interLiving

Designing Interactive, Intergenerational
Interfaces for Living Together. IST-2000-26068
http://interliving.kth.se

Technology Probes for Families

Deliverable 2.1 • 2002-01-31

Technology Probes for Families

Deliverable 2.1 • 2002-01-31

Michel Beaudouin-Lafon, Benjamin B. Bederson, Stéphane Conversey,
Björn Eiderbäck, Hilary Hutchinson

interLiving deliverable D2.1
Design: Bosse Westerlund

CID/NADA
KTH
SE-100 44 Stockholm
Sweden
interliving@nada.kth.se

http://interliving.kth.se

Cover illustration is from the second Swedish
family workshop, made by: Molly Bartling.
Photographs: interLiving researchers.
Type: Meta, Sabon
Print: pdf-version
Stockholm 2002-01-31

ISSN 1650-8009
ISBN 91-7283-261-4

Contents

1 Introduction

The goal of the InterLiving project is to use participatory design techniques to create technologies for families that foster intergenerational activities among multiple households. One challenge of using participatory design in this context is that many observation techniques are too invasive for family life: we cannot live with the families to observe them directly for extended periods of time, and we cannot rely exclusively on interviews to understand their activities.

We have designed *technology probes* to help us better understand the aspects of family life that we address in the InterLiving project. Technology probes build on the idea of partnering with users, which has a long history in the HCI community. Methodologies including contextual design (Wixon et. al., 1990), cooperative design (Bjerknes et. al., 1987), and participatory design (Greenbaum et. al., 1991) all allow adult users to work with technologists. More recently, Druin has extended this partnership to include children through the method of cooperative inquiry (Druin, 1999). We extended this idea to work with distributed, multigenerational families, which we believe will result in new methodologies as well.

The specific idea of a technology probe is more directly motivated by Bill Gaver's work with cultural probes – maps, postcards, disposable cameras, and other materials "designed to provoke inspirational responses from elderly people in diverse communities" (Gaver et. al, 1999). These probes were distributed to a group of elderly people, who returned them over the course of a month filled with informal information about their lives and cultures. We extended this idea to use technologies, rather than physical objects, to gain an understanding of communication needs, rather than cultural norms.

In the rest of this deliverable, we describe the rationale behind technology probes and we present the three technology probes that we have developed and are about to install in the families homes.

1.1 From cultural probes to technology probes

Cultural probes were introduced by Gaver as a novel design approach for settings similar to ours, i.e. when part of the problem is to discover what the problem is, and when traditional observation and analysis techniques that would help understand the problem are too intrusive. A cultural probe is an object with some aesthetic qualities that is given to the subject, with instructions on how to use the object and how to return it to the designer after it has been used. A typical example is a disposable camera with instructions to take pictures of "things you like" and "things you don't like". Cultural probes are low-tech and easy to manipulate, i.e. they do not require any special skill. At the same time they can be quite challenging as they require the subject to reflect on his or her activity in order to properly respond to the instructions. Taking pictures of things they like and things they don't like forces the subjects to decide what they like and what they don't like, why, and how to select the most significant ones.

Cultural probes were not designed to be used in a participatory design context and in many ways they are at odds with the participatory design approach. Cultural probes explicitly do not put the subjects in a design situation. They ask them to reflect on their activities and to reinterpret them through the (distorting) lens of each specific probe. They reveal as much as they hide, they create intimacy as much as distance between the designers and the subjects, they maintain the distinction between the designer (who has the power to create the probe) and the subject (who has the power to not use the probe).

Nevertheless the question remains open whether cultural probes can be used with participatory design. Indeed, we have used cultural probes in our early work with the families (see deliverable 1.1, Beaudouin-Lafon et al 2001), and we will continue to use them throughout the project. They have given us useful insights in the everyday life of the families and so far they have not adversely affected our participatory work with the families.

The notion of Technology Probe came up at the beginning of the InterLiving project when we were brainstorming how to introduce some technologies to the families to trigger ideas for the later stages of our design process. Initially, we had called them "seeding technologies" and we intended to use off-the-shelf technologies to see how families would react to them. Through participatory and iterative design, the seeding technologies would evolve into one or more proposals and become the outcome of the project.

We revised this plan though because it became clear that the seeding technologies would embody too much, too soon of what the project would end up producing. What we needed was a way to use technology to learn more about the families and use this to seed the design process. A technology probe is therefore similar to a cultural probe in that it aims at producing input to solve a particular design problem by attempting to elicit reactions that will inform us on various aspects of family life. Unlike a cultural probe though, a technology probe is not low-tech. It offers a glimpse of what future technologies might offer and therefore it challenges the subjects not only on how to respond to the instructions given with the probe but also to its design. Therefore we believe that technology probes bridge the gap between cultural probes and participatory design.

1.2 Designing technology probes

We designed the technology probes by following the same type of approach that we had used for the cultural probes: brainstorming, scenarios, mock-ups and walk-throughs. During the brainstorming sessions, the ideas that emerged all had to do with devices that supported a lightweight form of communication, as opposed to more intrusive forms of communication such as the telephone.

The first probe emerged quickly around the idea of sharing short, hand written messages. We had observed that many families have a board where they leave notes to each other, and it was a natural extension to be able to leave notes to remote family members as easily as with local family members. This led to the Message Probe, a board where users can share electronic Post-it notes.

The second probe came up starting with the idea of a simple mediaspace between different households. A full-fledged mediaspace, i.e. with live video and audio, seemed both technically difficult and looking too much like Big Brother, so the design evolved to sharing only still images. It seemed that it would be interesting to see how the families would accommodate a low-bandwidth system that could only exchange a few pictures a day. This led to the Video Probe, a picture frame coupled with a fixed camera to share snapshots of everyday life.

Since the first probe dealt with written messages and the second probe dealt with images, we decided to base the third probe on audio messages. The initial idea was to annotate physical objects with voice messages that would be played the next time someone would touch the object. This could be used for remote communication, for example by attaching a voice annotation to a postcard and sending the postcard to a relative. The current design, called the Audio Probe, uses PDAs to hold the voice messages and bar codes to tag objects.

Other ideas for probes were considered but not developed further, since we had limited time for this stage of the project. For example, it seemed natural to come up with a Touch Probe to complement the Message, Video and Audio probes. Another idea for which we did not find a satisfying implementation was the notion of twinned objects. For example, some superballs contain a device that blinks when the ball is used. Such a superball could be twinned with a second one that would blink when the first one is used. A child would see his superball blink, understand that his cousin is playing with his superball, and they could engage in remote play. Or the door of my grandmother's refrigerator could be twinned with mine so that when she opens the door of her fridge, the door of my fridge glows, giving a peripheral awareness that she is getting some food.

1.3 State of the project

Designing the technology probes proved more challenging than we had expected. First of all, the concept itself evolved and was refined as we were designing them, and our understanding of them emerged through the design process itself. Second, we realized that there were significant constraints on what we could implement, in terms of technologies we could use, cost, and robustness. We have now recruited a 7th extend family in the United States. We the current total of seven families, each with up to three households, and so deployment is actually a significant effort – especially considering it needs to run unattended 24 hours a day, 7 days a week. Reliability and robustness are particularly critical and hard to achieve especially since all our probes rely on large area networks, which are notoriously difficult to operate in a bulletproof way. In addition, the probes must include logging and data collection facilities so we can gather usage data for further analysis.

As of January, 2002, we have developed the first two probes introduced above fully and made significant progress on the third (audio) probe, we have acquired most of the equipment to deploy them in the families, we have visited the families to find out where they would like the probes installed and which other households in their families they would like to involve. We have also started to get the various DSL or cable ISP subscription to install the probes. We have already deployed the first probe (Message Board) at the three households of the family in the United States, and we are planning to deploy that probe in the families houses in Sweden and France by February, 2002. This is behind schedule according to the project, however it must be pointed out that technology probes are more ambitious than the seeding technologies that we had originally envisioned, and that they will speed up the rest of the design process by giving us more information about families.

2 Message Probe

Today’s families are more geographically distributed than ever. Children attend schools far away from their parents; grandparents may live in a different country than grandchildren. Letters, email, instant messages, and telephone conversations can help keep remote family members up to date on major family events, but the patterns of everyday life are often missed. In addition, these communication techniques are all either strictly synchronous or asynchronous, and each suffers from some bothersome complications.

Letters and email are asynchronous activities that don’t provide any remote awareness to the participating parties about one another. Letters are addressed to only one household and require a trip to the mailbox or post office. Email requires computer and Internet competence, time wasted dialing up and logging in, and isolation from collocated family members. Both also assume that participants are able to read and write.

Instant messaging and phone conversations are synchronous activities, requiring both parties to be present to communicate at the same time, and are not persistent – once you log off or hang up, there is no record of the interaction. Like email, instant messaging requires computer knowledge and literacy, and can lead to wasted time and isolation. Phone conversations can be expensive and are limited in the number of participants.

To address and explore these problems, we have developed a Message Probe, a software program designed to be used with a digital writing surface and display where family members can write or draw notes to each other, much like paper sticky notes. Local and remote family members can have boards in multiple locations (e.g. home, work, school), and all are networked together so that all the messages posted show up on all the probes in real time. As a technology probe, the Message Probe was designed to be adaptable to a variety of uses and scenarios so that family members could experiment and discover the most valuable ones (see Design Issues below).

The probe can function synchronously, with two or more family members communicating at the same time, or asynchronously, with family members checking their probes periodically for new messages. This second function allows family members to see messages that may be totally unrelated to them (e.g. “Pick up milk after work”), but help give a sense of daily events. In addition, events that might not otherwise be communicated to remote family members become common knowledge and topics for future discussions (e.g. “Soccer practice at 4 today”).

The probes are connected only to a small set of family members, removing the need for complicated setup and remembering names, addresses, or buddy lists. There is no mouse or keyboard – just a pen - and literacy is not required. Finally, the probe hardware can be embedded in social areas of the home such as a family room or kitchen, and can be made portable via wireless technology.

2.1 Related Work

The Message Probe design encompasses work from a variety of fields, which we describe below. The technology is heavily influenced by shared whiteboard projects in CSCW and commercial communication software such as instant messaging. As a device for families, our work builds on growing research into technology for the home. In an effort to keep remote family members connected in a meaningful way, we were influenced by research in remote awareness. Our user-interface design is based on past experience with zoomable user interfaces. Finally, our desire to involve our users in the design process comes from experience in participatory design and lead to the concept of technology probes.

