

# A Hypermedia Environment for Global Collaboration

Kenji Takahashi and Eiji Yana  
*NTT*

We developed a hypermedia environment for global collaboration by knowledge workers, which consists of InterPOD and TeamSmart. InterPOD creates a meeting “room” environment for participants to access and share multimedia information from various sources. The TeamSmart hypermedia tool supports collaboration on document production by globally dispersed teams.

Global collaboration on knowledge work, such as product development, is vital in many industries as competition extends across regional borders. Collaborating in such an environment means working between cross-functional, cross-cultural teams separated by time-zone differences and geographical distance. The Internet infrastructure has made global collaboration possible for large numbers of people worldwide, and collaboration tools support many project activities. However, the successful execution of projects still largely depends on personal efforts, such as leadership by “super” designers, late-night (or early-morning) meetings, and expensive time-consuming international travel. To address these problems, we developed an integrated environment to systematically support collaborative knowledge work. The environment consists of InterPOD and TeamSmart.<sup>1</sup> InterPOD supports conferences by providing fuller control over the various media used during a conference. The TeamSmart tool supports document creation by a globally dispersed team through hypermedia.

Collaborative knowledge work is inherently difficult. The stakeholders in a project (designers, clients, users, and so on) are typically distributed. Plus, any given project is usually one of several that involve each stakeholder. Global projects thus represent a collaborative process using and creating large amounts of information that must be managed and shared effectively. Our research has focused on

- Extending the Web into a true hypermedium to collaboratively create and manage knowledge rather than to merely disseminate information in a one-way manner.
- Developing specific applications to support and manage collaborative activities such as face-to-face meetings and distributed document creation.
- Implementing an environment to provide multimodal support—including synchronous/asynchronous and colocated/dispersed support—to overcome time-zone differences and geographical distance.

## Problems

Time-zone differences and distance create major stumbling blocks in collaborative knowledge work by a globally distributed team. In practice, any significant global project involves face-to-face meetings to exchange information, ideas, and understandings concerning the current project status. Between meetings, team members work on their responsibilities mostly in an asynchronous manner. This process of synchronous work (meetings) and asynchronous work is usually repeated until the end of the project. However, face-to-face meetings for global projects are expensive and time consuming because of distance. Also, scheduling remote conferences (such as telephone conferencing) proves difficult because of time-zone differences.

Meetings themselves are difficult enough to conduct even when participants are in the same room. Kaiya and Saeki<sup>2</sup> videotaped and analyzed meetings held to discuss the requirements analysis of software systems. They found that a large portion of the discussion failed to reflect the specification documents and that time was spent inefficiently. They reported that 36.3 percent of all decisions made during meetings were left out of the written minutes, 21.7 percent of the total meeting time was spent on discussions that went around in circles, and 25 percent of the meeting time was spent on discussions that led to no decisions.

In addition, global collaboration projects are often cross-cultural, cross-organizational endeavors that involve the participation of various stakeholders with different backgrounds, goals, and levels of commitment.<sup>3</sup> These stakeholders use a variety of representations and documents as part of a distributed, iterative process where the stakeholders,

understandings, and representations evolve.

Time and distance barriers are obstacles to reaching common understandings among stakeholders in the following activities:

- *Communication*. What should be discussed? What are the real problems?
- *Agreement*. What's agreed upon and what's undecided? What's the current project status?
- *Traceability*. Which artifacts (that is, objects of collaboration such as product design) have been or should be changed? Why and how were they changed or should be changed?

Here we consider two approaches to solving these problems:

1. Improve the efficiency and quality of remote or face-to-face conferencing.
2. Maximize conferencing output and efficiently use asynchronous collaborative work.

We implemented InterPOD for the first approach and TeamSmart for the second one. We combined these two systems to overcome time and distance barriers, thus providing a global collaboration environment.

Researchers developed electronic whiteboard systems such as Collab<sup>4</sup> and Tivoli<sup>5</sup> based on the first approach. These systems use specialized large-screen devices to implement advanced user interfaces such as direct-object manipulation by pen and gesture. Such specialized devices, however, make these systems expensive, difficult to use with other systems, and hard to maintain. In addition, these systems don't meet user demands for using office space efficiently and flexibly because they're large and not portable. Group-decision support systems (GDSSs)<sup>6</sup> focus on a specific activity—the decision-making process—in meetings based on process models (such as voting). A GDSS complements other enabling technologies, such as multimedia conferencing tools, to provide total solutions for meetings over distance.

To provide an advanced meeting-support environment suitable for a wider range of workplaces, we designed InterPOD as an affordable, interoperable, modular, and portable system. InterPOD uses a commercial off-the-shelf analog video-signal-switching system because it's fast, easy to set up, and less expensive. InterPOD supports collab-

oration in knowledge work rather than a seminar-style meeting in which one person makes a presentation to the other meeting participants.

The video-switching system enables the rapid exchange of multimedia information. To control the video switching, InterPOD uses Java and Web technologies. The portability and modularity of the InterPOD system lets users move the systems around and combine them with other systems in a variety of configurations. InterPOD provides a smaller set of functions than the systems just mentioned, but is more collaborative, cost effective, and easier to deploy.

Based on the second approach, researchers developed design-tracking tools such as gIBIS,<sup>7</sup> Sybil,<sup>8</sup> and Tuiqiao,<sup>9</sup> to record the design-decision-making process in a textual format. NoTime,<sup>10</sup> Filochat,<sup>11</sup> and Audio Notebook<sup>12</sup> locally record and replay face-to-face meetings in audio and video format. Streams<sup>13</sup> is a server that can record broadcast seminars (or one-to-many communication) in audio and video and replay these to remote clients on demand.

In addition, many document management tools such as OpenText (<http://www.opentext.com>) and Documentum (<http://www.documentum.com>) are commercially available. These systems only support the management part of the collaboration between globally dispersed knowledge workers that concerns a single set of documents. Such collaboration involves various work styles—including synchronous/asynchronous and collocated/dispersed—and requires complete traceability of working documents during the transition between the different work styles. TeamSmart provides integrated support for global collaboration on document creation using networked hypermedia. TeamSmart seamlessly supports both synchronous and asynchronous collaboration by integrating real-time communication services (such as videoconference recording/replaying) and non-real-time document processing (for example, editing and versioning). Internet protocol (IP) multicast handles real-time communication services, while enhanced Web technologies take care of non-real-time document processing.

