.

Many computer systems support simultaneous interaction by more than one user. However, most of them support multi-user interaction in a way that prohibits cooperation – they give each user the illusion that he or she is the only one using the system. To support and encourage cooperation, cooperative applications must allow users to be aware of the activities of others. The purpose of providing cooperative awareness is to establish and maintain a common context and to allow the activities or events associated with one user to be reflected on the other users' screens. For example, Lotus Sametime [5] is a family of real-time collaboration products. It provides instant awareness, communication, and document sharing capabilities and brings the flexibility and efficiency of real-time communication to the business world. The cornerstone of Sametime is awareness. With awareness of coworkers, partners, or customers online, users can communicate in a variety of ways. However, a direct reflection of all the activities of co-workers on users' screens is not practical. The first reason is that it wastes communication bandwidth, especially when users are far apart and the amount of data to be transmitted, such as video data, is huge. The second reason is that many users may not like the situation that all of his or her activities are broadcast to all the other members of the team. The third reason is that each user is concentrating on his or her work and does not have the energy and motivation to monitor every movement of other users. Thus, it is critical for a collaborative computing system to analyze activities of a given user, detect that user's important events, but show only the information necessary to other users.

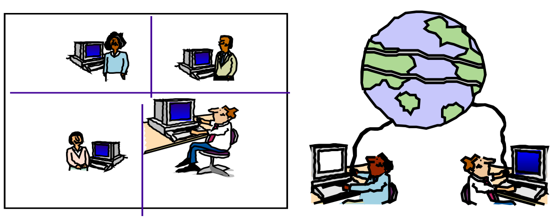
When more and more people are working in a distributed cooperative environment, especially when more and more people are working from home, the requirement of staying aware of co-workers' status and activities will become increasingly important. Parallel with the advances made in CSCW in recent years, there have been interesting developments in the fields of Intelligent Agents and Distributed Artificial Intelligence, notably in the concepts, theories and deployment of intelligent agents as a means of distributing computer-based problem solving expertise. The concept of intelligent agents has given rise to an exciting new technology of wide-potential applicability. In particular, the paradigm of multi-agent systems forms a good basis for the design of CSCW architectures, and the support of CSCW operations. Intelligent agents that can undertake sophisticated processes on behalf of the user and dynamically and intelligently adjust the "distances" among co-workers will be a necessary part of any organization's virtual structure. The digital multi-agent organization will capture the dynamics of teamwork, adjust the awareness level among co-workers, and re-shape the form and characteristics of collaborative work. The automation brought by this virtual organization will dramatically reduce certain types of frictional costs. On a larger scale, it is our belief that in the future, the WWW will not only be the knowledge pool of human society, but also be the digital world where people can meet and sense each other.

The remainder of this paper is organized as follows. The next section proposes the concept of smart distance. Section 3 describes the architecture of Agent-Buddy - an agent mediated CSCW system that provides an adaptive awareness among co-workers. Section 4 defines the concept of an awareness network, which is a key concept behind Agent-Buddy. Section 5 details the mechanism of adaptively adjusting the awareness levels in Agent-Buddy. Section 6 empirically studies the influences of agents' behaviors on global performances with respect to different kinds of tasks. Section 7 presents brief conclusions.

2. SMART DISTANCE

People are separated by distance and they like to adjust it when there are choices. For example, when working at the same table, the two persons in Figure 1(a) are quite close, while the two persons in Figure 1(b) are not so close. In Figure 1(c), physical rooms are built to separate co-workers. Technologies, however, can bring distant people closer, as shown in Figure 1(d).

(a) (b)



(c) (d)

Figure 1: Illustration on distances

Distance, as an abstract concept here, refers to the degree of objective difficulties in sensing other people through taste, touch, smell, hearing, and sight. Physical distance, as determined by the geometrical distance of body centers, is one of the major factors that determine the distance between people. However, it is not the only factor. Environment also contributes to the sense of distance. For example, occlusions can increase the difficulties in sensing thus increasing the distance.

Technologies can provide more communication channels and thus shorten the distance. In a two-person telephone conversation scenario or video conferencing scenario, the distances between people are made much shorter because they can hear or see each other. However, these distances are still bigger than the scenario in which they are in the same room.

Smart distance refers to the situation where people intelligently adjust their distance based on various social contexts and preferences. For example, Figure 1(a) and Figure 1(b) show the social context where two persons adjust their distance by attitudes and by physical distances. As a matter of fact, distance adjusting appears in almost all social activities and working environments. A company brings people to work at the same location; however, it also allocates people to different rooms (Figure 1(c)).

Technology has now advanced so that a rich choice of distances is available. With ubiquitous connectivity on the horizon, and as more and more people work at the same time from different places, the issues of how to design the virtual organization and how to automatically adjust distances among people will become more and more important. The project "Smart Distance and WWWaware" is an effort along this line. Our goal is to build a multi-agent system called "Agent-Buddy" that can automatically detect different events associated with co-workers and can intelligently adjust awareness levels among co-workers. In this paper, we concentrate on the smart distance aspect of Agent-Buddy and study the influence of individual agent behaviors on the global team performance with respect to different kinds of tasks.

3. THE ARCHCHITECTURE OF AGENT-BUDDY

Software agents are studied from two complementary perspectives. The first views software agents as entities with different skills and knowledge within a larger community of agents [9]. Each agent is independent or autonomous. It may accomplish its own task or cooperate with other agents to perform a personal or global task. The second approach concentrates on the necessity for agents to interact with users at the level of the interface [7] (Laurel). The critical points here are how agents can understand the needs and goals of the user, how agents should behave, and how agents' behaviors can be perceived by the user.