The idea of a networked, digital writing surface has a long history in the CSCW literature through numerous implementations of shared whiteboard technologies. From early work such as Wang's Freestyle and Xerox's Tivoli projects to more recent applications including Flatland and Rekimoto's Pick-and Drop, these whiteboards have provided innovative features for synchronized, networked communication in the workplace (Mynatt et. al., 1999);(Pederson et. al., 1993);(Rekimoto, 1998);(Wang Laboratories, 1989).

The shared whiteboard idea quickly gravitated from dedicated devices to standard PC desktops and from synchronous activity to asynchronous messaging via virtual notes. Lotus' TeleNotes application was among the first projects to recognize the need for shared, asynchronous workplace communication by supporting virtual desktop sticky notes (Whittaker et. al., 1997). Greenberg's Notification Collage is a more recent example that supports more advanced communication by allowing colleagues to post pictures and converse via live video in addition to posting notes to one another (Greenberg et. al., 2001).

In the commercial arena, virtual note applications are ubiquitous in the PC and PDA markets. TurboNote+ is a shareware program that allows Windows PC users to create onscreen sticky notes that can be delivered over the Internet via IP or via email (TurboNote+, 2001). Electric Pocket has developed an application called BugMe! Messenger that allows users of Palm OS-equipped PDAs to exchange handwritten, text, and graphic notes to other PDA's or via email (BugMe!, 2001).

In the home, asynchronous communication via notes and more popular email soon gave way to synchronous communication via instant messaging (IM) and chat applications such as AOL's Instant Messenger and Internet Relay Chat (IRQ) (AOL Instant Messenger, 2001);(Oikarinen et. al., 1993). Recently, both research and commercial efforts have been made to identify and exploit additional remote awareness information available during IM and chat sessions.

Nardi et al. have identified a number of uses for IM in the workplace that fall outside of traditional communication, including negotiation of availability and sustaining social connections (Nardi, 2000). Researchers at Fujitsu are experimenting with augmenting IM on cell phones to include icons indicating emotions and text memos (Mitsuoka et. al., 2001). Yahoo's Messenger IM service has recently integrated Web cam functionality to allow users to see each other via live video (Yahoo Messenger, 2001). In the chat arena, traditional text-based applications have been augmented with avatars equipped with a selection of gestures and expressions (Kurlander et. al., 1996) and abstract shapes that convey information about a user's activity graphically (Viegas et. al., 1999).

Our Message Probe borrows features from all of these previous projects and products, but the combination results in a unique application: first, it is meant for home use by a fixed set of users; second, it is meant to be used with an embedded or portable writable tablet display; third, it can be used both synchronously and asynchronously; fourth, it is meant to support remote awareness; fifth, it makes use of a persistent, graphical, zoomable user interface; and finally, it is a technology probe whose design is being guided by the families using it.

This first difference is perhaps the most significant. Designing technology for the home is far different than for the workplace. People have goals other than improving productivity or efficiency when using technology in the home. For instance, the HomeNet study at Carnegie Mellon found that interpersonal communication (e.g. email) is more popular than information or entertainment applications (Kraut et. al., 1998). Home users are also likely to be less tolerant of ugly, utilitarian designs and hardware or software failures. Finally, they are far more diverse, in every sense of the word, than the target audiences of many technology products (Scholtz et. al., 1996) – people of all ages, interests, and abilities are potential users.

Despite these differences, households and designers of household technologies continue to treat home technologies such as the PC as work-related devices. The social spaces in the home where family members spend most of their time interacting with one another (e.g. kitchen, den) are separated from work spaces (e.g. "home offices") where PC's are kept (Mateas et. al., 1996);(Venkatesh, 1996). Thus, technologies such as email and instant messaging that home users appear to want to use to stay in touch with remote friends and family can have the unwanted side-effect of keeping these users isolated from their collocated family members, perhaps even causing declines in psychological and social well-being (Kraut et. al., 1998).

To avoid this problem of isolation, technologies can be embedded in more social areas of the home, or made lightweight and portable so they can be carried and shared where people wish to use them. As part of the Disappearing Computer Initiative, the InterLiving project seeks to develop technologies that do exactly this. The evidence for home users' desiring such technologies is compelling. In a recent study by MediaOne Labs, home users given portable, wireless, Internet-enabled tablets cited portability and the ability to multi-task as the nicest features of the tablet as compared to a PC (McClard et. al., 2000).

Interval Research's Casablanca project used ethnographic field studies and consumer testing of design concepts to gauge home users' interest in new technologies for the home (Hindus et. al., 2001). One of these devices, a prototype simulation of a ScanBoard, provided similar functionality to the Message Probe. Users could post messages using a writable LCD screen networked to other family members, as well as scan in photos, drawings, and other paper artifacts to be digitized and shared. Users appreciated the ability to keep in touch with or monitor family members in a fun, low-cost, simple way, and specifically liked the ability to share via scanning and to communicate in more expressive ways.

The Casablanca project also revealed that in addition to the more obvious goals of simple, low-cost devices to use to keep in touch, users wanted devices that respected privacy, did not create new obligations, and offered multiple communication modes. The Message Probe addresses all of these criteria with its communication mechanisms. Note posting can be done synchronously, like IM or chat, or asynchronously, like email. Privacy is ensured because only known family members are connected to the network and there is no monitoring aspect. There is no obligation to reply immediately or at all to a message.

In addition to supporting both synchronous and asynchronous communication, we were also interested in providing remote awareness for family members separated by distance, making frequent face-to-face meetings impossible. This is especially relevant because it motivates the Video Probe as well as the Message Probe. Work in this area, such as the Xerox PARC's Media Space project, and the Portholes, Peepholes, and Thunderwire applications, has focused on helping remote colleagues work together and maintain informal connections using video, audio, and icons to create virtual media spaces (Bly et. al., 1993);(Dourish et. al., 1992);(Greenberg, 1996);(Hindus et. al., 1996).

In later work, the AROMA project sought to find more abstract representations for mapping remote activities into local displays (Pederson et. al., 1997). IBM's Babble software augmented a traditional chat interface with "social proxies" – small digital dots that moved in and out of a circle to indicate participation in a conversation (Erickson et. al., 1999). Recently, research in this area has spread to the home and is becoming especially popular as the baby boom generation ages. For example, Mynatt's Digital Family Portrait was designed to help adult children check in on aging parents in an unobtrusive manner via active icons on a picture frame (Mynatt et. al., 2001). Likewise, the persistent, real-time updating of colorful notes and drawings on the Message Probe provides a sense of presence to remote family members.

Another difference between our Message Probe and many other communication technologies is its persistent, graphical, zoomable organization of messages. This user interface design grew out of a number of years of experience with designing zooming user interfaces (ZUI's). Unlike most chat and IM applications, which are text-based and transient, we used the Jazz toolkit (see Implementation below) to help users arrange and navigate graphical messages written with a digital pen in a large zoomable space (Bederson et. al., 2000).

A recent study by Bederson and Boltman indicates that the animated transitions between viewpoints in this sort of zoomable environment improve users' abilities to reconstruct information spaces (Bederson et. al., 1999). The Family Message Board aims to help users organize and find their messages by allowing them to arrange their messages in a persistent space. Users can zoom in and out of the space and drag notes in and out of a default grid arrangement to design their space of notes in a meaningful way.

2.2 Design Issues

Our main goal in designing the Message Probe was to keep it as simple, adaptable, and open-ended as possible. As a technology probe, the design needed to allow families to find innovative and unexpected uses for it without being encumbered by restrictive functionality.

We decided to build a messaging device based around virtual notes because of the universal popularity of paper sticky notes for informal family communications and reminders. We would lose the very nice feature of being able to stick notes on anything anywhere in the house, but gain an unlimited supply of notes and the ability to share them remotely with others. As much as possible, we wanted to simulate the experience of writing real paper notes, moving away from standard desktop computing and towards a single, small, embedded, portable, device that users could view and write on with a digital pen.

This design goal was reinforced by results from the MediaOne web tablet study, which showed that users found small, portable keyboards and handwriting recognition were difficult to use with the tablet (McClard et. al., 2000). The Message Probe only takes free-form input from a single pen. We also chose to stay away from added features like voice or video annotations, as supported in the Notification Collage (Greenberg et. al., 2001), or the ability to scan in real paper, as supported in the Scanboard (Hindus et. al., 2001), for two reasons. First, we didn't want to complicate the device or introduce features that might threaten families' perceptions of privacy. Second, as a technology probe, we wanted the device to encourage families to suggest such features on their own if they really wanted them.

The interface for the Message Probe provided two interesting design issues. First, with the potential for multiple remote family members to be viewing, manipulating, and writing on their devices simultaneously, there were a number of usability and synchronization issues to consider. Not only do family members at multiple locations share the message space, but also multiple family members at the same location share a single message creation and viewing device. As a result, there is really no sense of individual ownership in the space. Second, with obstacles such as novice computer users and busy families already encumbered by cell phones, email, etc., we needed to make the probe as simple, fast, and natural to use as possible. Both of these considerations led to multiple design iterations after testing within our own research group and with families.

We chose to implement a bulletin board-like interface rather than one involving mailboxes or separate visual areas for notes to or from individual users, topics, or devices. All users share control of the notes in the message space. Anyone can write on, move, or delete any note in the space, regardless of who created it. All actions except for drawing are delayed on remote devices until the device is idle for 10 seconds to prevent remote actions from interfering with someone interacting with a device locally.

Tapping a virtual notepad located in the lower right-hand corner of the display creates a new note. New notes are sent to all the devices in the family and are displayed in the same location on all devices. By default, new notes are arranged according to their creation time in a grid demarcated by a gray background. New notes appear in the lower right corner of the grid and older notes are scaled to progressively smaller sizes and pushed to higher rows in the grid (see Figure 2).

We did not want to force any kind of organization of notes on users, but needed some way of arranging notes initially and of managing the space required to display a large number of notes. We chose to arrange them in a grid according to their time of creation because creation time is the only note feature that is certain. Any one of the multiple family members that share a device can create a note, and any other family member, locally or remotely, can later modify it.

In our initial design (see Figure 1) when a note was created, a margin near the top of the note was stamped with the name of the device that created it (chosen by each family location when the device was installed) and the date and time it was created. This information was meant to provide a sense of remote awareness and timing when the board was used asynchronously. In later iterations (see Figure 2), we removed this information, deciding instead to allow family members to develop their own conventions for identifying note authors and times, and to encourage them as design partners to make suggestions for such features if they wanted them.