We constructed a prototype of a collaboration environment that bridges Tokyo, Japan, and Palo Alto, California, by installing InterPOD and TeamSmart at each site. We connected the two sites via a closed IP network over an asynchronous transfer mode (ATM) network using a 1.5-Mbps per second (Mbps) constant bit-rate (CBR) service. This environment supported the development of

products such as collaboration tools and networking systems. We provided the following collaboration services for these trials:

- An integrated group interface
- The ability to share screen images from computers and various media equipment
- Web-based remote control of various media equipment
- Multipoint videoconferencing
- Videoconference recording, indexing, and playback
- Application program sharing

### InterPOD

The InterPOD system supports meetings related to collaborative knowledge work by providing participants with seamless access to various media and information. InterPOD serves as the core meeting “room” environment, which uses multimedia presentation and communication technologies to augment and facilitate social protocols among participants.

InterPOD’s design builds on three concepts:

- *Seamless information access.* Any information or person in charge of a particular matter can be accessed during a meeting via a local-area network (LAN) or the Internet without leaving InterPOD. Such information access lets participants make decisions with sufficient information.
- *Integrated interface for groups.* The group interface isn’t a mere collection of individual user interfaces, but is integrated to facilitate natural, social protocols among participants.
- *Flexible media handling.* The input and output equipment for various media (video cameras, videoconferencing systems, electronic blackboards, and so on) can be easily controlled with little or no need to touch physical connectors or plugs. This lets participants use any type of media that are most appropriate for their purposes.

The main functions of InterPOD are Web-based workspace management and integrated multime-

dia device control. Through Web browsers, InterPOD lets meeting participants create public and private workspaces, and exchange the workspaces in a controlled manner based on their roles (for example, chairperson). The workspaces can be remotely controlled and transmitted over distance. The device-control functions provide hardware and software interfaces to various devices including videoconferencing systems. InterPOD system units can be combined with each other to provide additional capacity as needed.

### InterPOD configuration

The InterPOD system consists of a main unit and seven personal information boxes (PIBs). The main unit includes a personal computer that acts as a control server and a video-switching system. The PIBs let individual participants connect the input and output of their personal computers and monitors to the InterPOD main unit. The meeting environment—centered around the InterPOD system—can consist of individual monitors for the participants, shared displays, media equipment such as a videoconferencing system, and the participants’ computers. Participants usually bring their own portable computers. The control server manages all the components centrally via a Web-based user interface.

### Workspaces

InterPOD has two types of workspaces—personal and common. Each user has a personal workspace, and all users share common workspaces. Personal workspaces let users retrieve information and take notes. Common workspaces permit users to share information and work on the current topic. Having two types of workspaces is analogous to a conventional daily meeting—while participants have their own notebooks and documents, all share the current topic on a whiteboard.

InterPOD lets users project their personal workspaces onto the common workspaces. It also lets them copy the information from the common workspace into their personal workspaces. In this way, users can exchange information and ideas. However, InterPOD protects the privacy of the personal workspace: only a chairperson can project a personal workspace, and participants can’t see the personal workspaces of others unless their owners or the chairperson publishes the workspaces.

InterPOD doesn’t implement a specific floor-control mechanism. Anyone can take over the common spaces at any time. As a result, conflicts that occur must be solved through human negoti-

ation. The current implementation of InterPOD has two common workspaces. For example, one common workspace might display information on the current topic, and the other might enhance the meeting atmosphere by showing a video image of a remote site participating through videoconferencing. Alternatively, participants can use the two spaces to compare relevant information.

InterPOD provides three types of user interfaces depending on the roles of users: user, chairperson, and administrator. The interface for users lets them do only two things—project the contents of their personal workspaces onto the common workspaces and display a common workspace in their personal workspace. The chairperson can send information from any personal workspace to the common workspaces and vice versa. The administrator has full control over the display of personal and common workspaces for maintenance not usually performed during meetings, for example, configuring InterPOD systems.

InterPOD-based meetings can be conducted in several ways. For example, a chairperson can take a leadership role and centrally operate InterPOD. The chairperson in this case projects the personal workspace of a participant onto the common workspace as needed on behalf of all the participants. Alternatively, participants can proceed by themselves without a chairperson and project their personal workspaces onto the common workspaces cooperatively.

We implemented the user interface programs of InterPOD as a Java applet (Figure 1). Wireless remote operation by a personal digital assistant (PDA) is also an option.

### Use scenario

Suppose that Kenji and Eiji in Japan, and Jeff in the US hold a meeting using InterPOD. Kenji is the chairman, and Eiji and Jeff are participants. If Eiji displays tables of data from his workspace—the screen of his laptop computer—in a common work-

space, or shared display, Kenji will have the screen of Eiji's computer displayed on his personal monitor (Figure 2). In addition, Kenji, the chairperson, can send the output of the videoconferencing system to the second shared monitor to see Jeff. On

