Human-Computer Interaction
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t775653648

Integrating Physical and Digital Interactions on Walls for Fluid Design Collaboration

Scott R. Klemmer [a]; Katherine M. Everitt [b]; James A. Landay [b]
[a] Stanford University, [b] University of Washington,

Online Publication Date: 01 April 2008

To cite this Article Klemmer, Scott R., Everitt, Katherine M. and Landay, James A.(2008)'Integrating Physical and Digital Interactions on Walls for Fluid Design Collaboration',Human-Computer Interaction,23:2,138 — 213

To link to this Article: DOI: 10.1080/07370020802016399

URL: http://dx.doi.org/10.1080/07370020802016399

HUMAN–COMPUTER INTERACTION, 2008, Volume 23, pp. 138–213
Copyright © 2008 Taylor & Francis Group, LLC
ISSN: 0737-0024 print / 1532-7051 online
DOI: 10.1080/07370020802016399

Integrating Physical and Digital Interactions on Walls for Fluid Design Collaboration

Scott R. Klemmer[1], **Katherine M. Everitt**[2], and **James A. Landay**[2]
[1]Stanford University
[2]University of Washington

ABSTRACT

Web designers use pens, paper, walls, and tables for explaining, developing, and communicating ideas during the early phases of design. These practices inspired The Designers' Outpost. With Outpost, users collaboratively author Web site information architectures on an electronic whiteboard using physical media (sticky notes and images), structuring and annotating that information with electronic pens. This interaction is enabled by a touch-sensitive electronic whiteboard augmented with a computer vision system. The Designers' Outpost integrates wall-

Scott Klemmer is an Assistant Professor of Computer Science at Stanford University, where he codirects the Human–Computer Interaction Group; his research passion is enabling designers and other innovators to create interactive media that integrates the physical and digital worlds. **Katherine Everitt** is a Ph.D. student at the University of Washington, specializing in human–computer interaction; her research interests include multimodal interfaces and collaborative interaction using large surfaces. **James Landay** is an Associate Professor in Computer Science & Engineering at the University of Washington, specializing in human–computer interaction; his current research includes automated usability evaluation, demonstrational interfaces, ubiquitous computing, user interface design tools, and Web design.

CONTENTS

scale, paper-based design practices with novel electronic tools to better support collaboration during early-phase design. Our studies with professional designers showed this integration to be especially helpful for fluidly transitioning to other design tools, access and exploration of design history, and remote collaboration.

1. INTRODUCTION

> We interact with documents in two separate worlds: the electronic world of the workstation, and the physical world of the desk. Interaction styles in these two worlds do not resemble each other, functions available are different, and interaction between the two is limited.
> — Pierre Wellner (1993, p. 87)

For three decades, pundits have touted the imminent arrival of the paperless office. However, paper remains a central artifact in professional work practices, and its use has risen significantly in the computer age: "In 1975, offices consumed less than 100 pounds of paper per head, now they consume more than 200" (Brown & Duguid, 2002). Paper is tangible, portable, readily manipulable, and easily editable. Newman and Landay's (2000) study of Web design practice found that pens, paper, walls, and tables were often used for explaining, developing, and communicating ideas during the early phases of design. Designers prefer these tools because they are flexible, immersive, and calm. In addition, a wall-scale workspace allows multiple people to simultaneously view, discuss, and modify a design. However, when using paper and walls for design, it is laborious to manually update structuring information (such as links and annotations), create multiple versions, or collaborate with designers at another location. Later phase design, where detailed page mockups are generated, occurs mostly on the computer. This finding is consistent with work practice studies across many design and engineering domains (Bellotti & Rogers, 1997; Hymes & Olson, 1997; Whittaker & Schwarz, 1995).

These wall-scale, paper-based practices inspired The Designers' Outpost, an interface that integrates the affordances of paper and large physical workspaces with the advantages of electronic media for supporting information design. With Outpost, users collaboratively author Web site information architectures on an electronic whiteboard using physical media (Post-it® notes and images), upon which they structure and annotate the information with electronic pens. This interaction is enabled by a touch-sensitive SMART Board augmented with a computer vision system. Thus, paper in the physical world becomes an input device for the electronic world. A rear-mounted projector outputs electronic information onto surfaces in the physical world.