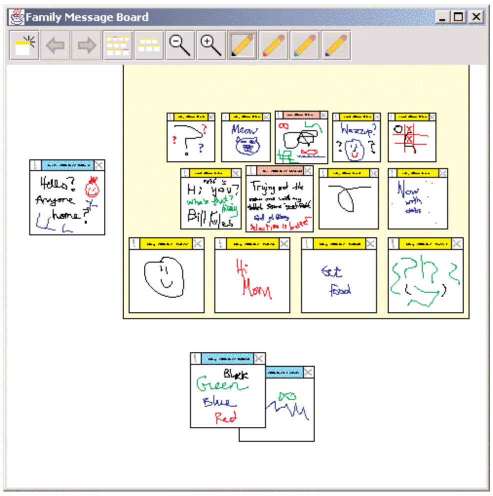

Figure 1. Initial Design.

Figure 2. Current Design.

Our initial design also included icons in the top corners of the notes for emphasizing and deleting notes. Tapping an "!" icon in the top left corner of the note caused it to become slightly larger and changed the background color of the top margin. Tapping an "X" icon in the top right corner of the note deleted it. We later chose to remove these features for the same reasons we removed the time and author information. As a result, the notes are completely blank, just as a paper sticky note would be.

Organization and personalization of notes beyond the default placement is entirely up to users. Notes can be dragged out of the message grid anywhere in the message space. Notes can also be dragged back into the grid, where they resume their place in the time-based order. As notes are added or removed from the grid, the grid reorganizes itself to fill up empty space. This design choice means that spatial consistency is lost as notes are moved in and out of the grid, perhaps making notes harder to find in the grid.

However, we believe that users removing notes from the grid to organize the notes themselves will achieve spatial consistency. Without the automatic reorganization, the grid would rapidly fill with holes, wasting space. Thus, the design does not preclude the idea of organizing notes by topic, creator, ink color, etc; rather it leaves this decision up to the family.

This design also allows for some interesting, and perhaps unexpected interactions, which adds to users' sense of remote awareness. Two users can draw on the same note at the same time. There is also no erase functionality – users simply add to existing notes, create new ones, and move them around. Like paper sticky notes, we believe crossing out errors or simply starting over is less effort than finding an eraser. Of course, family users may feel differently, and suggest adding an eraser.

Although the arrangement of the notes in space is the same for all devices, each device controls its ability to create new notes, its pen color for writing, and its view of the messages via zooming. This allows multiple users to interact with the message space at the same time without disrupting the interactions or views of other remote users. Tapping a note with the pen makes it the active note and zooms in on it so it covers the whole screen. The user can then draw on the note. Tapping outside of a note zooms the space out to provide a view of all the notes. Remote users do not see this zooming and are free to zoom their view of the space in other ways.

In our initial design, we implemented a toolbar at the top of the screen with six navigation buttons to enable local users to view the message space in various ways, independently of remote users. Left and right arrow buttons navigated through notes in the order they were created. Zoom in and out arrow buttons animated the camera view to focus on more or less of the message space. A "Show Recent" button zoomed the view so that only the two most recent rows of messages in the grid were visible. A "Show All" button zoomed the view so that all the messages in the space fit in the device window. There was also a button to create a new note and four buttons to select pen colors.

However, in our later design, we found that we could dispense with the toolbar in favor of simpler, more natural, less "computerish" interactions. As mentioned above, a notepad in the bottom right corner of the message grid is used to create new notes. When a note is selected, a palette of colors appears next to it instead of in the toolbar. The palette disappears when the view is zoomed out. Zooming in on a note is achieved by tapping on it. Zooming out is achieved by tapping outside the note. Moving left and right is achieved by tapping on the note to the left or right of the currently selected note. We did not implement a "Show Recent" function, but this could be done. We also removed the window border and window resizing options. Family users are thus free to organize and interact with simple pen gestures and no windows, icons, menus, mice, or keyboards.

2.3 Implementation

The Message Probe software was built using Java 2 and three Java-based toolkits: the University of Maryland's Jazz, Sun's Java Shared Data Toolkit 2.0 (JSDT), and Interbind's XIO, all available for download on the web (Interbind, 2001);(Java Shared Data Toolkit, 2001);(Jazz, 2001). The Message Probe hardware requirements include a writable LCD display, such as Sony's Slimtop (Sony VAIO, 2001) or Wacom's PL Series (Wacom PL, 2001) pen tablets, and a Windows-based PC. The software will also work with a mouse or graphics tablet, such as a Wacom Graphire, and a monitor.

We used the Jazz toolkit for the spatial arrangement of notes in the Message Probe. Jazz provides a two-dimensional scene graph structure for organizing graphical objects in a large, zoomable canvas. Objects are viewable and zoomable through a virtual camera and can be translated, rotated, and scaled. Notes in the probe are arranged on the canvas in a grid as they are created, with older notes shifted and scaled to less prominent grid positions. Individual notes and areas of the grid can be zoomed in or out, and notes can be dragged out of the grid and placed in arbitrary locations on the canvas.

We used JSDT to support communication between the multiple probes scattered among the various households of a distributed family. JSDT provides support for collaborative, networked applications by supporting full-duplex, multicast communication. Multiple clients can join and leave communication sessions in order to exchange and share information. Each instance of the Message Probe is a client that joins a well-known session established by a central server, who is also a client in the session. A separate JSDT registry process keeps track of all the clients in the session.

Each time a client creates or modifies a note, JSDT sends information about this message to all other clients and the server using a reliable, TCP-based communication channel. When a client receives this note information, it creates or updates its local copy of the note and updates its display to reflect the change. When the server receives this note information, it stores it locally so that new clients who join the session later can request the current notes in the system. The receipt of new or modified note information is synchronized at each client so that only one is processed at a time in the event that multiple remote devices are active.

Finally, we used Interbind's XIO to provide robustness in the event of a server failure. XIO is a Java package that can be used to read and write Java objects to and from XML files. Users create templates describing the objects in a class that they want written out to an XML file. XIO uses the template, a serialization manager, and the class's JavaBeans setter and getter methods for these objects to create the file when writing and to recreate the objects from the file when reading. The server for the Message Probe uses XIO to write out information about all the notes to an XML file once every minute (if there are changes) and when the server exits normally or abnormally. If the server crashes, all of the note information can be retrieved from the XML file to recreate the message space.

2.4 Logging and Data Collection

The Message Probe has two kinds of logging facilities. On the server side, note information is logged in XML format as described in the Implementation section above. All the information necessary to recreate each message (location, time of creation, pen strokes, etc.) is stored in the file so that the message space can be recreated in the event of a server crash. On the client side, each client maintains a plain text, space delimited file of actions taken by that client, including drawing strokes, pen color changes, zooming, and note creation. Each log entry contains the name of the client who took the action, the time of the action, what the action was, and a unique note id if the action involved a particular note. Consistent time information across computers and time zones is maintained by having all time stamps originate from the server. This allows us to create unique note id tags composed of the client name and the creation time of the note. We will use this log information to analyze trends and frequencies of various events.

3 Video Probe

The Video Probe is designed to support a lightweight, asynchronous mode of communication by making it easy to share images of people and objects among multiple households of a family. By installing the Video Probe in the homes of the InterLiving families, we want to confront them with a device that is very different from anything they have used before, to observe how they invent its use and to analyze how it affects their everyday life.

The InterLiving families already use photos to communicate across generations. The Orange Family, for example, takes pictures of their children or of family events, then selects some of them with the help of the children and sends them together with a letter to the grandparents. However, we do not seek to support this type of use of picture-based communication in this exact form. The Video Probe is a fixed device that is installed in the house, not a substitute for a portable digital camera. Its goal is to support background awareness of other households, to create a sense of "being together" and to share the daily emotions that constitute the fabric of everyday life.

The Video Probe must be extremely easy to use so that it appeals to every member of the family, including children and grandparents. We want the interface to be as invisible as possible so that it becomes as natural to use as checking the mailbox when getting home or pulling a chair to join a discussion. Since the goal is for the Video Probe to be used over a long period of time by a broad set of users, our aim is for the system to work essentially without any explicit user intervention. We envision the Video Probe to take pictures when it "sees" something interesting, and to automatically display the pictures received from the other sites.

3.1 Related Work

Since it is based on video and images, the Video Probe draws from the area of research known as mediaspaces (see Mackay, 2000 for an overview) and more specifically the awareness devices such as PortHoles (Dourish & Bly, 1992). PortHoles links together multiple sites by displaying a mosaic of still images of various locations at each site (typically the offices and commons of a workplace) taken at regular intervals. It makes it easy to monitor the activity in this synthesized shared mediaspace and gives a sense of being together.

The Video Probe differs from PortHoles and other similar systems in two ways. First, it does not generate a stream of pictures taken at regular interval, i.e. a very low frame rate video, but it takes pictures of things it deems interesting. Therefore it requires that the user be active in a certain way for images to be sent and is therefore not a passive device like PortHoles. Second, it does not display the images as a mosaic, which is easy to consult quickly, but only sequentially. The images can be browsed by the user explicitly or scroll on the screen automatically. This too requires a more active role from the user.

Another line of related work is the concept of Ambient Display. One of the first examples of ambient display was Weiser's Wire (Weiser, 1996), a long piece of string hung from a DC motor placed in the ceiling, driven by the activity of the local Ethernet network: fewer packets and the string turns slowly, many packets and it swirls around. Other examples include work at the MIT Media Lab (Ishii & Ullmer,1997), Georgia Tech's Digital Family Portrait (Mynatt et al., 2001), CMU's Information Percolator (Heiner & Hudson, 1999), and Xerox PARC's stock fountain (Avzav, 1999). The common point is that a physical device's activity is used to represent information that may be useful to the users and can be perceived through peripheral awareness.

The Video Probe shares with ambient displays the idea of visualizing information, namely that something potentially interesting has happened, and displaying it in a non-intrusive way. However, the display of the Video Probe is less abstract than other ambient displays, e.g. the wire or even the family portraits. Also, the Video Probe is more interactive than most ambient displays: the user can explicitly browse the images, and can easily generate the "interesting events" that the Video Probe captures. In fact we expect that the Video Probe will generate this kind of interaction between distant family members.

3.2 Use Scenarios

During the design of the Video Probe, we worked with a few scenarios to help us focus on the purpose of the probe and to simplify the design as much as possible. These scenarios were drawn from our own family lives, as we did not want to involve the families in this part of the project in order to surprise them with the probe.

Scenario 1

The grandparents live far away from their children and grandchildren. Flowers are blooming, so the mother puts them into a vase and places the vase in front of the Video Probe. The probe grabs an image and sends it to the grandparents. Later in the afternoon, the grandparents aim their Video Probe towards the window to show that it is gloomy and rainy at their place.

Scenario 2

The kids come back from school and they know their mother has made an appointment at the hairdresser for them later in the afternoon. Before leaving, they go in front of the Video Probe and have it take a picture of them pulling each other's hair. The grand parents, when they see the picture show up on their probe, don't know exactly what it is about. When the children come back from the hairdresser, they go back to the Video Probe and have it take a new picture of themselves. They act as if they were sad for their lost hair. The next morning, as the grandparents run into the new picture and understand what happened, they give them a call to cheer them up.