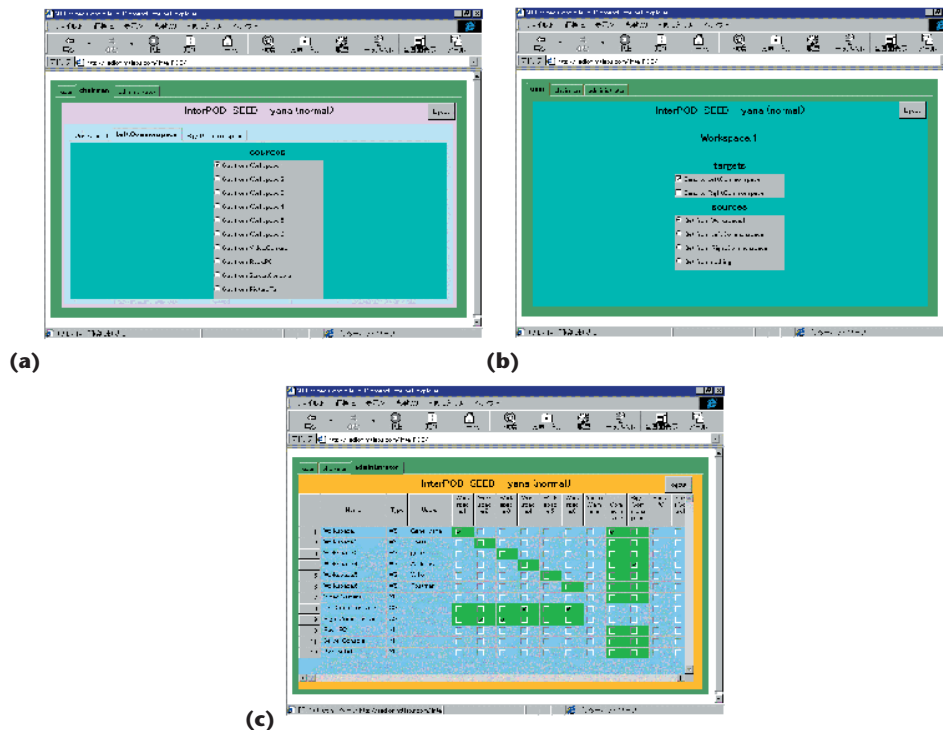


Figure 1. Three InterPOD user interfaces. (a) Chairpersons select the information to project onto a common workspace. (b) Participants choose a target common workspace from which to project their workspaces and display the source information on their monitors. (c) Administrators decide which combinations to configure between the information sources and target workspaces. The highlighted cells show one such combination.

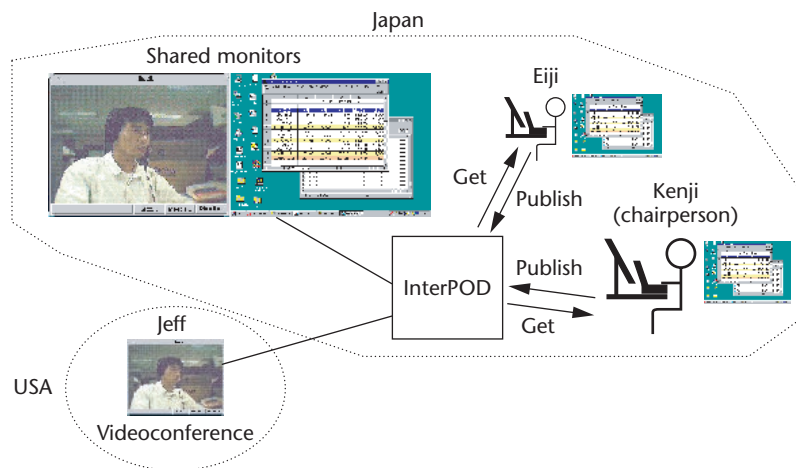
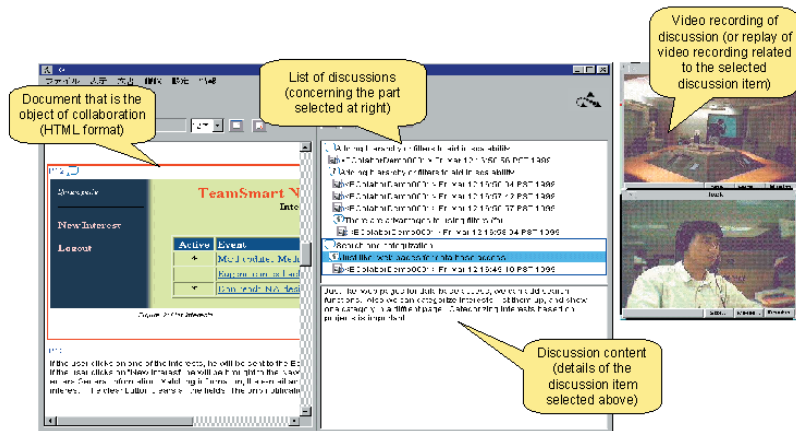


Figure 2. Screen sharing in InterPOD. The system lets Eiji display his tables of data in a common workspace so that Kenji and Jeff can see them on their monitors.



**Figure 3.** Main features of TeamSmart: An HTML document that's the object of collaboration (top left), list of discussions (top middle), video recording of discussions from one remote site (top right), close-up of participant at the other remote site during the videoconference (bottom right), and details of the discussions (bottom middle).

behalf of Eiji, Kenji can also display either Jeff's image or the tables of data from Eiji's monitor.

### TeamSmart

A hypermedia tool, TeamSmart supports the multimodal collaborative work of creating documents by team members in different locations (Figure 3). It's based on the Inquiry Cycle model,<sup>14</sup> and implemented using metalevel links, which we discuss in detail in the sidebar "The Inquiry Cycle."

Documents, such as design specifications, represent the final deliverables in many collaborative knowledge-work projects. Team members work on documents in various ways. For example, team members may work asynchronously and concurrently on a part that's their responsibility and make comments. All members still have to meet, though, to synchronize each member's progress and conduct a peer review of the entire documents. Some members may not be able to attend meetings, and some may leave or join in the middle of a project.

Although many collaboration and documentation tools exist, it's difficult to maintain complete consistency and traceability of contributions to the documents by team members. Current tools mostly support distinct project activities—such as e-mail for exchanging comments, document-management tools for change control, the Web for publishing documents, and videoconferencing tools for peer reviews. For example, members point out problems in documents by using e-mail, solve them through videoconferences, and

## The Inquiry Cycle

The Inquiry Cycle, devised by Potts, Takahashi, and Anton<sup>1</sup> is a model for collaborative document creation. It involves the repetition of three activities: expression, discussion, and commitment (Figure A).