Structured capture has no intrinsic merit; it is through the use of the captured material that value is produced. There are three primary avenues through which Outpost's integrated interactions deliver value to designers. First, through its electronic capture of designs, Outpost supports the transition from early representation to later electronic tools. Second, The Designers' Outpost supports informal history capture and retrieval. Its history interface comprises three novel visualizations for collaborative early-phase design. The most important of which, the main time line, is a visually navigable, time-ordered sequence of design thumbnails. Our focus on early-phase design led us to fluid, informal techniques that capture information users produce in the normal course of their activities and structure this information for later retrieval. Third, to better support remote collaboration, The Designers' Outpost introduces an interaction paradigm where objects that are physical in one space are electronic in the other space, and vice versa. This system also introduces two remote awareness mechanisms: transient ink input for gestures and a shadow of the remote collaborator for presence.

Over the 3 years of the project, we conducted in-lab evaluations of Outpost with 27 users. These studies were part of our iterative design process, where we assessed the usability of the system at each stage of development and gathered information that fueled subsequent development. Aspects of this work have been reported in three papers (Everitt, Klemmer, Lee, & Landay, 2003; Klemmer, Newman, Farrell, Bilezikjian, & Landay, 2001; Klemmer, Thomsen, Phelps-Goodman, Lee, & Landay, 2002). In this article, we expand on these earlier publications by presenting more information about both the initial design explorations and the final implementation. The Introduction (section 1), Related Work (section 5), and Conclusions (section 6) sections are also new, providing an expanded opportunity to discuss the work as a whole. We also present a number of enhancements to the Outpost system, many of which were driven by the experiences of the participants in our studies. Content throughout was updated so that in all cases, the description is accurate as of the end of the project in May 2003, and the writing was clarified based on our revisions and of those suggested by readers and reviewers of drafts. Videos of the Outpost system are available online at http://hci.stanford.edu/research/outpost/video.

1.1. Current Physical Practice: Benefits and Drawbacks

In one common early-phase design practice, designers collect ideas about Web site content onto Post-it notes and then arrange them on the wall into categories. This technique, often called affinity diagramming (Beyer & Holtzblatt, 1997), is a form of collaborative "sketching" used to determine the site structure (see Figure 1). The large workspace of a wall or whiteboard offers several clear benefits for collaborative design tasks. Large workspaces permit the representation of large, complex information spaces without the loss of contextual, peripheral information (see Figure 2). In contrast with the heavyweight, formal operations of the computer, it is relatively easy to fill a wall with pieces of paper and move them around to suggest different associations. Paper and walls "make information, any kind of information, tangible, easy to manipulate, and easy to organize" (Rettig, 2000, p. 4). Collaboration is aided both by the persistence of the design artifact, which supports asynchronous collaboration and constant awareness of the state of the project, as well as by the greater-than-human-sized space, which allows multiple people to simultaneously view, discuss, and modify the artifact. Covi, Olsen, Rocco, Miller, and Allie (1998, p. 59) referred to the work posted on walls in project rooms as "coordination documents" because of the important role these highly visible artifacts play in collaboration.

There are drawbacks, however, to the traditional paper-centric representation. Much of the information exists in the relationships between information

Figure 1. **A designer sitting in front of a Post-it note covered wall. The Post-it notes represent chunks of information and are arranged spatially into groups of related information. These notes are linked with marker lines to show organizational relationships. Image courtesy of Hugh Beyer and Karen Holtzblatt (Beyer & Holtzblatt, 1997).**