Scenario 3

Grandmother is getting pretty old, and her daughter, who is now retired and lives across town, is often worried that something may have happened to her. She calls ten or twenty times a day to check on her, to remind her to take her pills, etc. With the Video Probe, they have developed a code to keep each other aware of their respective activities. They have a number of objects that they put in front of the probe, such as a small car when the mother goes out or a teapot when the grandmother has lunch. When they have visitors, they often play jokes with the probe by putting random objects in front of it. They still call each other several times a day, but there conversations are very different from the inquisitive tone that they used to have. The Video Probe has become a part of their everyday life, a ritual that punctuates their daily activities.

3.3 Design Issues

Like the Message Probe, our main design goal for the Video Probe was to keep it simple. Ideally, a user should be able to walk to it, stand or put an object in front of it and have the system take a picture. The system must be able to work autonomously, with minimum user intervention. The system should not take unwanted pictures, e.g. when the camera is moved or when someone is just passing by it. The user should also be able to easily browse the snapshots that have been received from other sites.

We decide to use a flat display so the probe looks like a picture frame. At first, we considered using a touch screen: the screen would be divided in four areas: left and right for browsing images, top and bottom for two other, as yet undefined, functions. In addition to the cost, the problem with a touch screen was that the sensitive areas would not be labeled (we want the picture to use the whole screen real estate), and that it would get dirty very quickly. An alternative was to add physical buttons on the four sides of the screen, but this was not easy to do in a way that was both robust and aesthetically pleasing. Finally we opted for a small remote control. Being able to control from a distance seemed more natural for watching pictures on a screen, and supported group interaction better than direct interaction on the screen. In order to avoid loosing the remote control, we decided that it would be tethered to the screen.

The next issue was the content of the display. We worked on several designs that would display an overview of the pictures as well as individual pictures full-screen, but we could not see any real advantage to this added complexity so we decided that the pictures would be browsed one at a time, like in a photo album. We then realized that the users would be confronted with a large number of pictures fairly quickly (especially if the automatic picture grabbing did not work as well as expected). This lead us to the concept of aging: pictures would progressively look older and disappear altogether after a few days. In addition to reducing the number of pictures to browse, we hypothesized that it would encourage using the system, especially when there would be no more pictures in it after a few days out of use. It also emphasized the temporary aspect of the probe. Later on we introduced the notion of album described in the implementation section to allow saving images and avoid losing them. This resulted in two additional commands: add and remove from album, bound to the up and down keys of the remote control.

The main challenge of the design however was to come up with a simple way to take "interesting" pictures automatically. We did not want to engage in complex image processing since we did not have the time (or the competence, for that matter), and more importantly because we wanted the system to be somewhat predictable so that, after a few attempts, users would know what to do to, e.g., take a picture of themselves. The final design is described in detail in the next section on implementation. In short, a picture is taken if something changes significantly in the field of view of the camera and then stays still for a few seconds. The system maintains a notion of current background (called the reference frame) to avoid taking pictures of the background after an object of interest has appeared, then disappeared from the field of the camera. We also added a command to explicitly take a picture when we realized that our approach did not work well to take pictures of babies or animals, who are difficult to maintain still in front of a camera.

Finally, we had to solve the problem of multiplexing two modes of operation: grabbing pictures and browsing images. When a user walks up to the probe to grab images, it is very likely that he will stay still in front of the screen for long enough that the system will grab a picture of him. We considered a layout were the camera shoots sideways with respect to the screen so that someone in front of the screen would not trigger the grabbing of pictures, but we realized that users will want to monitor on the screen what image was being taken. We also considered a design with two screens, or the use of mirrors, but none was worth the added complexity. Our current solution involves a combination of implicit and explicit mode switch and is described below. Experience will tell us whether it works well in practice.

3.4 Implementation

This section describes the current implementation of the Video Probe.

3.4.1 Hardware and software platform

The hardware requirements to implement the Video Probe are:

- a computer with hardware-accelerated graphics;
- a monitor (a flat LCD monitor is preferred so it looks like a picture frame);
- a video camera (typically a web-cam);
- a remote controller (e.g. Keyspan's Digital Media Remote);
- a network connection (cable or DSL).

At the time of this writing, the first prototype runs on the following platform:

- a PC running Linux;
- a 17" monitor;
- a Philips ToUCam Pro USB web cam, with its video4linux driver;
- no remote controller - the keyboard is used to simulate the remote controller;
- a LAN connection.

For portability and efficiency reasons, the Video Probe uses OpenGL (http://www.sgi.com/software/opengl) for rendering graphics. OpenGL supports transparency, arbitrary geometrical transforms, lighting and other effects that may prove useful as we evolve the design.

The first prototype does not require a hardware-accelerated graphics card, but we plan to use various graphical effects in the future, so it is important that the hardware supports them.

The Video Probe also uses the VideoSpace toolkit (Roussel, 2001) to capture, transform and display video streams. VideoSpace supports various input sources, such as video cameras, recorded video files, or video streams over the network. It supports various video sinks such as display windows, files and network streams. Finally, VideoSpace provides real-time filters such as image-differencing or edge detection.

3.4.2 User Interface

The Video Probe uses a full screen window, thus taking the whole monitor real-estate. It has two modes: the mirror mode and the browser mode. By default it is in mirror mode and the monitor is entirely blank. Mirror mode is used to capture images, while browser mode is used to view the images received from the other sites.

Mirror mode

The mirror mode acts as a selective mirror that only displays the images captured by the camera when "something happens", i.e. when changes are detected in the video stream. When motion is detected, the live video stream fades in progressively (figure 4). To look like an actu-

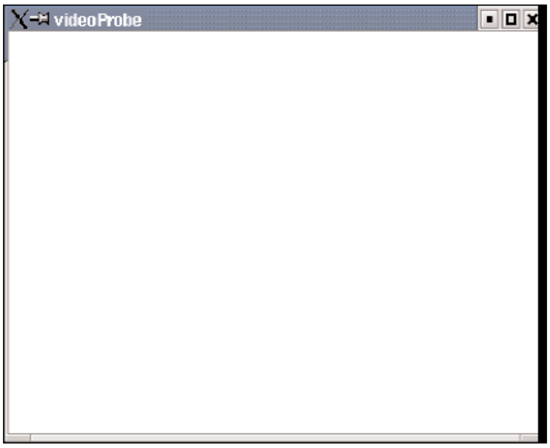

Figure 3: The blank screen when the Video Probe is idle.

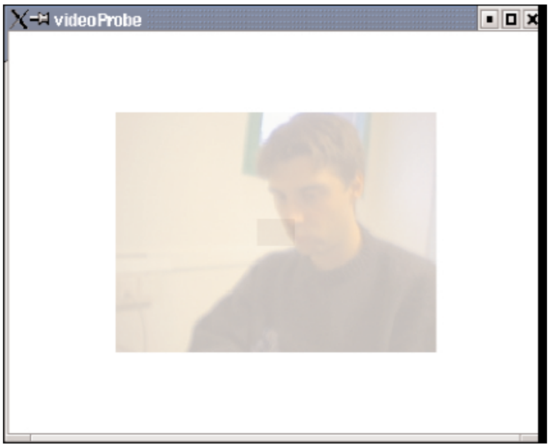

Figure 4: When someone is detected, the video recording fades in.

al mirror, images are reversed left to right. According to our informal tests, this makes it easier for a person to move herself or the object she wants to show to get a proper framing.

Once the mirror is on, if the person or the object that has been detected is still for a certain amount of time (3 seconds in the prototype), the Video Probe takes a snapshot. The snapshot is then shown, non-reversed, for another 3 seconds. The Video Probe then goes back to mirror mode and is ready to grab another image.

When the Video Probe detects that the person or object has been removed from its view, the live video image fades out and the display returns to the idle mode of figure 1.

Mirror mode is designed to make it easy to grab pictures. The blank screen and video fade-in are designed to encourage using the system: someone casually passing in front of the Video Probe will trigger the mirror, but will not capture and send a new image. However she will probably notice the fade-in and may decide to stop to actually capture and send an image, or look at the recorded images by going into browser mode. In other words, the goal is to achieve a balance between the design goals of a device that stays in the background while at the same time encouraging its use.

Automatically taking the "right" picture is of course a difficult endeavor. We do not want to rely on sophisticated image analysis because we do not anticipate all the uses of the system. Therefore we only rely on a crude form of motion detection based on measuring the differences between successive images. We also maintain what the system considers as the current reference image, i.e. the image of the background when nothing of interest is in front of the camera.

In order to capture an image, a person or object must stay still in front of the camera for a few seconds. If motion is detected, the Video Probe waits until the stream is stable. This avoids taking pictures when someone is in front of the camera with the aim of capturing an image

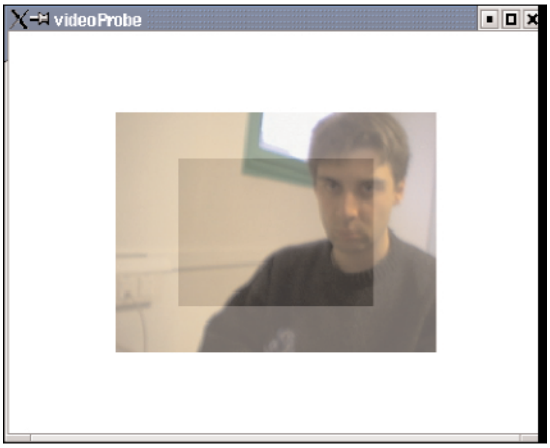

Figure 5: When the person is still, a black translucent rectangle appears and expands until it reaches the borders of the video image.

of herself, but is not ready yet to grab the image. By waiting until person is still, we allow her to better control the captured image. Furthermore, this approach avoids taking too many unimportant or unwanted pictures, e.g. when someone is passing quickly in front of the camera. On the other hand, it may be difficult to capture images of small children who, in our experience, have a hard time staying still in front of the camera.

When the Video Probe decides to capture an image, i.e. when the video stream is steady enough, the interface provides a clue about the time remaining before the snapshot is actually taken. As soon as the user is still, a translucent black rectangle appears at the center of the screen and starts to expand (figure 5). When the rectangle overlaps the whole image, the image is captured. If the person or object moves while the rectangle is expanding, the rectangle disappears and the Video Probe starts again waiting until a steady video stream.