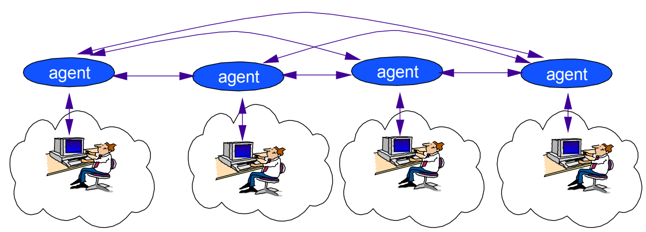


Figure 2. The architecture of Agent-Buddy

The Agent-Buddy approach is a combination of the above two approaches. Figure 2 shows the architecture of the Agent-Buddy system. The goal of an agent in Agent-Buddy is to perceive events or status associated with one user and to selectively provide the perceived information to other users of the team. The Agent-Buddy system can be added to any CSCW system or virtual organization system to enhance the sense of working "together" concurrently and, at the same time, keep the privacy of each user.

An agent in Agent-Buddy is a computational system that inhabits dynamic collaborative environments. It has knowledge about its own user and about the conventions of the working group. This knowledge can be used to guide its interactions with its responsible user and other agents of the group. The goal is to make the collaborative work easier and more efficient for members of the working group. Figure 3 shows modules within an agent. The User Interface Module is responsible for obtaining input from the user and input from various devices. It is also responsible for performing dialogues between the user and the agent, such as delivering the negotiation results to the user. The Event Perception Module analyzes input from various devices within a user's working environment and detects events. The Knowledge Base Module contains the knowledge of the agent on the user and the group. This knowledge includes appointment schedules of the user, preferences of the user, and the relative importance of different group members, etc. The Plan Generation module generates action plans for the user by combining the user's request, the knowledge from Knowledge Base, the events detected by the Event Perception Module, and the requests of agents of other users delivered by the Negotiation Module. The Negotiation module is used to perform communications and negotiation dialogues with agents of other users. XML [2] is used to encode messages among agents.

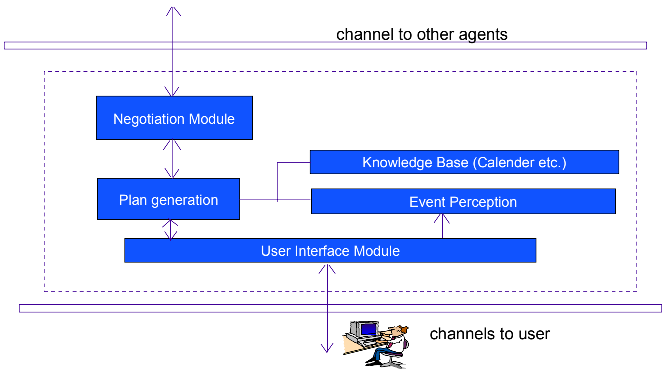


Figure 3. The modules within an Agent

There are two important features of the Agent-Buddy system. One is the event perception ability of each agent. The other is the automatic distance adjusting ability of the agent network. Each agent can perceive events with respect to its user based on signals perceived by all the devices within the user's environment. We have proposed a method that uses eigen-space and the eigen-pyramid to perform events for agents [16]. In this paper, we concentrate on distance adjustment.

To create a sense of group work, each agent has an interface to display events or live video and audio associated with other users. Events associated with a user can be whether the user is logged on, how frequently the user is typing on the keyboard, what program the user is running, whether the user is entertaining himself by browsing the Internet, whether the user is working on the project, whether the user is on the phone and who the user is talking to, whether the user is happy, sad or simply normal, whether the user has a visitor, and even whether the user needs a break because he is not being efficient at all, etc. However, an agent is not able to display all events of other users. There are two major concerns here. The first and the most important one is that the agent must communicate with agents of other users and must ask for permission to access events detected by those agents. It is up to other agents to decide what should be revealed to the asking agent. For example, events that intrude upon privacy cannot be accessed. For different asking agents, the criteria will be different. The second concern is that an agent should not display all the events of other users because it is usually unnecessary and impossible to do this within a single screen or through a multi-model interface. An agent must intelligently select events to display for the benefits of its user.

4. AWARENESS NETWORK

We use an *awareness network* to represent the awareness status provided by agents in the Agent-Buddy. An awareness network is a complete directed graph *G=(V,D)*. Where *V* is the vertex set of *G* and *D* is the edge set of *G*. Each element $v \in V$ corresponds to an agent in Agent-Buddy. For any two vertex v_i and v_j, there exist direct links d_{ij} and d_{ji} (Figure 4). The link d_{ij} gives the distance from user *i* to user *j*, or in other words, it give the degree of difficulty for user *j* to perceive the activities of user i. It is a measurement of the amount of information about user *i* that is exposed to user *j*. The more the information is exposed, the smaller the value of d_{ij}. This value is selected by agent *i* by considering various factors. Similarly, d_{ji} gives the distance from user *j* to user *i*. Please note that in many situations $d_{ij} \neq d_{ji}$.

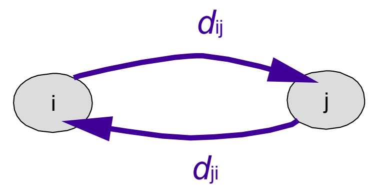


Figure 4. Agents *i* and *j*, and links d_{ij} and d_{ji} between them.

The values on the links of *G* are not constants; they keep updating at different times because of various factors such as the current tasks and the current events. Thus d_{ij} is a function of time. The awareness matrix,

$$\Psi(\tau) = \begin{pmatrix} d_{11}(\tau).....d_{1N}(\tau) \\ \vdots \\ d_{N1}(\tau).....d_{NN}(\tau) \end{pmatrix},$$

gives the awareness status of the Agent-Buddy time instant τ, where *N* is the total number of users in the system.

For convenience, throughout the rest of this paper, we will use "agent *i*" to refer the "agent of user *i*", and use "the distance from *i* to *j*" or "the distance from agent *i* to agent *j*" to refer to the "distance from user *i* to user *j*".

5. THE DETERMINATION OF DISTANCE

5.1 The Springs Potential Energy Analogue

We propose a physics-based framework for each agent to determine its distances to other agents, or in other words, the amount of information to expose to other users. This framework features dynamic models that incorporate various factors that must be considered.