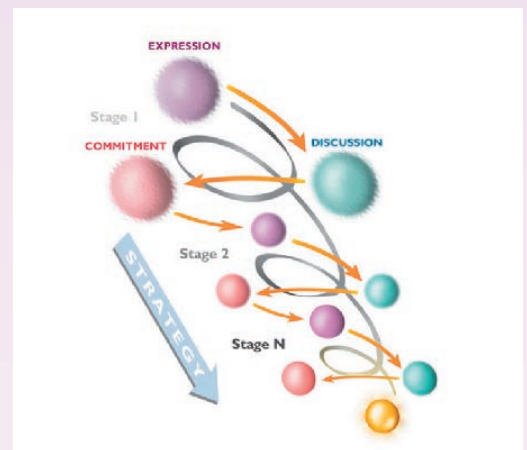
- Expression deals with preparing and presenting ideas to be documented.
- Discussion involves discussing the documents and includes sharing comments and individual annotations on the documents.
- Commitment includes planning changes to the documents based on results from the discussions—such as change requests and agreements about terminology—and executing the plans.

The Inquiry Cycle has two main advantages over similar models:

1. artifact-centered integration of expression, discussion, and commitment, and
2. simple, flexible, and generic structures.

Having the artifact (document) at the center of discussion and commitment contributes to improving it while maintaining traceability because participants focus their efforts on the document. This approach also helps users resolve disagreements.

Likewise, the model's simplicity and flexibility permits instantiations of the model with varying degrees of formality or the use of different con-



**Figure A.** The stages of the Inquiry Cycle—expression, discussion, and commitment.

ventions based on the users' needs. TeamSmart uses "questions and answers" as a model for the discussion phase to capture the rapid exchange of ideas and information among project members. However, analytical models (such as the Issue-Based Information System, or IBIS,<sup>2</sup> and questions, options, and criteria, or QOC<sup>3</sup>) can be used for detailed problem analysis. In our instantiation, commitment comes from a rough consensus or the decision of the document editor. Commitment can also be based on a voting system or in more formal environments on a multiple level "sign-off" where both a vendor and a customer would need to sign an agreement concerning changes.

### Metalevel links

We use metalevel links to represent and keep track of relationships among the pieces of information generated through the Inquiry Cycle.

Metalevel links implement "true" hypermedia capabilities, which enable the dynamic creation and cross-referencing of knowledge on the Web. Metalevel links relate Web resources to one another in a formal manner, outside of the Web pages' contents.<sup>4</sup> We implemented the metalevel linking mechanism as an extension to the hypertext protocol (HTTP v.1.1). For example, Figure B compares how HTML and metalevel links represent relationships between a question and its answer. HTML links can't formally relate Web resources, because the relationships among anchors, destinations, and the entire Web resource can't be clearly defined. Although Extensible Markup Language (XML) formally represents and contains the relationships within contents, manipulating the relationships remotely still proves difficult with XML alone.

Metalevel links are useful for supporting collaborative knowledge work in several ways. For example, users can

- Derive the knowledge structure represented in hypermedia faster because it's not necessary to transmit and parse the entire contents of Web resources.
- Manipulate links from remote hosts without changing the contents of Web resources.
- Detect changes to links made via formal protocols in real time.
- Represent bidirectional relationships using a pair of metalevel links opposite each other.

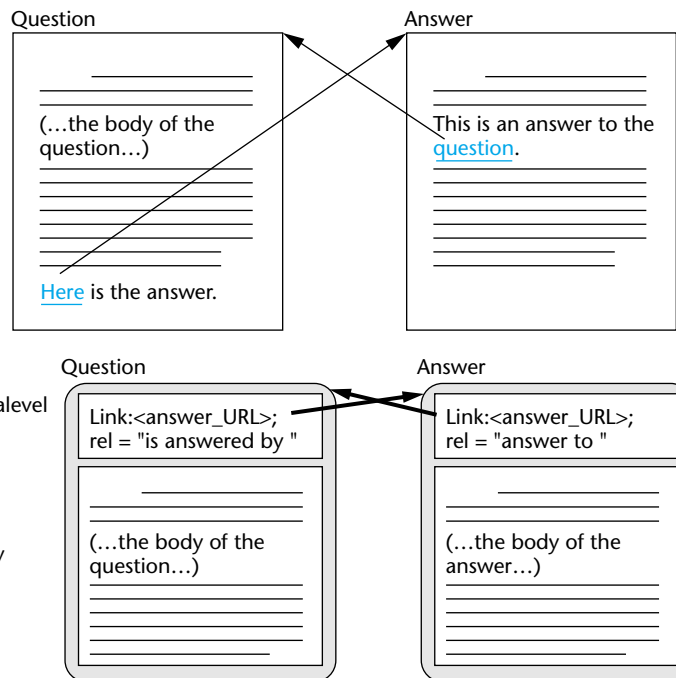


Figure B. Comparison of how HTML and metalevel links represent relationships between a question and its answer. Two Web resources refer to each other through HTML links (top), while two Web resources relate to each other through metalevel links (bottom).

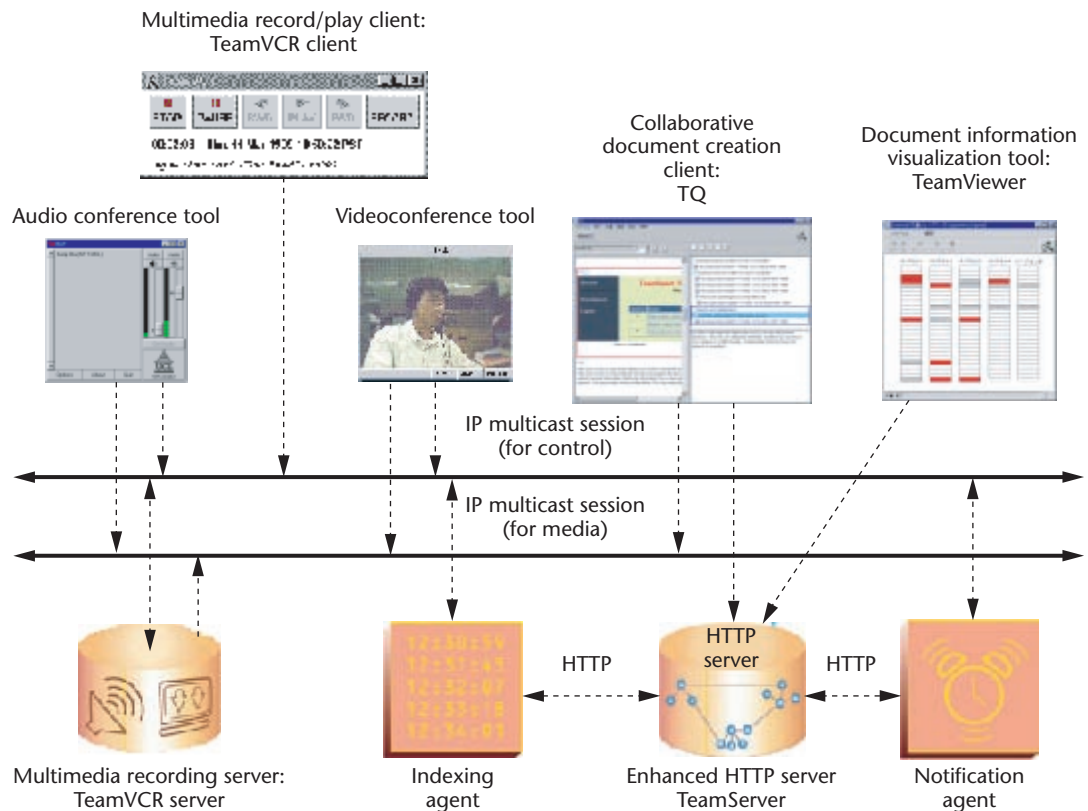
- Do intelligent searching based on relationships. For instance, if resources are linked with "parent-child" links, users can retrieve the "grandchildren" of a resource by traversing resources "two links" away.