In order to detect whether something new has appeared in front of the camera, the Video Probe continuously grabs images and compares them to a reference image that contains only the background. Grabbed images may differ from the reference image for one of the following reasons:

1 someone or something has appeared in the field of view of the camera;
2 lighting conditions have changed;
3 the camera has moved (even by a small amount).

Ideally, the Video Probe should ignore cases 2 and 3 and only take pictures when case 1 is detected. In the current prototype, we have not attempted to distinguish between these cases. We think that changes in lighting conditions (e.g. turning a light on or off) and changes in the camera position and orientation may be interesting to remote users, and we wait for users' feedback before trying to refine the system.

Nevertheless, even with this simplification, using a reference image leads to the problem of updating it to deal with cases 2 and 3. For example, lighting conditions change when the sun is rising, and grabbed images will always differ from the reference image, leading to continuously capturing and sending images. Since we do not want to force users to explicitly set the reference image, we need to update it automatically. To update the reference image, the Video Probe compares the last snapshot (i.e. the last image that was captured and sent to remote sites) to the previous one. If three successive snapshots only differ by a small margin, the system infers that the reference image is not valid anymore, and the last snapshot is used as the new reference image. The threshold used to detect whether a snapshot is a new reference image is lower than the one used to detect if something new has appeared in front of the camera. This lowers the chances of taking an object of interest or a person as a new reference image. Furthermore, in order to maximize the chances that this algorithm works well in practice, we will instruct the families to install the Video Probe in a location where it is unlikely that a reference image includes an object of interest or a person. Our long-term tests in the laboratory have shown that this approach should be robust enough, at least initially.

The mirror mode of the Video Probe is best described with a state machine (figure 6). Transitions between states occur when conditions are true. Conditions are on the first line of the label of each transition. “diff img > thrx” states that a condition is true if the difference between the last grabbed image and the image “img” is greater than a threshold “thrx”. “img” can be “ref”, the reference image, “succ”, the last grabbed image, or “last”, the last snapshot. When a transition occurs, the actions described in the second line of the label of each transition are executed (e.g., taking a new reference image, arming a timer...). The implementation is based on this state machine, which makes it easy to test alternative designs or add new features.

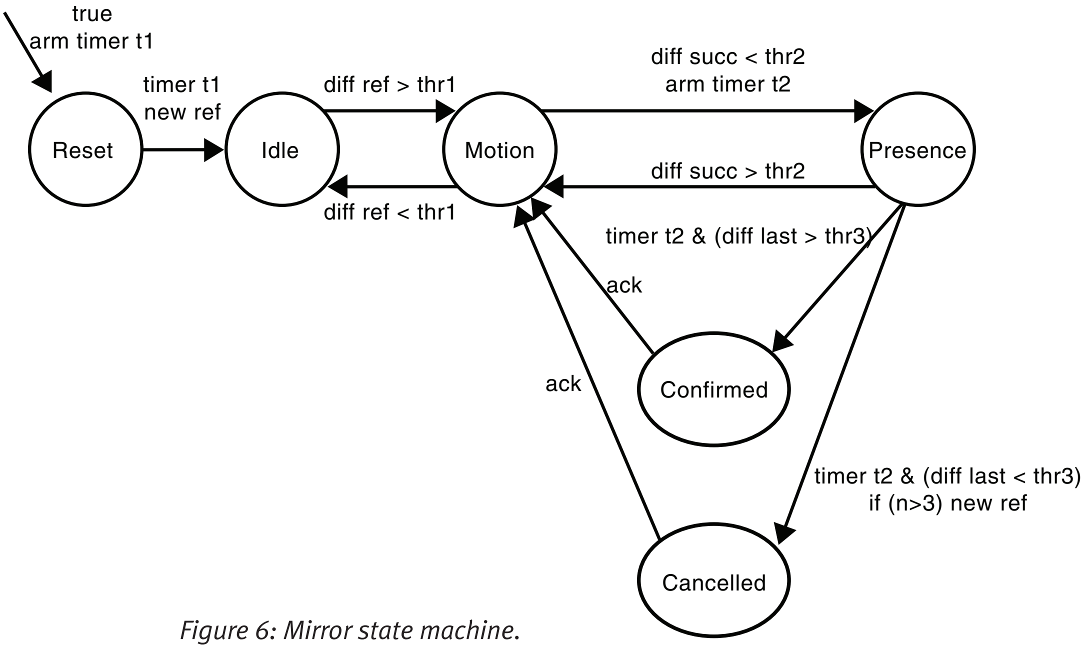

Figure 6: Mirror state machine.

Browser Mode

The second use of the Video Probe is the browser mode, which is used to browse through local and remote snapshots. When a snapshot is taken, the Video Probe automatically stores it in the local browser and sends it to remote browsers. Currently, the user cannot cancel or undo the sending process once the Video Probe has decided to take a snapshot. We did not want to make the interface more complicated than it is, and we want to find out whether users will require this feature or not.

To enter the browser mode, the user presses the “left” key of the remote controller, and the Video Probe shows the last received snapshot. By pressing the “left” and “right” key, the user can navigate through images in a reversed chronological order. Each image is slightly rotated: this gives a more informal look, and acts as a feedback of the browser mode being active.

Images have a limited lifetime: they disappear after a few days (3 in the prototype). To give a sense of the age of each image, the Video Probe applies a filter to make them look older over time. The aging mechanism first degrades the colors of the image, by transforming them progressively into black and white images. It then increases the luminance to progressively remove contrast so the image fades out (figure 7). When the image is almost completely white, it is removed altogether.

The user can save images in an album by pressing the “up” key, and remove it from the album by pressing the “down” key. Images in the album are not rotated, and they do not age. When an image is put into the album, it recovers its colors if it had already started to age; when an image is taken out of the album, it starts to age again. Therefore, removing an old image from the album still makes it available for the next 3 days. Finally, in order to maintain time consistency, all images, whether they are in the album or not, are viewed in reverse chronological order.

While in browser mode, mirror mode is re-entered when pressing the “right” key while the most recent snapshot is shown. Since the person viewing the image is likely to be in front of the camera, her image will immediately fade in. This is consistent with the time-order of snapshots: after the most recent snapshot comes the “present”, i.e. the live video stream from the camera.

While in browser mode, the Video Probe does not take snapshots. Thus, if someone forgets to quit mirror mode after viewing the pic-

Figure 7: Pictures age: colors and contrast progressively disappear.

tures, no snapshots would be taken until someone enters mirror mode. To avoid this problem, the Video Probe goes back to mirror mode automatically when the user is not in front of the camera, i.e. when the images grabbed by the camera match the reference image, and when no navigation commands have been input for a while (3 minutes in the prototype).

Exchanging snapshots

The first prototype used a peer-to-peer mechanism that allowed the video Probes to exchange snapshots. However this approach does not scale well and it is not robust: in case of network failures or when a computer is down, snapshots could be lost.

The current prototype uses a separate server that runs at our lab on a machine permanently connected to the Internet. Each Video Probe sends their snapshot to that server on a dedicated port. When a Video Probe is not connected to the Internet, it stores its snapshots locally. As soon as the connection is up, it sends them to the server, which stores them and sends them to the other video Probes as soon as they are available. The Video Probe communicates with the server using the "POST" method of the HTTP protocol [HTTP Home Page, 2001]. Using HTTP makes it easier to run through firewalls, because it uses a well-known port that is usually allowed.

Privacy concerns have not been addressed in the probe, although it would be easy to use public key encryption to exchange the images between the server and the video Probes. The images could be saved encrypted on the server so that the researchers would not be able to access them without the families' authorization.

3.4.3 Possible evolutions

In the process of the design of the Video Probe, we tried to create a system as simple as possible. At the same time, a large number of ideas came up that we may implement in the future according to the users' feedback. Here is a non-limitative list of features that we considered and left out for the first version:

- Use a key of the remote control to explicitly take a snapshot.
- Use a second camera that can be easily moved around, and use the button on the camera to take snapshots explicitly (the fixed camera would only work in automatic mode).
- Give a sense of activity by using special effects, such as accumulating successive images in order to display a motion blur.
- Automatically enter a slide-show mode at random times when the system is idle, to grab users' attention.
- Use two monitors, one for the local node (grabbing images) and one for the remote nodes (browsing images).
- Use a tablet-LCD screen (as used for the Message Probe) and / or a microphone to support written and oral annotations.

3.5 Logging and Data Collection

The Video Probe logs data both on the server and clients side. On the client side, user actions are logged together with a time stamp and a node identifier:

- taking and sending a snapshots;
- switching between mirror and browser mode (including automatic switch);
- browsing the snapshots forward / backward;
- putting a snapshot into the album / removing a snapshot from the album.

On the server side, the images are time stamped and archived and the clients' logs can be retrieved and consolidated. The format is similar to that of the Message Probe. The consolidated logs allow us to reconstruct the complete sequence of activities among multiple sites and study the patterns of use.

We have considered adding remote control facilities in order to, e.g., adjust some thresholds or time-outs without having to physically go to the families. We anticipate that the settings that work well in our lab will not work as well in the actual settings. For example, we may find out by looking at the logs that the system is taking too many snapshots because it does not identify a new reference image after the camera has been moved. The current version of the prototype does not support such remote control.

4 Audio Probe

The audio probe is a shared audio annotation service based on use of PDAs (Personal Digital Assistants). The goal is to provide a means to attach voice (audio) annotations to any object as a probe to understand what people want to communicate about. One family member can attach a sticker to a physical item, record a message on a hand held PDA, and by means of a bar code reader identify the message with the sticker. Another family member can read the sticker with a bar code reader thereby trigger a replay of the attached message.

4.1 Motivation

The motivation for the audio probe is divided into two subsections; one motivating the development of the audio probe as such and the other one motivating the particular implementation we have chosen.

4.1.1 Motivation for the probe

We are designing the technology probes to encourage the families to think about new and different technologies, other than the standard desktop computing environment, that might be useful in their lives. We have designed the message probe to investigate means of co-operation and scheduling of activities within a family, possibly situated in different households. We have designed the video probe to investigate means of sharing visual artifacts among members in the family. The intention with the audio probe is to investigate if/how audio and handheld computers, maybe enhanced with wireless communication, can be used in family communication situations.

4.1.2 Instantiation

The instantiation we have chosen for an audio probe is the *voice sticker*. The voice sticker is a shared audio annotation service based on use of PDAs. A family member can attach a sticker to a physical item, record a voice annotation on a hand held PDA, and by means of a bar code reader identify the annotation with the sticker. Another member can read the sticker with another bar code reader, thereby triggering a replay of the attached message on their own PDA or computer.

One of the goals is to understand what people want to communicate about. Another goal is to trigger the users in the process of inventing, or describing, new tools and means of communication.