Our idea comes from the elastic potential energy in physics. As is well known, if there is no force applied to a spring, the spring will be at its equilibrium position. However, if there are either compression or stretching forces applied on a spring, the spring will be deformed. The energy used to change the spring's displacement is stored in the coils as elastic potential energy. Most springs demonstrate a linear relationship between displacement from their natural positions and the applied forces and satisfy Hooke's Law: *F=-kx*, where *x* is the displacement and *k* is a constant that measures the stiffness of a spring. The elastic potential energy is given by $E = \frac{1}{2}kx^2$. When Hooke's Law is not satisfied, then the value of *k* will be a function of *x* and the potential energy can be given by: $E_p = \int_0^x k(t)tdt$. Now, let us consider a physical system as shown in Figure 5. Figure 5(A) shows equilibrium positions of the springs. Figure 5(B) shows the situation when a horizontal massless flat plate is applied to this system. Each spring is connected to the plate. The potential energy of the springs system is given by:

$$E(x) = \int_0^{x-x_1} k(t)tdt + \cdots + \int_0^{x-x_n} k(t)tdt \ .$$

Where $x_1, \cdots, x_n$

cooperative with respect to user *j*. The weight w_{ij}^{task} gives the degree to which user *i* emphasizes a collaborative task. The higher the value of this weight is, the more collaborative agent i is.

Agent i will select the distance that minimizes $\delta(y)$ as the distance from i to *j:* $d_{ij}(\tau)=y^*$, such that $\forall y, \delta(y^*) \le \delta(y)$. If all the virtual springs satisfy Hooke's law, we have:

$$y^* = \frac{w_{ij}^i k_{ij}^{i,e_v} d_{ij}^{i,e_v} + w_{ij}^{str} k_{ij}^{str} d_{ij}^{str} + w_{ij}^j k_{ij}^j d_{ij}^j + w_{ij}^{task} k_{ij}^{t_q} d_{ij}^{t_q}}{w_{ij}^i k_{ij}^{i,e_v} + w_{ij}^{str} k_{ij}^{str} + w_{ij}^j k_{ij}^j + w_{ij}^{task} k_{ij}^{t_q}}$$

5.3. The Multi-channel Nature of Distance

In the above discussions, we assume that the stiffness function $k(x)$ is a monotonous function with respect to a single-variable measurement of distance x. In most situations, however, distances are of multi-channel natural and may not be measured just by one single variable. For example, John and Mary are working in different places. Suppose that there are three ways for John to know Mary's activities: (a) John can watch Mary's activities through only a video camera installed at Mary's office, the quality of the video can be adjusted; (b) John can listen to Mary's activities through only an audio device installed at Mary's office, the quality of the audio signal can be adjusted; and (c) John can watch and listen to Mary's activities through both the above mentioned audio and video devices. It is easy for us to see that the distance from Mary to John for situation (c) is closer than that of situation of (a) or that of situation of (b), when the qualities of video are the same for all the situations and when the qualities of audio are the same for all the situations. However, it is a much more difficult job for us to compare the distances for situations of (a) and (b). We cannot really answer whether the distance for situation (a) is closer or further than that for situation (b), because they are coming from two different channels. Similarly, when the audio and video qualities are not constants, it is also difficult to compare two situations in situation (c). Thus, in this example senario, distance d should be measured by two channels, $d=d(a,v)$. Variable a refers to the quality of the audio signal. The higher the quality of the audio signal, the bigger the value of a. Variable v refers to the quality of video signal. The higher the quality of the video signal, the bigger the value of v. A pair $<a,v>$ defines a communication setting from Mary to John. All the different pairs of $<a,v>$ determine the total possible communication settings from Mary to John. For any two different settings, distances may be comparable or may not be comparable, however, potential Back-to-Ideal energies can always be calculated because the difference in distances can always be obtained. Suppose we have two settings $<a_1,v_1>$ and $<a_2,v_2>$. If $a_1<a_2$ and $v_1<v_2$, then we have $d(a_1,v_1)>d(a_2,v_2)$. If $a_1=a_2$, then the function $d(a_1,v)$, or $d(a_2,v)$, is a monotonous decreasing function with respect to variable v. Similarly, if $v_1=v_2$, then the function $d(a,v_1)$, or $d(a,v_2)$, is a monotonous decreasing function with respect to variable a. On the other hand, if $a_1>a_2$ and $v_1<v_2$, then we are not able to determine which distance is closer as there is no way to compare signals coming from two different channels. We, however, are able to calculate the Back-to-Ideal potential energy for any situations. Suppose $d(a_1,v_1)$ is the ideal distance from Mary to John with respect to John and $d(a_2,v_2)$ is the actual distance selected by Mary. If we imagine that there is a virtual spring within each channel, then the Back-to-Ideal energy is the sum of frustrations cause by both audio and video and can be calculated by: $\int_0^{a_2-a_1} k_a(x)xdx + \int_0^{v_2-v_1} k_v(x)xdx$, where k_a and k_v are stiffness functions for audio and video with respect to John. The differences in k_a and k_v reflect the relative importance of audio and video in John's mind. In general, different channels have different stiffness functions for a given person. For the same channel, different people may have different stiffness functions.

In general, we need to first figure out how many channels a peer-to-peer communication can have, and then determine all the relevant stiffness functions involved. The channels should be selected such that for a given channel, when the qualities of signals from all the other channels are fixed, the distance function should be a monotonous decreasing function with respect to the quality of the signal of the given channel. In the multi-channel scenario, a distance is no longer specified by a single variable as we did in Section 5.2. Instead, a distance is specified by a set of variables that give the qualities of signals from all the channels. Suppose that there are totally z different channels. Let d_{ij}^{i,e_v,c_r} be the ideal quality of the signal for channel r from user i to user j with respect to user i under the situation e_v and let k_{ij}^{i,e_v,c_r} be the corresponding stiffness function. Then the Back-to-Ideal potential energy from i to j for user i with respect to signal quality setting $\langle s_1,\cdots,s_z\rangle$ is given by:

$$\delta_1(\langle s_1,\cdots,s_z\rangle) = \int_0^{s_1-d_{ij}^{i,e_v,c_1}} k_{ij}^{i,e_v,c_1}(x)xdx + \cdots + \int_0^{s_z-d_{ij}^{i,e_v,c_z}} k_{ij}^{i,e_v,c_z}(x)xdx .$$