Finally, metalevel links provide different views of knowledge structures to individual users by filtering links based on link types.

### References

1. C. Potts, K. Takahashi, and A.I. Anton, "Inquiry-Based Requirements Analysis," *IEEE Software*, Vol. 11, No. 2, March 1994, pp. 21-32.
2. W. Kunz and H. Rittel, *Issues as Elements of Information Systems*, Working Paper 131, Inst. Urban and Regional Development, Univ. of California at Berkeley, 1970.
3. A. MacLean et al., "Questions, Options, and Criteria: Elements of Design Space Analysis," *Human-Computer Interaction*, Vol. 6, No. 3 and 4, 1991, pp. 201-250.
4. K. Takahashi, "Metalevel links: More Power to Your Links," *Comm. ACM*, Vol. 41, No. 7, July 1998, pp. 103-105.

Figure 4. TeamSmart architecture.



revise the documents. No one may participate in all the activities, which take place at different locations and times. Misunderstandings and mistakes can easily slip in during the transition between activities done by different team members who have different work styles and use different tools. With this approach, no systematic support exists to manage the activities and relevant information that led to the refinement of the documents.

To address these problems, TeamSmart captures, manages, and provides all the information referred to and generated (including each version of documents, annotations, and video-recorded discussions) during a project as hypermedia woven with metalevel links. TeamSmart seamlessly supports asynchronous and synchronous collaboration by, for example, enabling multimedia conferences and recording them in a way that links the records to relevant portions of documents and comments. The hypermedia format gives globally distributed collaborators easy access to the details of document changes and results. Team members unable to attend a meeting because of time or distance barriers can refer to the complete record of a meeting through TeamSmart. The main functions of TeamSmart are

- Collaborative editing of documents
- Version control and configuration management
- Recording, indexing, and replay of videoconferences
- Client sharing
- Visualization of the document creation process

Here's an example of a typical, simplified session using TeamSmart. A team in Tokyo, Japan, and one in Palo Alto, California, collaborate on design documents for a software tool. The Tokyo team is responsible for the client part of the tool and the Palo Alto team for the server part. First, the members of each team work asynchronously and prepare the first drafts of their work using TeamSmart on their desktop computers. Members can also use other word-processor programs to write and input the drafts to TeamSmart in Hypertext Markup Language (HTML) format. Once the drafts are available in TeamSmart, the members review and attach comments to them (still asyn-

chronously). At some point, each team meets in its InterPOD room to have a team-to-team videoconference. Using the synchronous functions of TeamSmart (such as client sharing) in the InterPOD environment, both teams synchronously review the same documents, respond to the comments previously made, agree on the changes to make, and find new problems in the documents. The meeting is video-recorded and TeamSmart automatically generates indices on the topics discussed. After the meeting, members revise the documents and solve the open problems, referring to the multimedia records of the meeting by using TeamSmart at their desktops. This cycle repeats until members complete the documents.

### TeamSmart architecture

TeamSmart consists of several clients and servers. Users mainly interact with TQ (from the Chinese word *tuiqiao*, which means elaboration), the main client of the TeamSmart system, to review, annotate, and edit documents. TQ communicates with TeamServer, an enhanced Web server that centrally manages the documents and related information. Figure 4 shows the TeamSmart architecture.

### TQ

TQ lets users edit HTML documents on the basis of the Inquiry Cycle. By using IP multicast technology, multiple TQ users can simultaneously view and perform operations on the same image of a document. For example, if user A selects a certain paragraph of the document in TQ, that paragraph is also selected in the TQ display of user B. Figure 3 shows the screen shot of TQ.

### TeamServer

As TeamSmart's core server, TeamServer manages all the data for TeamSmart. The server uses enhanced HTTP to manipulate metalevel links and manages different versions of Web resources. TeamServer and TQ communicate through enhanced HTTP. TeamServer creates and manages each item of information, including paragraphs of documents, questions, and answers as individual Web resources. The information items have their own URLs, connected by metalevel links. In this way, users can access and manipulate each information item via HTTP from anywhere in the world.

In Figure 5, for example, version 1 of a document consists of four paragraphs—P1, P2, P3, and P4. P1 is linked to question Q1, which links to two alternative answers, A1 and A2. A1 links to commitment C1, which leads to P1', a new version of

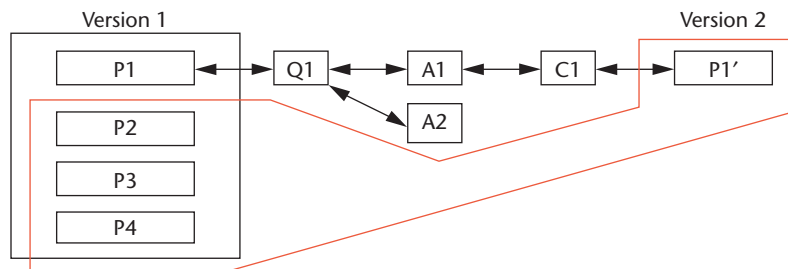


Figure 5. Version and configuration management by TeamSmart.