4.2 Related Work

Much previous work has influenced the development of this probe. Some of the major influences include Alan Kay's Dynabook, precursor of today's modern notebook computers. In the 1960's, Kay envisioned a PDA to be carried around by all people at all times (The ArtMuseum 2001, Kay 1996). Research on ubiquitous handheld computers at EuroParc (Lamming 1993), and at a more specific technical level, WebStickers (Ljungstrand & Holmquist 1999) is also related. We also draw on more recent experiences with barcode sound stickers from some of the InterLiving partners' involvement in the just finished EU Esprit Experimental School Environments project KidStory (Bayon 2001, Stanton 2001).

At the development level we have been inspired by Occam's Razor (Encyclopedia Britannica 1995), by William of Ockham, one of the most influential philosophers of the 14th century and a controversial theologian. This principle says than one should make things as simple as possible by cutting off the unnecessary. Another major influence is Design Patterns (Gamma et al 1995, Eiderbäck 2001) that discusses how to make software as effectively as possible. Finally, a major influence is eXtreme Programming (Beck 2000) which also, as Occam, promotes "the simplest thing that could possible work", but also articulates how to involve clients in the whole development process.

4.3 Design Issues

There are several issues related to the design of this probe. First, we have to consider the functionality of the probe. Second, we must consider how to design the probe in such a way that all family members, independently of their age or previous experiences with computers, can use it effortlessly. Third, the probe must be as robust as possible, since we plan to let one family use the probe for quite a long time and it is neither practical nor desirable if the probe needs maintenance by us every now and then. All these concerns lead to issues in both hardware and software.

4.3.1 Hardware issues

As a mode for associating the annotations with physical objects, we discussed RFID tags, bar code tags, or numerical codes typed or written directly onto the device.

Platform issues

We wanted a slim, easy to carry solution with ability to handle voice, network connections, and a bar code reader. These requirements lead us to choose a hand held computer platform. The platform of our choice also needed to be easily programmed, ideally allowing prototype development on an ordinary computer. These considerations led us to the decision to develop this probe on the Compaq iPaq PocketPC platform. The IPAQ has a screen keyboard, built-in audio recording capabilities, and handwriting recognition for input.

Annotation media
We decided to investigate the usefulness of using bar codes for attaching annotations to an item. Therefore, the probe must also be equipped with a bar code reader and either ready-made bar codes or a means to easily print out new bar codes.

What about bar codes?
We decided to investigate the usages of bar codes further. For practical reasons we decided that the bar codes should be pre-printed and handed to the families in a binder.

What about radio tags?
We also considered the usage of radio tags. The radio tags are admirable since we more easily, than with bar codes, could attach behaviour to the items they are attached to. But we decided not to use them since they are both technically more complicated and much more expensive than bar codes.

What about tags with just a number?
We also considered simplified versions of the bar code approach. One approach could be to used provide simple stickers with just a number printed on each of them.

What about no tags at all?
A solution based on no tags at all is technically desirable since less hardware, as bar code readers, has to be handled. But this solution also has some severe drawbacks, which will be discussed in section 4.3.3.

4.3.2 Networking

There are a number of choices for how to communicate the contents of the voice stickers between host computers and other PDAs . One way or another, each IPAQ needs to synchronize with a host computer to download and upload the voice recordings to a network database. The quick and dirty way is with the ActiveSync software that comes with the IPAQ that allows you to sync via a wire connecting the IPAQ to the USB port of your computer. IPAQs can talk to each other via infrared. You can also equip them with wireless cards if giving families a wireless network is feasible. Yet another alternative is to base the communication on some widespread telephone protocol, as GSM or RPGS.

4.3.3 Annotations and Associations

As a mode for associating the annotations with physical objects, we discussed RFID tags, bar code tags, or numerical codes typed or written directly onto the device. The latter has some advantages since users don't have to stick things to their physical objects or deal with scanning devices. They can simply tell the recipient of the object what number to type into their device to play the sound, or else make it available via a nice interface to the database. The main problems with this approach are that it is not obvious that an item has an annotated message, the family member that annotates the item must tell other members that the particular item has an annotation and its number in the shared annotation database, and a family member that wants to listen to the annotation must first find out both that it is annotated and the annotation's number, and eventually type the number in order to listen to the annotation. Maybe this simple solution is good enough for our purpose of triggering ideas from the families, so it could be worthwhile to try it, or at least consider it further before deploying the probe to the families. As a mix in between we could use pre-numbered tags where the user could stick a tag to an item, record a voice message, and finally identify the tag with the number in order to complete the annotation. This latter solution has the advantage that we do not need a bar code reader and that the user does not have to remember the number of the association between the tag and the item.

4.3.4 Design Issues and Scenarios

In the basic setting, a family member should be able to annotate items with audio by means of recording a sound and attaching corresponding identifying bar code. Thereafter, any user that possesses an audio probe can read the sticker by using a bar code reader to trigger the replay of the attached audio annotation.

An alternative, if you do not want to or if it is not possible to attach a sticker to an item, is attaching a sticker to a replicate or a proxy. Then the replicate could be passed to another member, carrying the information without physically moving the original item.

We expect the audio probe to be used in a variety of situations. Examples are children telling a story about a precious item to her/his grandparents, diaries consisting of both text pictures and annotated audio, a journal of pictures with attached describing audio, and attaching descriptions to important objects in the home. In general one can use the probe as a means to dress messages to other members in attractive clothing.

General design issues

We have considered a lot of various possibilities and required functionality of the audio probe. Below follows a table with possible functions in the left column. In the right column the status of the function in the first version of the audio probe is stated.

Function	Included in the first version
Record a sound and annotate it to an item	yes
Listening to the annotation of an item	yes
Appending sound to a voice sticker	no
Replace the sound on a voice sticker	no
Remove the sound of a voice sticker	no
Copy the sound to another voice sticker	no
Passing a voice sticker to another person	yes
Ordering the voice stickers	no
Graphical or animated content attached to a voice sticker	no

The most essential functions are further described in the next section.

Scenarios

In this subsection, some of the typical design issues are more practically investigated by means of scenarios. The scenarios are both used as requirements and as guidelines while implementing the probe. In order to just focus on the most essential the scenarios are described in as short and idealised, but still informative, form.

Scenario 1, recording a message

One of the fundamental and necessary operations in the voice sticker is to record a message and annotate a bar code to it. This can be done in at least two different ways.

Recording alternative 1

- The user starts by putting a (voice) sticker on an item.
- Use the bar code reader to identify the bar code
- Record the message that should be associated with this particular bar code.

Recording alternative 2

- The user starts by recording a sound.
- Then he/she puts a bar code on an item.
- Finally the bar code is read by the bar code reader

Since we strive for an as simple as possible handling of the interface both of these alternatives will be provided in the probes to the families.

Scenario 2, Replaying a Message

- Read a voice sticker with the bar code reader.
- If the voice sticker has an attach message play it.

Scenario 3, Harvesting Existing Bar Codes

- One family member is wandering about with the voice sticker probe.
- While he/she comes across an interesting item he/she first looks at the item and then searches for a bar code attached to the item.

- If a bar code is found he/she uses the bar code reader to read its code and thereby the annotated message will automatically be replayed for him/her.
- Thereafter the family member alternatively stops “the harvesting” or continues with the next item.

We could easily envision more advanced scenarios where for instance the person that harvests bar codes wanders about and collects several annotations before, later on, replaying all of them one after the other. But by the principle of simplicity we postpone this and other more advanced harvesting techniques.

Scenario 4, passing an Item with a Bar Code to a Family Member in another Household

Given an annotated item, one could pass the item to a family member in another household. The other member could use his voice probe to read the message. A scenario could be:

- Annotate an item
  - o The annotation and its identification are delivered by means of Internet communication to the shared database where it is stored.
- The item is sent by (old fashioned) mail to another household.
- A member in the other household receives the item.
- This member starts the voice probe and reads the bar code of it
  - o The voice probe searches for an annotation with the current bar code identification. First a search on the local PDA is made and when not found the shared database is consulted from where the annotation is (automatically) downloaded.
- The voice probe plays the attached message.

We can easily envision alternative ways of passing annotations to other members, as for instance instead of passing the real item (perhaps a precious, fragile, or just heavy one) we could pass a proxy. But in this version of the voice sticker we will only provide the basic functionality described in the scenario.

Scenario 5, the Storage

The audio probe could also be useful to annotate items stored in a warehouse, attic, cellar, or alike. Often one has a set of hardly accessible boxes and to check the content of one in the bottom one has to remove all the ones on top of it before it contents could be revealed.

- One person packs a box with items.
- This person annotates the box before placing it in the storage.
- Another user searches for the box.
- The person that searches listens to the annotations on each until he finds the right one. Then the boxes on top it are removed and the search box is brought out of the storage.

A similar scenario is also applicable for a museum although in this situation most of the publicly available items not are hidden in boxes.

Further in the particular situation in this scenario a version with radio tags would have been more useful, since that would make it more easy for us to search for the right one.

Scenario 6, playing the Game of Finding Things (the Treasure map)

Finally we give a slightly different scenario, more belonging to entertainment. By means of two different audio probes we envision a game of searching for a certain place or item.

- One person, or some people together, builds a weave, or map, of interrelated annotations leading to the big treasure.
- The players equipped with one audio probe each set out to search the treasure by searching for annotated items and listening to the guiding instructions attached to them.
- The first to find the treasure is the winner and the prize is to set the next round.

This scenario is included to point to another type of expected usage of the probe.

4.4 Implementation

General Consideration

We decided to deploy the simplest thing that could possibly work to the families and wait for responses from them before deploying new versions.

On the other hand we also have to investigate the technology further and see if we for example could exploit new protocols used by cellular phones.

The first version(s) of the probe

To make it as simple as possible to handle the audio probe we strive to exploit the iPAQ’s standard facilities as making a audio recording by just pushing its special purpose recording button. For comparison, we also constructed our own recording facility. In both cases we store the annotation on a file locally on the iPAQ. At certain times the annotation is uploaded to the shared annotation database. This synchronization could be made in a number of different ways. At the moment we are investigating two different approaches namely:

- Direct synchronization by means of socket communication with obvious benefits since the database is kept up to date at all times. However this approach requires wireless Internet connection.
- Synchronization when the iPAQ is put in its cradle. The main benefit is that we do not require a wireless connection. However this approach is technically more complicated to implement and requires

some activities by the user of the probe. Further, another drawback is that a scenario where two different probes are used requires that they are synchronized via their cradles from time to time, which limits the ability for one user to make annotations while the other listen to them.