Similarly,

$$\delta_2(\langle s_1,\cdots,s_z\rangle) = \int_0^{s_1-d_{ij}^{str,c_1}} k_{ij}^{str,c_1}(x)xdx + \cdots + \int_0^{s_z-d_{ij}^{str,c_z}} k_{ij}^{str,c_z}(x)xdx ;$$

$$\delta_3(\langle s_1,\cdots,s_z\rangle) = \int_0^{s_1-d_{ij}^{j,c_1}} k_{ij}^{j,c_1}(x)xdx + \cdots + \int_0^{s_z-d_{ij}^{j,c_z}} k_{ij}^{j,c_z}(x)xdx ;$$

$$\delta_3(\langle s_1,\cdots,s_z\rangle) = \int_0^{s_1-d_{ij}^{t_q,c_1}} k_{ij}^{t_q,c_1}(x)xdx + \cdots + \int_0^{s_z-d_{ij}^{t_q,c_z}} k_{ij}^{t_q,c_z}(x)xdx .$$

Where the term $\delta_2(\langle s_1,\cdots,s_z\rangle)$ gives the Back-to-Ideal potential energy from i to j from organizational structure point of view, when the signal quality setting is $\langle s_1,\cdots,s_z\rangle$. The term $\delta_3(\langle s_1,\cdots,s_z\rangle)$ gives the Back-to-Ideal potential energy from i to j for user j, when the signal quality setting is $\langle s_1,\cdots,s_z\rangle$. The term $\delta_4(\langle s_1,\cdots,s_z\rangle)$ gives the Back-to-Ideal potential energy from i to j from the point of view of current task, when the signal quality setting is $\langle s_1,\cdots,s_z\rangle$. The functions $k_{ij}^{str,c_1},\cdots,k_{ij}^{str,c_z}$, $k_{ij}^{j,c_1},\cdots,k_{ij}^{j,c_z}$, $k_{ij}^{t_q,c_1},\cdots$, and $k_{ij}^{t_q,c_z}$ are the corresponding stiffness functions. Suppose that for channel c_r ($r=1,\cdots$, or z), there are totally z_r ($r=1,\cdots$, or z) different signal qualities. Then the total number of different signal quality settings is given by $M=z_1\times\cdots\times z_z$. Each setting corresponds to a distance.

Similar to Section 5.2, to determine the final distance, or the final choice of signal quality setting, agent i uses a weighted sum of

Back-to-Ideal potential energies of the above factors as its objective function:

$$\delta(\langle s_1,\cdots,s_z\rangle)= w_{ij}^{i}\times\delta_1(\langle s_1,\cdots,s_z\rangle)+w_{ij}^{str}\times\delta_2(\langle s_1,\cdots,s_z\rangle)+w_{ij}^{j}\times\delta_3(\langle s_1,\cdots,s_z\rangle) +w_{ij}^{task}\times\delta_4(\langle s_1,\cdots,s_z\rangle).$$

Where the weights w_{ij}^{i}, w_{ij}^{str}, w_{ij}^{j}, and w_{ij}^{task} have the same meaning as those in Section 5.2.

5.4. Back-to-Ideal Vector and Matrix

In most application situations, it is difficult to provide stiffness functions and to calculate the Back-to-Ideal potential energies. Furthermore, as illustrated in Section 5.2, it might also be difficult to compare different distances given the multi-model nature of the Agent-Buddy. In order to avoid these difficulties, we propose a method that uses a set of Back-to-Ideal energy difference vectors and matrices to guide agents in the selection of distances.

As analyzed in Section 5.3, there are totally *M* different ways to expose one user's status to another user. These M different ways correspond to M different distances d_1 , …, d_M among users. From a certain point of view, these distances encode the Z different virtual walls among team members. Suppose that there are totally *Q* different events to be concerned with respect to users in Agent-Buddy.

The Back-to-Ideal potential energy matrix from *i* to *j* with respect to user *i*, H_{ij}^{i}, is given by: $\mathrm{H}_{ij}^{i}=\begin{pmatrix} h_{11}^{i}\cdots h_{1M}^{i} \\ \vdots \\ h_{Q1}^{i}\cdots h_{QM}^{i}\end{pmatrix}$. Where h_{uv}^{i} gives the Back-to-Ideal potential energy when user *i* is at event *u* and agent *i* selected distance d_v as the distance from *i* to *j*. If distance d_v happens to be the ideal distance from *i* to *j* under event *u* with respect to agent *i,* then $h_{uv}^{i}=0$. In general, although user *i* might be at different states, only some special events might have different ideal distances. In most situations, user *i*'s ideal distance will be the same. Matrix H_{ij}^{i} is available to agent *i* at the beginning and is specified by user *i*. The values of the elements of H_{ij}^{i} encode the degrees of frustrations or tensions user *i* has for different selected distances under different events. Since the elements of the matrix provide the Back-to-Ideal potential energies with respect to user i, the calculation of $\delta_1(\langle s_1,\cdots,s_M\rangle)$ is avoided during the run time.

The Back-to-Ideal potential energy vector from *i* to *j* with respect to the organizational structure is given by: $H_{ij}^{str}=(h_1^{str},\cdots,h_M^{str})$. Where h_v^{str} gives the Back-to-Ideal potential energy with respect to the organization when agent *i* selects d_v as the distances from *i* to ***j***. If $h_v^{str}=0$, then d_v is the ideal distance. The vector H_{ij}^{str} is provided by the organization to agent *i* at the beginning. Thus the calculation of $\delta_2(\langle s_1,\cdots,s_z\rangle)$ is avoided during the run time.

The Back-to-Ideal potential energy vector from *i* to *j* with respect to agent *j* is given by $H_{ij}^{j}=(h_1^{j},\cdots,h_M^{j})$. Where h_v^{j} gives the Back-to-Ideal potential energy with respect to agent *j* when agent *i* selects d_v as the final distance. This vector encodes agent *j*'s preference on distances and is given by user *j* to agent *j* and is then passed by agent *j* to agent *i*. The calculation of $\delta_3(\langle s_1,\cdots,s_z\rangle)$ is thus avoided.