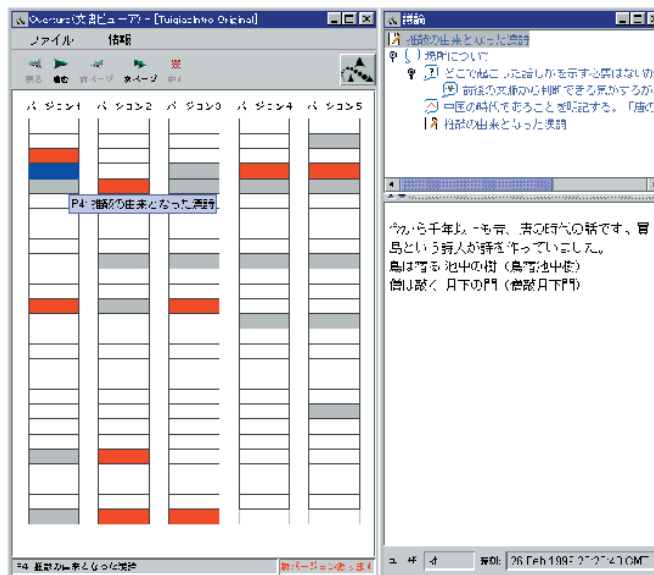


Figure 6. Visualizing the document creation process by TeamViewer.

P1. These links represent the following relationships among paragraphs, questions, and answers: Q1 challenges P1. A1 and A2 provide answers to Q1, and users select A1. Based on A1, users decide on and execute C1, which results in P1'. Once members are satisfied with this status of the document, they agree to revise the entire document. The system then configures version 2, which includes P1'.

### TeamViewer

The TeamViewer tool visualizes the document creation process. It provides users with a bird's eye view of the process by graphically displaying in one window how a series of versions of a document has evolved. In the display, paragraphs that have been changed or discussed appear in different colors. Figure 6 shows how TeamViewer visualizes the process. On the left side, each column represents a version of a document. A column consists of rectangles, each of which stands for a paragraph or



Figure 7. Architecture of TeamVCR.

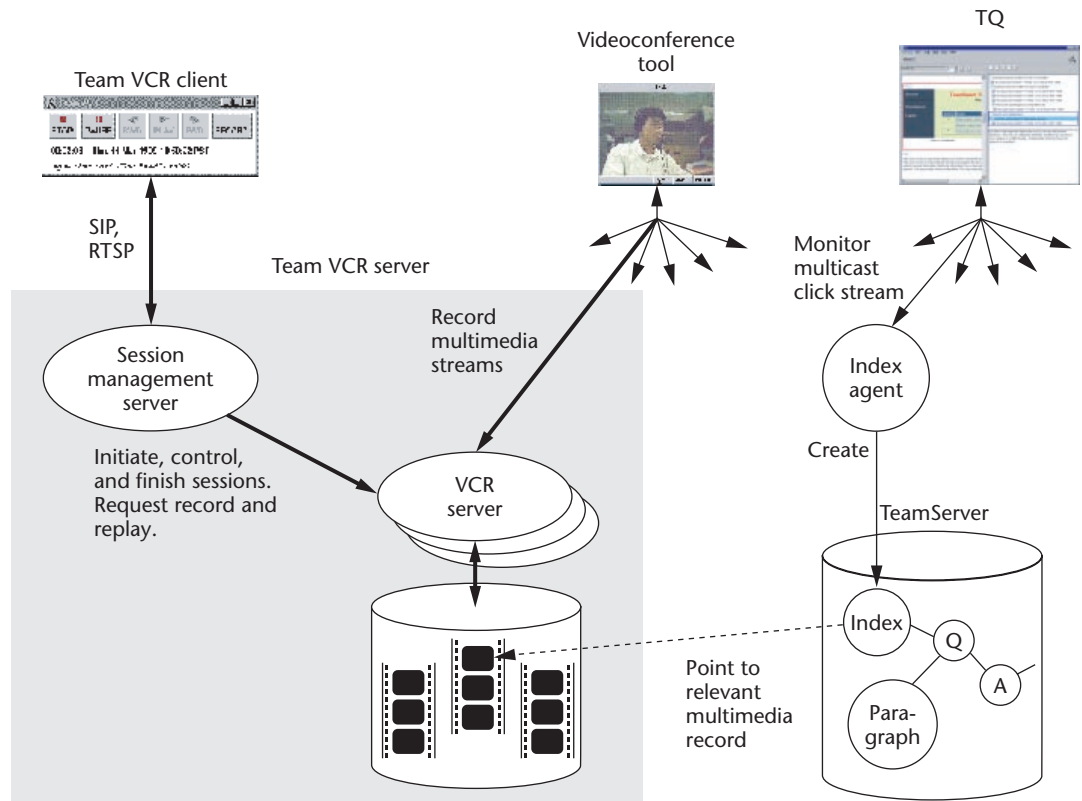


diagram. The red rectangles are paragraphs and ongoing diagrams that users have revised. The gray rectangles represent the discussions about the paragraphs. The paragraph that users selected appears in blue. Detailed information on the selected paragraph appears on the right.

### TeamVCR

TeamVCR provides a multimedia note-taking service. During a videoconference conducted using TQ, TeamVCR records the conference and automatically indexes its content. After the conference, TeamVCR plays back the recording at the user's request. The videoconferences are conducted using IP multicast tools such as vic (videoconference tool) and RAT (Robust Audio Tool).<sup>15</sup>

**Action-based indexing and recording.** TeamVCR automatically creates an index of the recording based on the times at which users selected or created paragraphs or discussion items during the conference. The system then generates links to the object of that selection or creation. Users can employ the index to replay only the recorded parts that pertain to paragraphs or discussion items of interest. To index the multimedia streams, TeamVCR creates index objects that

point to specific portions of continuously recorded streams without cutting or inserting indices into the records themselves.