Exploiting the iPAQ's audio facilities

While an audio note is taken by means of the iPAQ's ordinary audio recording facility a file with the audio content is created. While the bar code reader is activated we try to match the time of this event with the creation time of the audio file, by searching for the ones that are nearest in time to each other. Thereby we can guess which bar code that matches a certain recording. The problem is that this approach is not 100% reliable, especially since the audio recording could take place before the reading of the bar code or vice versa. It would also technically be smoother to be able to control everything in the application code. Another problem is that this approach is quite hardware dependent since we rely on a hardware button. However, the latter problem could be solved. On, for instance, a laptop we could write our own software and just dedicate one key on the keyboard to emulate the iPAQ's hardware button.

Making our own design

Another alternative is to design our own recording facility. The benefits are that we have full control, potentially faster performance, and get a system that easily could be transferred between audio probes running on different hardware. The main drawbacks are that it probably is less natural than using the hardware button and that it requires special skills from the user to start our facility.

At the moment we are evaluating the two different approaches before deciding if just one of them or both should be deployed to the families. We could also try to combine the two approaches and use the hardware button directly in our own programs. But this is technically more complicated and the efforts and costs for that could not be motivated at the moment.

4.5 Design Variations

The Voice Sticker on a Laptop

We developed one version of the voice sticker on a laptop with an attached bar code reader. The purpose is to smoothly investigate the techniques, discuss ideas, and try out technical solutions before deploying the version on a PDA.

As you see in the figure to the left the voice sticker runs in its own window simultaneously with other applications.

The Voice Sticker on a Handheld Computer

The version of the voice sticker on a handheld computer is the one that we plan to first deploy in the families.

The version in the picture to the left is a development version of the voice sticker running on an iPAQ with an attached wireless network card and bar code reader.

The Voice Sticker on a Cellular Phone

To overcome problems with the lack of Internet access and investigate how the simplest solution could be developed, we also considered the voice sticker on a cellular phone. At this stage, the main purpose is for us to investigate the feasibility of this approach. However, if it works smoothly we will also consider deploying this version in some families.

In the picture to the left you see a mockup of the voice sticker running on a Cellular Phone.

4.6 Desired Results

The desired results that we expect from this probe are:

- Demonstrating feasible technology to the families.
- Trigger ideas from families.
- Solving certain design issues concerning PDAs and disappearing computers.
- Input for further development.
- Finding the limits of the present technology.

5 Conclusion

As described in the introduction, we have spent the technical efforts of this first year of the InterLiving project focusing on building "Technology Probes". They are explicitly designed to support emergent use patterns. That is, instead of making it very easy for users to perform a few basic tasks, but difficult to do other tasks that we did not imagine – we purposefully designed interfaces that support a broad set of tasks. In this manner, we expect to learn from our family design partners more about how they communicate, and how new communication techniques can support their needs, and change the way they do communicate.

We currently are actively working on deploying the three technology probes into the homes of our family partners. Perhaps not surprisingly, this is turning into a substantially difficult task. While we know that we will have to train the family members to use the probes, it turns out there are several issues to face before we can get to that. The issues related to deploying the technology probes are: physical design of probes, getting each household connected to the Internet with a high speed always-on connection, finding a place in each household that is suitable for the technology as well as the family. Finally, we have to address the fears of some family members who are unfamiliar with custom high-tech devices.

In terms of physical design, we feel that it is important to deploy the technology probes to the families in a style that does not feel like a traditional computers. Since our software does run on regular computers, we decided to concentrate on finding a physical computer that would feel most comfortable in a home environment. We found that the Apple "cube" computer satisfied this requirement the best. It is small, very attractive and unusual to look at, and also is completely silent. Because Apple OS X now supports Java 2, we were able to run the Message Probe on it directly. And because OS X is built on UNIX, it has been easy to port the Video Probe to run on it. However, Apple has recently discontinued the production of this "cube" computer, so we had to search far and wide before we were able to locate enough computers to purchase. We have now received these computers, but the process of acquiring them significantly delayed us.

The next practical issue related to deployment is getting the families homes connected to the Internet with an always-on high-speed connection. In both Sweden and Paris, we have had significant difficulties. Some family members are not in the region served by DSL. There is a long wait for installation from the telephone companies, and some of the services they offer have changed during this process. We are currently getting a combination of variations of DSL, ISDN, and cable modem connections to each of the households for each of the 3 extended families in Paris and 3 extended families in Sweden. We havd cable modems installed and the first probe deployed in the family in the United States.

We have been visiting the households of each extended family to come up with a good location for each probe. This turns out to be quite difficult in some cases as some of the houses or apartments are very small, and there is not much room for an extra computer. In addition, sometimes, the place where there is room is a place where some family members do not frequently visit. In addition, we have to find a location which is accessible by the high speed Internet connection, and doesn't require cables to be located in a spot that can be tripped over. We are currently finalizing decisions about where each probe will go in each household.

Finally, we are working with the family members – through workshops, and individual interaction – to address the fears that some have about technology in general, and about our probes in particular. A few older participants expressed concern about breaking our flat displays and tablets. One grandparent, in particular, did not want to even touch the tablet and pen. Only after seeing the rest of her family try it out, and with much convincing did she finally pick up the pen and try it – naturally discovering that it was not difficult to use, or particularly fragile.

In sum, while we are pleased with our progress with the technology probes, we have discovered that the practical matters of making a probe robust enough for independent family use, setting up of households, deployment, and training are substantially more difficult than we first anticipated. Nevertheless, we expect to deploy the Message Probe and Video Probe early in 2002 and will start collecting information from the families use of them.

6 References

AOL Instant Messenger Home Page, http://www.aol.com/aim/home.html.

Arai, T., Dietmar, A. and S. Hudson. “PaperLink: A Technique for Hyperlinking from Real Paper to Electronic Content.” Proceedings of CHI ’97. (1997), ACM Press, 327-334.

The ArtMuseum, http://www.artmuseum.net/w2vr/archives/Kay/01_Dynabook.html, (2001).

Avzav, Sentor. The Financial Post (Toronto, ON) Sowing PARC’s next revolution, April 21st 1999, Page C13 - Internet Stock Fountain.

Back, M., Cohen, J., Gold, R., Harrison, S. and S. Minneman. “Listen Reader: An Electronically Augmented Paper-Based Book.” Proceedings of CHI ’01. (2001), ACM Press, 23-29.

Back, M. and J. Cohen. “Page Detection Using Embedded Tags.” Proceedings of UIST ’00. (2000), ACM Press, 159-160.

Barrett, R. and P. Maglio. “Informative Things: How to Attach Information to the Real World.” Proceedings of UIST ’98. (1998), ACM Press, 81-88.

Bayon, V. “Classroom Components for Story Creation and Retelling.” In Augemented Space for Shared Story Telling. (2001), KidStory: Esprit Project 29310, 115-163.

Beaudouin-Lafon, M., Druin, A., Harvard, Å., Lindquist, S., Mackay, W., Plaisant, C., Sundblad, Y., Westerlund, B. “Cooperative Design with Families”, InterLiving Project Deliverable 1.1, CID/NADA KTH, Stockholm, ISBN 91-7283-183-9.

Beck, K. eXtreme Programming explained: Embrace Change. ISBN 0-201-61641-6, Addison-Wesley, 2000.

Bederson, B. “Audio Augmented Reality: A Prototype Automated Tour Guide.” Proceedings of CHI ’95. (1995), ACM Press, 210-211.

Bederson, B. and A. Boltman. “Does Animation Help Users Build Mental Maps of Spatial Information.” Proceedings of Info Vis ’99. (1999), IEEE, 28-35.

Bederson, B., Meyer, J. and L. Good. “Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java.” Proceedings of UIST 2000. (2000), ACM Press, 171-180.

Bjerknes, G., Ehn, P. and M. Kyng. Computers and Democracy: A Scandinavian Challenge. Aldershot, UK: Alebury, 1987.

Bly, S., Harrison, S. and S. Irwin. “Media Spaces: Bringing People Together in a Video, Audio and Computing Environment.” CACM, 36 (1). (1993), ACM Press, 28-47.

BugMe! Messenger Home Page, http://bugme.net/bugme-messenger.

Chalfonte, B., Fish, R. and R. Kraut. “Expressive Richness: A Comparison of Speech and Text as Media for Revision.” Proceedings of CHI ’91. (1991), ACM Press, 21-26.

Crampton Smith, G. “The Hand That Rocks the Cradle”. I.D., May/June 1995, 60-65.

The Disappearing Computer Initiative, http://www.disappearing-computer.org.

Dourish, P. and S. Bly. “Portholes: Supporting Awareness in a Distributed Work Group.” Proceedings of CHI ‘92. (1992), ACM Press, 541-547.

Druin, A. “Cooperative Inquiry: Developing New Technologies for Children with Children.” Proceedings of CHI ’99. (1999), ACM Press, 592-599.

Eiderbäck, B. “Object Oriented Frameworks with Design Patterns for Building Distributed Information Sharing”, PhD Thesis (2001), ISBN 91-7265-240-3, http://www.nada.kth.se/~bjorne/th.

Encyclopaedia Britannica, cited at http://www.cco.caltech.edu/~maronj/text/razor.html, (1995).

Erickson, T., Smith, D., Kellogg, W., Laff, M., Richards, J. and E. Bradner. “Socially Translucent Systems: Social Proxies, Persistent Conversation, and the Design of ‘Babble’.” Proceedings of CHI ’99. (1999), ACM Press, 72-79.

Fitzmaurice, G., Ishii, H. and W. Buxton. “Bricks: Laying the Foundations for Graspable User Interfaces.” Proceedings of CHI ’95. (1995), ACM Press, 442-449.

Gamma, E, Helm, R, Johnson, R, and Vlissides, J. “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1995.

Gaver, B., Dunne, T. and E. Pacenti. “Cultural Probes.” Interactions 6 (1). (1999), ACM Press, 21-29.

Gorbert, M., Orth, M. and H. Ishii. “Triangles: Tangible Interface for Manipulation and Exploration of Digital Information Topography.” Proceedings of CHI ’98. (1998), ACM Press, 49-56.

Greenbaum, J. and M. Kyng. Design at Work: Cooperative Design of Computer Systems. Hillsdale, NJ: Lawrence Erlbaum, 1991.

Greenberg, S. “Peepholes: Low Cost Awareness of One’s Community.” Proceedings of CHI ’96. (1996), ACM Press, 206-207.

Greenberg, S. and M. Rounding. “The Notification Collage: Posting Information to Public and Personal Displays.” Proceedings of CHI ’01. (2001), ACM Press, 514-521.

Heiner, J.M., Hudson, S.E. “The Information Percolator: Ambient Information Display in a Decorative Object”, Proceedings of User Interface and Software Technology 1999 (UIST 99), ACM Press, p. 141-148.