The Back-to-Ideal potential energy vector from *i* to *j* with respect to a given task t_q is given by: $H_{ij}^{t_q}=(h_1^{t_q},\cdots,h_M^{t_q})$. Where $h_v^{t_q}$ gives the Back-to-Ideal potential energy when agent *i* selects d_v as the distance from *i* to ***j***. The awareness requirements for a collaborative task might be given by the authority who assigns the task, or by the group conventions about the awareness level of the task, or by Agent-Buddy according to various experiences inputted by users. In general, Agent-Buddy divides collaborative tasks into different categories according to the degree of awareness requirements for each member. It stores these tasks and the associated Back-to-Ideal potential energy vectors in a common place such that each agent can retrieve the corresponding vector according to its role in the team. The potential energy vectors for all the tasks are available at the beginning, thus the calculation of $\delta_4(\langle s_1,\cdots,s_z\rangle)$ is avoided.

5.5. Determination of the awareness distance

As discussed in the above section, the related Back-to-Ideal potential energies are all available for agent *i*. Thus, when a new collaboration task is assigned to user *i* or a new event is happening to user *i*, agent *i* will update the distances from its user to all the other related users.

Suppose that at time τ, user *i* is at the state of event u and the current collaboration task is t_q, then the weighted Back-to-Ideal potential energies for distance d_v is:

$$\delta(d_v)=w_{ij}^{i}\times h_{uv}^{i}+w_{ij}^{str}\times h_v^{str}+w_{ij}^{j}\times h_v^{j}+w_{ij}^{task}\times h_v^{t_q}.$$

To select the best distance, agent *i* calculates the weighted Back-to-Ideal potential energies $\delta(d_1)$, $\cdots$, $\delta(d_M)$ for all the distances d_1, $\cdots$, d_M and chooses the distance *d* with the minimum energy as the value of $d_{ij}(\tau)$, the distance from *i* to *j* at time τ. In other words, if $\delta(d)\le\delta(d_v)$ ($v=1,\cdots,M$), then $d_{ij}(\tau)=d$.

At the beginning, all the distances within Agent-Buddy select their awareness distances to all the other agents according to the above method by assuming that there is no collaboration task. Thus, only the first three terms are involved in the calculation: $\delta(d_v)=w_{ij}^{i}\times h_{uv}^{i}+w_{ij}^{str}\times h_v^{str}+w_{ij}^{j}\times h_v^{j}$. After $\Psi(0)$ is determined, if there is no change in the status of any users and there is no new task, then the awareness status of Agent-Buddy will stay the same. This status will be updated whenever there are changes in events or tasks. When a change occurs, each related agent will update its distances to all the other agents according to the above described method. The awareness status $\Psi(\tau)$ of Agent-Buddy is a system that is adaptive to events and tasks. Each element $d_{ij}(\tau)$ of $\Psi(\tau)$ is an adaptive media wall in the virtual organization of Agent-Buddy. It is these virtual walls that keep the organization functioning and provide adaptive awareness to all the members of the team.

6. THE INFLUENCE OF INDIVIDUAL BEHAVIOUR ON GLOBAL PERFORMACE

Here we study the influence of an individual agent's behavior on the global team performance. There are many factors that can affect an individual agent's behavior. For example, the Back-to-Ideal matrixes and Back-to-Ideal vectors influence an agent's selection of distances. However, these factors encode the intrinsic properties of agents, the tasks at hand, and the organization. What we are interested in is how an agent's personal properties, such as how it balances various preferences for itself, other agents, the task at hand and the organization, influence the outcomes of various kinds of global tasks. We hope that the empirical results along this line can provide some guidelines in the construction of virtual organizations.

We use the following virtual organization structure for our experiments. In Figure 6, each small circle represents an agent. If there is a line connecting two circles, then users represented by the two circles have a direct management relationship, where the one above is the manager of the one below. For example, user b is the manager of users e, f, and g. User i is the manager of user r, s, t, and u. In this organization, a is the CEO. We assume that users of this company work in distributed places, thus awareness plays a big role in the functioning of the company. Please note that this figure is not the structure of the Agent-Buddy for the organization. The structure of the Agent-Buddy is represented by a complete directed graph where circles of Figure 6 are vertexes of the graph and there are two directed links connecting each pair of vertexes.

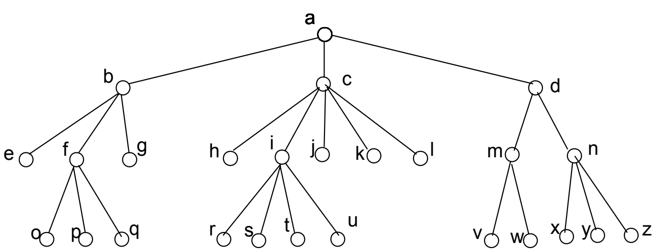


Figure 6. Topological structure of the organization

Suppose that there are 100 different distances $d_1, \cdots, d_{100}$ that can be used to provide awareness among co-workers of this company. Suppose that the smaller the index of the distance, the more the information is revealed to the receiving user. Thus, d_1 provides the maximum awareness and d_{100} provides the minimum awareness.

The ideal distances from the organizational structure point of view are given as follows. *The ideal distance from a user to its first line manager is* d_{35}. For example, distances from b, c, and d to a, distances from e, f, and g to b, distances from x, y, and z to n, etc., are all equals to d_{35}. When we say that the distance from e to b is d_{35}, we mean that user b can check the activities of user e with the awareness degree given by d_{35}. *The ideal distance from a first line manager to its direct employee is* d_{75}. For example, distances from a to b, c, and d, distances from c to i, distances from m to v and w, etc., are all equal to d_{75}. *The ideal distance from a user to his second line manager is* d_{55}. For example, the distance from e to a is d_{55}. The ideal distance from a user to his third line manager is d_{65}. For example, the ideal distance from q to a is d_{65}. *The ideal distance from a second line manager to his second line employee is* d_{91}. For example, the ideal distance from c to t is d_{91}. *The ideal distance from a user to his third line manager is* d_{65}. For example, the ideal distance from u to a is d_{65}. *The ideal distance from a third line manager to his third line employee is* d_{100}. For example, the ideal distance from a to u is d_{100}. This means that u has very little information on what a is doing. *The distances between any users that have the same first line manager is* d_{50}. The distance between any users that do not share the same management chain or the same first line manager is d_{95}. For example, distances from r to n, distances from r to q, distances from b to r are all equal to d_{95}. Figure 7 gives a subset of the ideal distance map of the organization.