TeamVCR generates the indices as follows. It continuously records the multimedia streams for entire sessions. When any of the participants creates or selects an information item with TQ, an index agent program automatically creates a Web object as an index of the multimedia streams. The index agent monitors the participants' actions based on information multicast by TQ about what commands the participants issue. TeamServer then links the index object to the information item that's created or selected.

**Architecture.** TeamVCR is itself a complex system that consists of TeamVCR clients, a TeamVCR server, and an index agent (Figure 7). The TeamVCR server consists of a session management server and VCR servers. A detailed discussion of the system appears elsewhere.<sup>16</sup> Here we briefly summarize the system architecture. The TeamVCR clients handle interactions with users and communicate with the session management server via the session initiation protocol (SIP)<sup>17</sup> and real-time streaming protocol (RTSP).<sup>18</sup> The session management server initiates, monitors, and controls the

recording and playing sessions, while the VCR servers record and replay multimedia streams. The session management server creates and controls the VCR server, or a program thread, for each session. The session management server mediates all interactions between users and the VCR servers.

Here's an example of a basic use scenario. Suppose that the TQ clients, TeamServer, the conference tools, and an index agent are already running. First, the user responsible for recording starts a TeamVCR client. Upon invocation, the client sends a request to the session management server with a resource description (such as the locations of the VCR server to be invoked, the index agent, and the communication channels) to initiate a recording session. The session management server checks whether the recording resources are available, and, if so, creates a VCR server thread. The session management server keeps monitoring the VCR server until the end of the session. Next, the session management server invites the index agent invoked by TQ to the session. The user then clicks the record button on the TeamVCR client and the VCR server starts recording the multimedia streams. The index agent keeps monitoring the click streams (that is, the streams of data about when and who selects or creates which paragraph) multicast by TQ. If the agent detects a creation or selection event, it sends a request to TeamServer to create an index object that contains the event information and links the index object to the relevant paragraph.

Currently, the resource description comes as a text file, which may be difficult to edit or modify. To improve usability, TeamSmart can cooperate with systems that provide collaborative virtual workspaces, such as CVW,<sup>19</sup> to specify TeamSmart resource allocations by assigning a virtual workspace to a particular project.

Typically, a five-party, 2.5-hour videoconference consumes about 600 to 700 Mbytes of disk space at about 10 frames per second in H.261 and pulse-code modulation (PCM) formats.

### **Trial results**

We used the global collaboration environment prototype, which included earlier versions of InterPOD and TeamSmart, for several projects in which meetings were held two or three times weekly over a period of about two years. The projects included global network operation and product development. We also used the environment to develop InterPOD and TeamSmart. We observed that the environment worked effective-

---

**Through trial use of our  
global collaboration  
environment,  
we found that it could  
effectively support  
collaboration.**

---

ly as a "group memory" for distributed teams.

Project teams consisted of network and software engineers and business development managers who had different nationalities (such as Japanese, Chinese, Taiwanese, American, Korean, and Canadian) and spoke different native languages. The number of team members ranged from 3 to 11. About every six months, team members from either side traveled to the other side's location and met in an InterPOD room to conduct extensive peer reviews using TeamSmart. InterPOD and TeamSmart were also installed at and used by other companies, US government agencies, and universities.

### **Process gains**

InterPOD helped meeting participants understand exactly what they were discussing by displaying various technical diagrams and pictures on computer screens or an electronic whiteboard. The system also displayed rough sketches written on paper shown through a video camera. These visual representations were easy to switch using Web browsers. This feature was particularly useful because these meetings focused on technical discussions.

InterPOD also let participants finalize agreed-upon changes to documents and diagrams during meetings. In addition, the multimedia records of meetings from TeamSmart allowed participants after the meetings to identify and work on discussions that had not led to conclusions. For example, we made it a custom to start meetings by checking each open discussion item listed in TeamSmart.

Meanwhile, TeamSmart let participants easily recognize and focus on topics of discussion during the meetings because an information item that participants selected on their TQ clients was

automatically highlighted on others' TQ clients even if they were in different locations. When using only videoconferencing tools without TeamSmart, we often had to tell participants at the remote sites which pages we were discussing. TeamSmart made this unnecessary because the synchronously shared TeamSmart clients highlighted the part selected by users and displayed the discussion context in one window.

The visual document evolution process from TeamViewer enabled late-joining participants to quickly catch up by providing them with a project overview and easy access to the information about the earlier part of the project. Also, project managers could quickly grasp the project's progress by using TeamViewer. The multimedia records accessible with TQ and the TeamVCR clients let them understand the current status of the project in detail. We made the project records for a one-year period accessible by TeamSmart. For one case, we had to access records taken six months previously to solve current problems.

In the InterPOD environment, participants that had different cultural backgrounds and spoke different languages could better understand each other by communicating via nonverbal language represented in various media. For example, videoconferencing tools helped the participants to know whether others understood and were satisfied with discussions by their facial expressions. TeamSmart also helped non-native speakers to understand what participants said by allowing them to replay the video records at their own pace.

#### Problems to solve

The trial results also revealed that a number of problems remain. Because there's a 17-hour time difference between Japan and California (16 hours in the summer), the system could only be used in a roughly two-hour period between 9:00 and 11:00 a.m. Japan time (5:00 and 7:00 p.m. Pacific Standard Time). Also, because multiple project teams used the system, the use of computer and network resources was concentrated during those times. Therefore, resource management and use scheduling are clearly important issues.

Organizing synchronous collaboration sessions on a global scale was a time-consuming and tedious task. No standard procedure existed to call together the participants in different time zones to begin a synchronous session. Also, participants had to share and precisely set many kinds of parameters (such as those for accessing work objects, setting up communication channels, and specifying

servers to use) to establish the session. Sufficient notification of changes in the conference schedule was also difficult to achieve. For these reasons, problems such as wasted time before beginning a conference and important participants not being available occurred.

If some participants couldn't fully use the services of the collaboration environment, they were at a great disadvantage in expressing opinions and influencing decisions made during meetings. This happened when, for example, project members were on business trips. The unfair differences made it difficult to form a consensus and resulted in decreased motivation for participating in the meetings. We need a mechanism or method for recognizing such differences in the environment and compensating for them.