Harrison, B., Fishkin, K., Gujar, A., Mochon, C. and R. Want. “Squeeze Me, Hold Me, Tilt Me! An Exploration of Manipulative User Interfaces.” Proceedings of CHI ’98. (1998), ACM Press, 17-24.

Hindus, D., Mainwaring, S., Leduc, N., Hagstrom, A. and O. Bayley. “Casablanca: Designing Social Communication Devices for the Home.” Proceedings of CHI ’01. (2001), ACM Press, 325-332.

Hindus, D., Ackerman, M., Mainwaring, S. and B. Starr. “Thunderwire: A Field Study of an Audio-Only Media Space.” Proceedings of CSCW ’96. (1996), ACM Press, 238-247.

Hoberman, P. Bar Code Hotel Web Page, http://www.hoberman.com/perry.

HTTP – Hypertext Transfer Protocol Home Page, http://www.w3.org/Protocols/.

Interbind home page, http://www.interbind.com.

Ishii, H. and B. Ullmer. "Tangible Bits: Towards Seamless Interfaces Between People, Bits, and Atoms." Proceedings of CHI '97. (1997), ACM Press, 234-241.

Java Shared Data Toolkit home page, http://java.sun.com/products/java-media/jsdt/index.html.

Jazz home page, http://www.cs.umd.edu/hcil/jazz

Johnson, W., Jellinek, H., Klotz, L., Rao, R. and S. Card. "Bridging the Paper and Electronic Worlds: The Paper User Interface." Proceedings of InterCHI '93. (1993), ACM Press, 507-512.

Kay, Alan. "The Early History of SmallTalk", in Bergin, Jr., T.J., and R.G. Gibson. History of Programming Languages - II, ACM Press, New York NY, and Addison-Wesley Publ. Co., Reading MA 1996, pp. 511-578.

Kraut, R., Mukhopadlhyay, T., Szczypula, J., Kiesler, S. and W. Scherlis. "Communication and Information: Alternative Uses of the Internet in Households." Proceedings of CHI '98. (1998), ACM Press, 368-375.

Kraut, R., Kiesler, S., Mukhopadlhyay, Scherlis, W. and M. Patterson. "On Site: Social Impact of the Internet: What Does It Mean?" CACM 41 (12). (1998), ACM Press, 21-22.

Kurlander, D., Skelly, T. and D. Salesin. "Comic Chat." Proceedings of CHI '96. (1996), 225-236.

Lamming, M. "Intimate Computing and the Memory Prosthesis: A Challenge for Computer Systems Research?", in ECOOP'93 - Object-Oriented Programming, 7th European Conference, (1993), Proceedings. Lecture Notes in Computer Science, Vol. 707, Springer, 1993, ISBN 3-540-57120-5, 1-3.

Lange, B., Jones, M. and J. Meyers. "Insight Lab: An Immersive Team Environment Linking Paper, Displays, and Data." Proceedings of CHI '98. (1998), ACM Press, 550-557.

Langheinrich, M., Mattern, F., Romer, K. and H. Vogt. "First Steps Toward an Event-Based Infrastructure for Smart Things." Ubiquitous Computing Workshop of PACT 2000. October, 2000, Philadelphia, PA.

Ljungstrand, P. and L. Holmquist. "WebStickers: Using Physical Objects as WWW Bookmarks." Proceedings of CHI '99. (1999), ACM Press, 332-333.

Mackay, W. and D. Pagani. "Video Mosaic: Laying Out Time in a Physical Space." Proceedings of Multimedia '94. (1994), ACM Press, 165-172.

Mackay, W.E., Ratzer, A., and Janecek, P. (2000). Video artifacts for design: Bridging the gap between abstraction and detail. Proceedings of ACM DIS 2000, Conference on Designing Interactive Systems. Brooklyn, New York: ACM Press.

Mateas, M., Salvador, T., Scholtz, J. and D. Sorenson. "Engineering Ethnography in the Home." Proceedings of CHI '96. (1996), ACM Press, 283-284.

McClard, A. and P. Somers. “Unleashed: Web Tablet Integration into the Home.” Proceedings of CHI ’00. (2000), ACM Press, 1-8.

Mitsuoka, M., Watanabe, S., Kakuta, J. and S. Okuyama. “Instant Messaging with Mobile Phones to Support Awareness.” Applications and the Internet. (2001), IEEE, 223-230.

Mynatt, E., Back, M., Want, R. and R. Frederick. “Audio Aura: Light-Weight Audio Augmented Reality.” Proceedings of UIST ’97. (1997), ACM Press, 211-212.

Mynatt, E., Rowan, J., Jacobs, A. and S. Craighill. “Digital Family Portraits: Supporting Peace of Mind for Extended Family Members.” Proceedings of CHI ’01. (2001), ACM Press, 333-340.

Mynatt, E., Igarashi, I., Edwards, W. and A. LaMarca. “Flatland: New Dimensions in Office Whiteboards.” Proceedings of CHI ’99. (1999), ACM Press, 346-353.

Nardi, B., Whittaker, S. and E. Bradner. “Interaction and Outeraction: Instant Messaging in Action.” Proceedings of CSCW 2000. (2000), ACM Press, 79-88.

Nelson, L., Ichimura, S., Pederen, E. and L. Adams. “Palette: A Paper Interface for Giving Presentations.” Proceedings of CHI ’99. (1999), ACM Press, 354-361.

Neuwirth, C., Chandhok, R., Charney, D., Wojahn, P. and L. Kim. “Distributed Collaborative Writing: A Comparison of Spoken and Written Modalities for Reviewing and Revising Documents.” Proceedings of CHI ’94. (1994), ACM Press, 51-57.

Oikarinen, J. and D. Reed. “Internet Relay Chat Protocol – RFC 1459.” (1993).

Pederson, E., McCall, K., Moran, T. and F. Halasz. “Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings.” Proceedings of CHI ’93. (1993), ACM Press, 391-398.

Pederson, E. and T. Sokoler. “AROMA: Abstract Representation of Presence Supporting Mutual Awareness.” Proceedings of CHI ’97. (1997), ACM Press, 51-58.

Rekimoto, J. “Pick and Drop: A Direct Manipulation Technique for Multiple Computer Environments.” Proceedings of UIST ’97. (1997), ACM Press, 31-39.

Rekimoto, J. “A Multiple Device Approach for Supporting Whiteboard-based Interactions.” Proceedings of CHI ’98. (1998), ACM Press, 344-351.

Roussel, N. “Exploring new uses of video with videoSpace. ” Proceedings of EHCI’01, 8th IFIP Working Conference on Engineering for Human-Computer Interaction (2001). Lecture Notes in Computer Science, Springer-Verlag, May 2001, pp 73-90.

Scholtz, J., Mateas, M., Salvador, T. and D. Sorenson. “User Requirements Analysis for the Home.” Proceedings of CHI ’96. (1996), ACM Press, 326-327.

SGI-OpenGL Home Page, http://www.sgi.com/software/opengl/.

Sony VAIO Slimtop Home Page, http://www.sonystyle.com/vaio/pentablet/glamour.htm.

Stanton, D., Abnett, C., Bayon, V., Cobb, S. and C. O'Malley. "The effect of tangible interfaces on children's collaborative behaviour." In Augemented Space for Shared Story Telling. (2001), KidStory: Esprit Project 29310, 235-246.

Stifelman, L., Arons, B. and C. Schmandt. "The Audio Notebook: Paper and Pen Interaction with Structured Speech." Proceedings of CHI '01. (2001), ACM Press, 182-189.

Stifelman, L., Arons, B., Schmandt, C. and E. Hulteen. "VoiceNotes: A Speech Interface for a Hand-Held Voice Notetaker." Proceedings of InterCHI '93. (1993), ACM Press, 179-186.

Streitz, N. and D. Russell. "Basics of Integrated Information and Physical Spaces: The State of the Art." Proceedings of CHI '98. (1998), ACM Press, 273.

TurboNote+ Home Page, http://turbonote.com.

Ullmer, B, Ishii, H. and D. Glas. "mediaBlocks: Physical Containers, Transports, and Controls for Online Media." Proceedings of Computer Graphics '98. (1998), ACM Press, 379-386.

Venkatesh, A. "Computers and Other Interactive Technologies for the Home." CACM 39 (12). (1996), ACM Press, 47-54.

Viegas, F. and J. Donath. "Chat Circles." Proceedings of CHI '99. (1999), ACM Press, 9-16.

Wacom PL Series Home Page, http://www.wacom.com/lcdtablets/index.cfm.

Wang Laboratories, Inc. "Freestyle." ACM SIGGRAPH Video Review Supplement to Computer Graphics, 45 (4). ACM Press, 1989.

Want, R. and D. Russell. "Ubiquitous Electronic Tagging." IEEE Distributed Systems Online. September 2000, 1 (2).

Want, R., Fishkin, K., Gujar, J. and B. Harrison. "Bridging Physical and Virtual Worlds with Electronic Tags." Proceedings of CHI '99. (1999), ACM Press, 370-377.

Weiser, M. "The Computer for the 21st Century." Scientific American, 265 (3). (1991), 94-104.

Weiser, M., Brown, J.S. "Designing Calm Technology" Powergrid Journal, 1.01, 1996.

Wellner, P. "Interacting With Paper on the DigitalDesk." CACM, 36 (7). (1993), ACM Press, 87-96.

Wellner, P. Mackay, W. and R. Gold. "Computer Augmented Environments: Back to the Real World." CACM, 36 (7). (1993), ACM Press, 24-26.

Whittaker, S., Swanson, J., Kucan, J. and C. Sidner. "TeleNotes: Managing Lightweight Interactions in the Desktop." ACM Trans. On Computer-Human Interaction 4 (2). (1997), ACM Press, 137-168.

Wixon, D. and K. Holtzblatt. "Contextual Design: An Emergent View of System Design." Proceedings of CHI '90. (1990), ACM Press, 329-336.

Yahoo Messenger Webcam Page, http://messenger.yahoo.com/messenger/web-cams.html

Yarin, P. and H. Ishii. "TouchCounters: Designing Interactive Electronic Labels for Physical Containers." Proceedings of CHI '99. (1999), ACM Press, 362-369.

interLiving

Designing Interactive, Intergenerational
Interfaces for Living Together. IST-2000-26068
http://interliving.kth.se

Collaborating Partners

Kungl Tekniska Högskolan, KTH

Universite Paris-Sud, LRI

Institut National de Recherche en Informatique
et en Automatique, INRIA

ISSN 1650-8009
ISBN 91-7283-261-4

Available on request from
CID/NADA/KTH
SE-100 44 Stockholm
Sweden