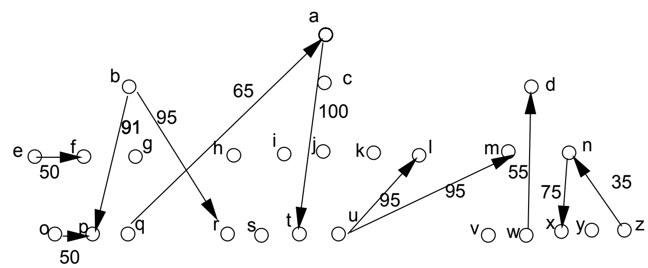


Figure 7. Subset of organizational ideal distance map.

Suppose that for any agent, its ideal distances to any other agents are d_{77} and it hopes that any other agents can expose their activities at the awareness degree of distance d_{31}.

To make the discussion easier, we assume that the Back-to-Ideal potential energy can be calculated according to Hooke's law and that the stiffness function equals to constant *1* under all situations. Thus, the tension vectors and matrix can be easily obtained or calculated. For example, if the ideal distance is d_{50} and the selected distance is d_{65}, then the Back-to-Ideal potential energy can be calculated by: $\frac{1}{2} \times 1 \times (65-50)^2 = 112.5$.

Our task is to evaluate how Agent-Buddy helps the productivity of a distributed collaborative work. This evaluation is based on how good the awareness is provided to team members by agents of Agent-Buddy, or in other words, how the various Back-to-Ideal potential energies or tensions are handled by those agents. The following formula is used to calculate the total Back-to-Ideal potential energies:

$$\Re(t_q) = \sum_{u \in Team(t_q)} [K_{str}\Delta_{str} + K_{task}\Delta_{task} + K_{self}\Delta_{self} + K_{others}\Delta_{others}].$$

Where t_q is the current task. $Team(t_q)$ is the set of all the team members for the given task. $\Re(t_q)$ is the total weighted tension from all the agents of the Agent-Buddy related to this task. The higher the value of $\Re(t_q)$ is, the worse the performance. The contribution of each team member is the weighted sum of four factors. Weights K_{str}, K_{task}, K_{self}, and K_{others} give the sensitivity of the task with respect to awareness tensions in organizational structures, the current task, agents' own expectations and other agents' expectations respectively. They satisfy $K_{str} + K_{task} + K_{self} + K_{others} = 1$. Δ_{str} is the total structural tension from agent *u* to all the other agents related to the task. In practice, Δ_{task} is the total task tension from agent *u* to all the other related agents. Δ_{self} is the total self tension from u to other agents. Δ_{others} is the total tension with respect to other agents' expectations from u to other agents. In practice, Δ_{str}, Δ_{task}, Δ_{self}, and Δ_{others} can be obtained from H_{ij}^{str}, $H_{ij}^{t_q}$, H_{ij}^{i}, and H_{ij}^{j} as described in Section 5.3. Here we directly calculate the values of Δ_{task}, Δ_{str}, Δ_{self}, and Δ_j.

Since K_{str}, K_{task}, K_{self}, and K_{others} give the properties of the task and w_{ij}^{i}, w_{ij}^{str}, w_{ij}^{j}, and w_{ij}^{task} determine agents' behavior, we are able to study the influence of agents' behaviors on the performance of Agent-Buddy by varying the above factors. In the following few experiments, we assume that agents b, f, q, i, and r are involved in the task.

Figure 8 shows the situation where agents' concerns on structural needs can influence the system's performance. The ideal distance for the task is d_{50} and it is neutral, which means that $K_{str} : K_{task} : K_{self} : K_{others} = 1:1:1:1$. For all the agents, the ratio of their behavior weights is given by $w_{ij}^{i} : w_{ij}^{str} : w_{ij}^{j} : w_{ij}^{task} = 1 : w_{ij}^{str} : 1 : 1$. Figure 8 shows how $\Re(t_q)$ and other Back-to-Ideal energies (tensions) are influenced when w_{ij}^{str} changes from *0.1* to *10*. We can notice that when agents put more weight on the organizational structure, the sum of the total structural Back-to-Ideal energies for all agents will decrease. The sum of the total "other" Back-to-Ideal energies for all agents will decrease. This is because when agents emphasize structure more, they will put less weight on task awareness requirements and other agents' awareness requirements. Thus, distance offsets with respect to these two factors will increase. It is interesting to note that the sum of the total "self" Back-to-Ideal energies for all agents will first decrease until the weight on structure equals to *3.7*, and then the sum will increase. This is because when forces that pull distances toward the structural ideal directions become bigger and bigger, they also happen to pull distances towards directions of "other" distances from the *global* point of view. This situation will be changed when forces along "structure" direction are too big such that actual distances pass "self" distances and go to other directions. We can notice that the value of $\Re(t_q)$ will decrease until the weight on structure is around *1* and will increase after that. This tells us that for a neutral task, agents that extremely over emphasize or de-emphasize the organizational structure are not good. Thus, it is better to assign a neutral task to a group of agents that are also neutral.

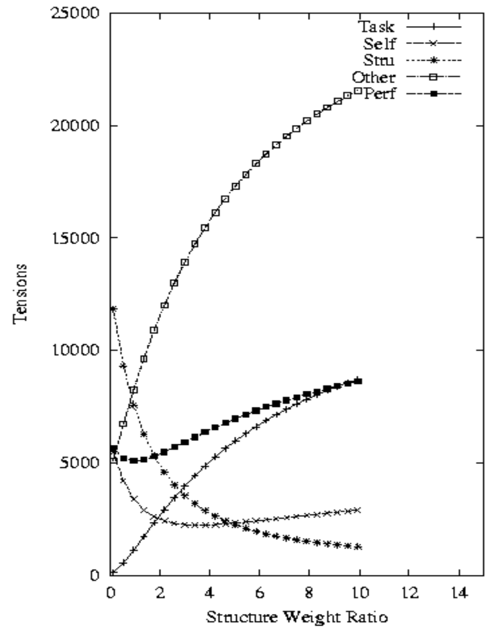


Figure 8. How agents' concerns on structure influence the system performance.