#### Conclusion

Through trial use of our global collaboration environment, we found that it could effectively support collaboration. We also identified requirements for further improvements. The environment still needs a thorough and quantitative evaluation in terms of process gains and losses. It also needs more sophisticated project coordination features such as procedures for synchronous sessions and resource management. We must also address security and access control issues.

We developed the environment through global collaboration. We were almost overwhelmed by the barriers of time, distance, and cultural differences during this project. Even with advanced high-speed networking and sophisticated collaboration tools, global collaboration is difficult. However, we believe that holistic, systematic support with Internet and hypermedia technologies will enable full global competition and collaboration that best uses talents and resources worldwide. **MM**

#### Acknowledgments

We deeply thank Jeffrey D. Smith, founder and CEO of TeamBridge, for his contribution to the development and deployment of InterPOD and TeamSmart while he was working at NTT Multimedia Communication Laboratories (MCL). Without his passion and insights into cross-cultural issues, the project would not be possible. Special thanks are due to Eugene Kim for his elaborate work on implementing the InterPOD systems. Finally, we thank all the development team members who contributed to this project.

This article was based on the work that we did while we were at NTT MCL.

## References

1. J.D. Smith, K. Takahashi, and E. Liang, "Living Web: Supporting Internet-Based User-Centered Design," *Proc. Internet-Based Groupware for User Participation (IGROUP)*, 1999, <http://www.idi.ntnu.no/~ice/igroup/proceedings/smith.doc>.
2. H. Kaiya and M. Saeki, *A Supporting Tool for Face-to-Face Meetings to Develop Software Specifications* (in Japanese), Tech. Report KBSE93-13, Inst. of Electronics, Information, and Communication Engineers (IEICE), Tokyo, 1993, pp. 9-16.
3. G. Hofstede, *Cultures and Organizations: Software of the Mind*, McGraw Hill, New York, 1991.
4. M. Stefik et al., "Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings," *Comm. ACM*, Vol. 30, No. 1, 1987, pp. 32-47.
5. T. Moran, W. van Melle, and P. Chiu, "Tailorable Domain Objects as Meeting Tools for an Electronic Whiteboard," *Proc. Computer-Supported Cooperative Work (CSCW 98)*, ACM Press, New York, 1998, pp. 295-304.
6. J.F. Nunamaker et al., "Electronic Meeting Systems to Support Group Work," *Comm. ACM*, Vol. 34, No. 7, 1991, pp. 40-61.
7. J. Conklin and M. Begeman, "gIBIS: A Tool for Exploratory Policy Discussion," *Proc. CSCW 88*, ACM Press, New York, 1988, pp. 140-152.
8. J. Lee, "Sybil: A Tool for Managing Group Design," *Proc. CSCW 90*, ACM Press, New York, 1990, pp. 79-92.
9. K. Takahashi and C. Potts, *Tuiqiao: A Hypertext Tool for Requirements Analysis*, Tech. Report GIT-CC-94107, Georgia Institute of Technology, Atlanta, 1994.
10. M. Lamming and W. Newman, "Activity-Based Information Retrieval: Technology in Support of Personal Memory," *Proc. Information Processing 92, Volume III: Personal Computers and Intelligent Systems*, North-Holland, Amsterdam, 1992, pp. 68-81.
11. S. Whittaker, P. Hyland, and M. Wiley, "FiloChat: Handwritten Notes Provide Accesses to Recorded Conversations," *Proc. Computer-Human Interaction (CHI 94)*, ACM Press, New York, 1994, pp. 271-277.
12. L.J. Stifelman, "Augmenting Real-World Objects: A Paper-Based Audio Notebook," *Proc. CHI 96*, ACM Press, New York, 1996, pp.199-200.
13. G. Cruz and R. Hill, "Capturing and Playing Multimedia Events with Streams," *Proc. ACM Multimedia 94*, ACM Press, New York, 1994, pp.193-200.
14. C. Potts, K. Takahashi, and A.I. Anton, "Inquiry-Based Requirements Analysis," *IEEE Software*, Vol. 11, No. 2, March 1994, pp. 21-32.
15. V. Hardman, M.A. Sasse, and I. Kouvelas, "Successful Multiparty Audio Communication Over the Internet," *Comm. ACM*, Vol. 41, No. 5, 1998, pp. 321-327.
16. F. Cao, J. Smith, and K. Takahashi, "An Architecture of Distributed Media Servers for Supporting Guaranteed QoS and Media Indexing," *Proc. Int'l Conf. Multimedia Computing and Systems (ICMCS 99)*, IEEE Press, Piscataway, N.J., 1999, pp. 122-129.
17. M. Handley et al., *SIP: Session Initiation Protocol*, RFC2543, Internet Engineering Task Force (IETF), 1999.
18. H. Schulzrinne, A. Rao, and R. Lanphier, *Real Time Streaming Protocol (RTSP)*, RFC2326, IETF, 1998.
19. S. Jones, "The Collaborative Virtual Workspace," *Linux Journal*, Jan. 2000, pp. 32-38.



**Kenji Takahashi** is a senior research engineer and supervisor of the Information Sharing Platform Laboratories at NTT in Tokyo, Japan. His research interests include collaborative knowledge

creation, location-based computing and application service provision over Internet. He is also interested in the business modeling of the new technologies. He received a PhD and an MS in computer science from Tokyo Institute of Technology. He also served as the co-chair of the Collaboration Tool Working Group at CommerceNet.



**Eiji Yana** is a research engineer of the Information Sharing Platform Laboratories at NTT in Tokyo, Japan. His research interests include CSCW, especially focusing on physical devices, social protocol, communication barriers and speech acts, effect of combination of various media with multimodes, and application service provision over the Internet. He is also interested in the business modeling of the new technologies. He received an MS in computer engineering from Waseda University, Tokyo.

Contact the authors at Information Sharing Laboratories, NTT, 3-9-11 Midoricho, Musashino, Toyko 180-8585, Japan, e-mail {[takahashi.kenji](mailto:takahashi.kenji@lab.ntt.co.jp), [yana.eiji](mailto:yana.eiji@lab.ntt.co.jp)}@lab.ntt.co.jp.