Figure 9 shows how agents' concerns on other agents' needs can influence the system performance. This time, the ratio of agents' behavior weights is given by $w_{ij}^{i} : w_{ij}^{str} : w_{ij}^{j} : w_{ij}^{task} = 1:1: w_{ij}^{j} :1$, and w_{ij}^{j} changes from *0.1* to *10*. We can notice that $\Re(t_q)$ is bigger when the weight is at *10 than* it is when the weight is at *1*. This tells us that sometimes extremely collaborative agents may not help team performance. It really depends on the nature of the task. The reason is that when agents are too concerned with other agents' needs, the needs from other sources might be neglected. As a result, the performance as a whole might decrease.

Figure 10 shows how the selfishness of an agent influence the system performance. The ratio of agents' behavior weights is given by $w_{ij}^{i} : w_{ij}^{str} : w_{ij}^{j} : w_{ij}^{task} = w_{ij}^{i} :1:1:1$, and w_{ij}^{i} changes from *0.1* to *10*. We can notice that the performance, $\Re(t_q)$, reaches its minimum when w_{ij}^{i} is around 1. Thus, for a neutral task, the more an agent emphasizes itself, the worse the performance.

Figure 11 shows how agents' concerns on the awareness requirement influences the performance. The ratio of agents' behavior weights is given by $w_{ij}^{i} : w_{ij}^{str} : w_{ij}^{j} : w_{ij}^{task} = 1:1:1: w_{ij}^{task}$, and w_{ij}^{task} changes from *0.1* to *10*. We can notice that the performance is best when w_{ij}^{task} is around 1. When agents over emphasize the awareness requirement of the task, the performance reduces rather than increases. This is because the property of the task itself is neutral, thus, a departure from its own requirement may not influence the success of the task with big impact. However, it does influence other factors. Thus, the combined results will reduce the performance of the team.

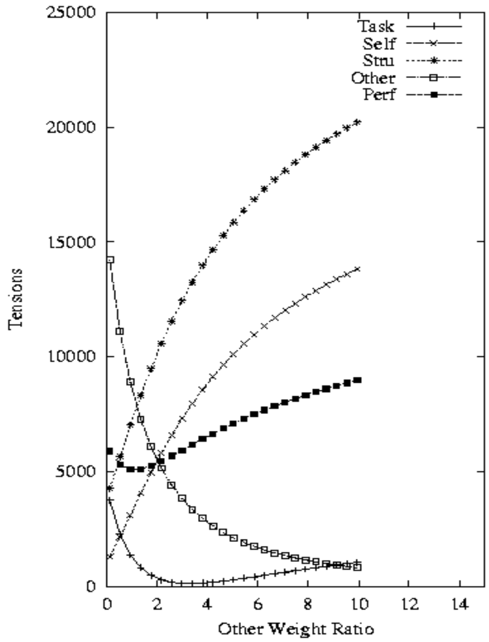


Figure 9. How agents' concern on other agents will influence the system performance.

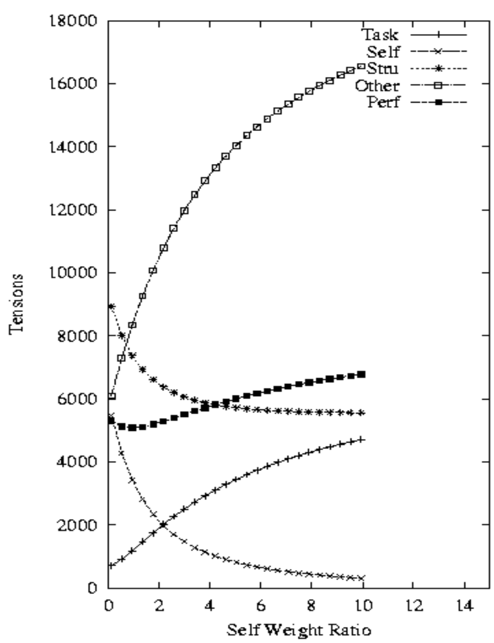


Figure 10. How agents' concern on itself will influence the system performance.

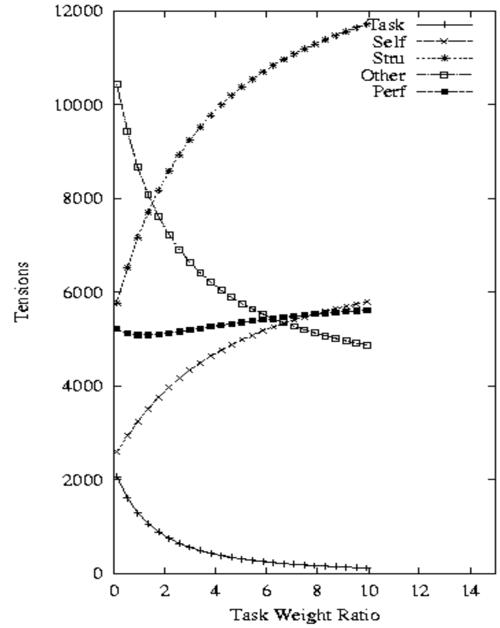


Figure 11. How agents' concern on the current task will influence the system performance.

Figure 12 shows how the value of $\Re(t_q)$ will be influenced when we change agents' behavior weights w_{ij}^{i}, w_{ij}^{str}, w_{ij}^{j}, and w_{ij}^{task}. For example, the curve "P_Str" corresponds to the situation that w_{ij}^{str} changes from *0.1* to *10* while other weights all equal to *0*. We can notice that the best performance occurs when the factors are around position *1*. This further illustrates that when agents' behaviors match the properties of the task, the performance of the system will be high.

Now, we change the property of the task in another test such that $K_{str} : K_{task} : K_{self} : K_{others} = 5:7:5:9$. The task is no longer neutral. Thus, neutral behaviors of agents will not generate the best performance. This is shown in in Figure 13. We can also notice that when corresponding weights pass the value of *1,* the Back-to-Ideal energies related to structure and the Back-to-Ideal energies related to agents themselves increase much faster than the other cases. This is because the task does not emphasize these factors. Thus, if agents over-emphasize them, the performance of the team will decrease. In general, the performance of the team depends on various factors and can be very complex. Based on our extensive experiments, we find that in most situations, a better match of agents' behaviors and task properties tends to provide a better team performance.

Figure 14 shows how the value of $\Re(t_q)$ will be influenced when we change agents' behavior weights w_{ij}^{i} and the task property K_{self}. Here the ratio for agents is: $w_{ij}^{i} : w_{ij}^{str} : w_{ij}^{j} : w_{ij}^{task} = w_{ij}^{i} : 1 : 1 : 1$, where w_{ij}^{i} changes from 0.1 to 10. The ratio for the task is:

$K_{self} : K_{str} : K_{others} : K_{task} = K_{self} : 1 : 1 : 1$. The term K_{self} changes from 0.1 to 2.7. The line shows that when the properties of the task and the properties of the agents match, the system obtains its best performance.

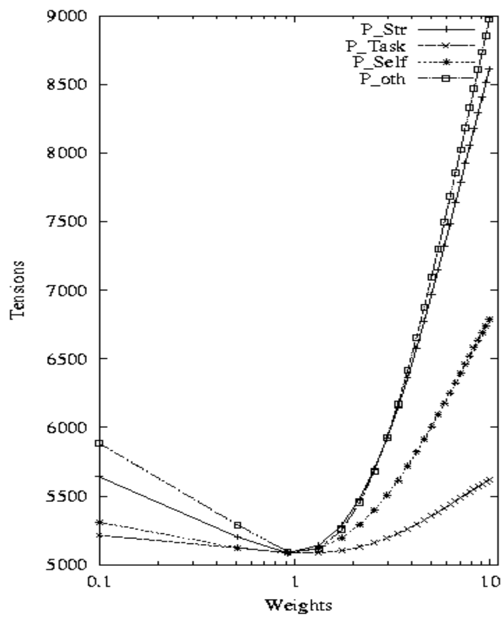


Figure 12. Performance comparison.

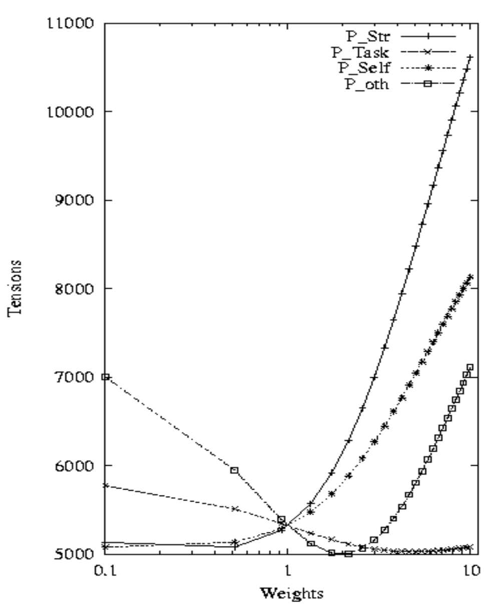


Figure 13. Performance comparison (continued).

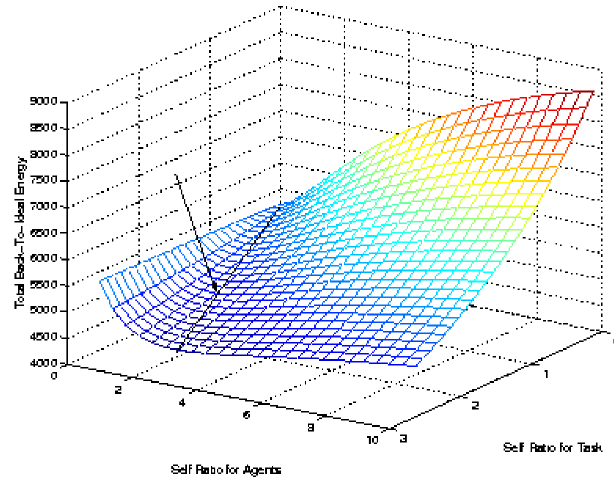


Figure 14. Performance Comparison (continued).

7. DISCUSSIONS AND CONCLUSIONS

Many researchers have addressed the issue of multiagent collaboration within a multi-user environment [6] [12] [13] 14] [1] [14] [15]. The one that most related to ours is the work done by Grosz and her group [6] on GIGAGENTS that models and supports explicit collaboration in planning and acting among both human and digital agents. Our work differs from theirs in that they emphasize the application of SHAREPLANS in group decision making, while we emphasize the adaptive adjusting of the awareness network by agents with the goal to minimize the total awareness frustrations of users for a given collaborative task. There is a significant body of work done by HCI and CSCW communities on collaboration [3] [8] [9] [10]. However, their works have a strong emphasis on the social aspect of collaboration with no agents involved, while our work addresses a multiagent approach to collaboration.

In this paper, we propose the concept of distance and smart distance in a distributed collaborative environment. We illustrate an Agent Mediated Collaborative system - the Agent-Buddy system that can create a sense of group presence and at the same time preserve the privacy of each user. We define the multiagent awareness network to represent the awareness situations among team members in a virtual organization or in a CSCW scenario. A virtual spring is used to model the awareness degree among team members. Each agent makes decisions by considering multiple factors. The goal of the multiagent team is to minimize the global awareness frustrations with respect to different kinds of tasks. Empirical studies have been conducted to analyze the individual agent behavior on the global performance.

With ubiquitous connectivity on the horizon, collaborative computing will become one of the major applications in the evolution of computing and communication. The goal of our research is to dynamically adjust the “distance” among people in a collaborative environment - breaking the isolation, providing group awareness, and at the same time, keeping the privacy. It is our belief that researches in multi-user and multi-agent aspects of virtual organizations such as an awareness network will become