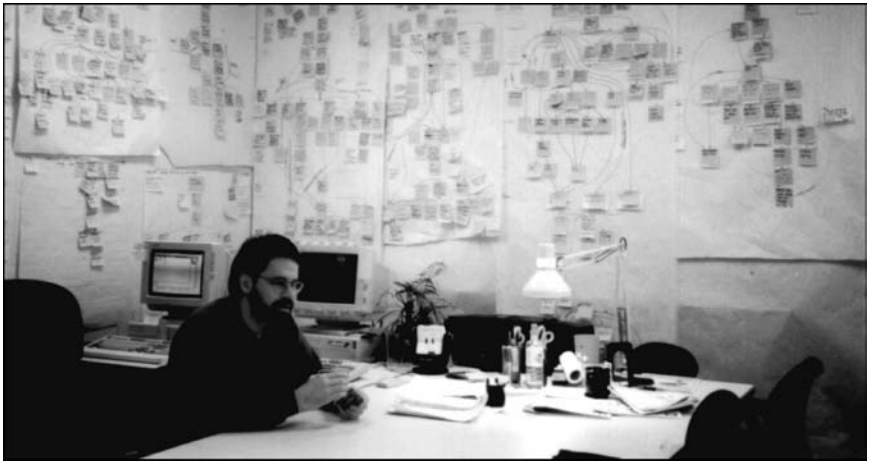
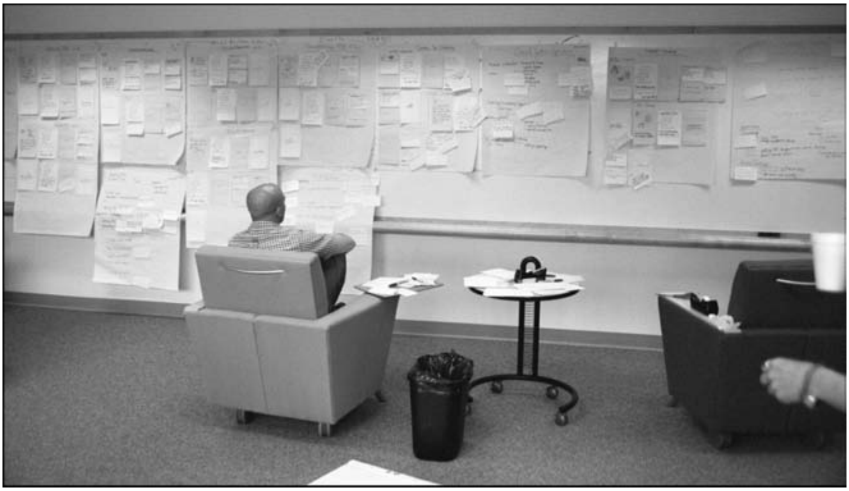


Figure 2. **One of two design rooms at a Silicon Valley Web site design firm.**



chunks (Post-it notes). Because structure must be maintained manually, content elements that are semantically related to other pieces of content, such as links or annotations, often fall out of sync with the notes as they are shifted around. At some point, whether hours after a brainstorming session or months after a project, the paper is removed and the Web site structure lost. The designers in our studies also lamented that versioning is not feasible in a paper-only representation. In addition, the paper-only work practice offers few opportunities for remote participants, whether at a desktop down the hall or in a meeting room across the world. Remote users have no way to update or even access, the information. We also found, as others have, that the transition from the early paper-centric design stages to the later pixel-centric stages is highly problematic (Landay & Myers, 2001; Wellner, 1993). As the site structure changes during development, the early paper artifact becomes increasingly out of date.

With Outpost, users have the same fundamental capabilities as in a noncomputational paper-based system: They can create new pages by writing on new Post-it notes and organize a site by physically moving notes around on the board (see Figure 3). Through its electronic capture of designs, Outpost supports the transition from this early representation to later electronic tools, such as DENIM (Lin, Newman, Hong, & Landay, 2000; Newman, Lin, Hong, & Landay, 2003; see Figure 4).

Outpost was part of our research initiative on informal user interfaces (Landay & Myers, 2001). Informal user interfaces support natural human input, such as speech and writing, while minimizing recognition-based transformation of that input. These interfaces, which document rather than transform, better support a user's flow state (Csikszentmihalyi, 1990) than tradi-

Figure 3. **The Designers' Outpost: A Web site information architecture using a combination of physical Post-it notes, physical pictures, and virtual links showing relationships between them.**

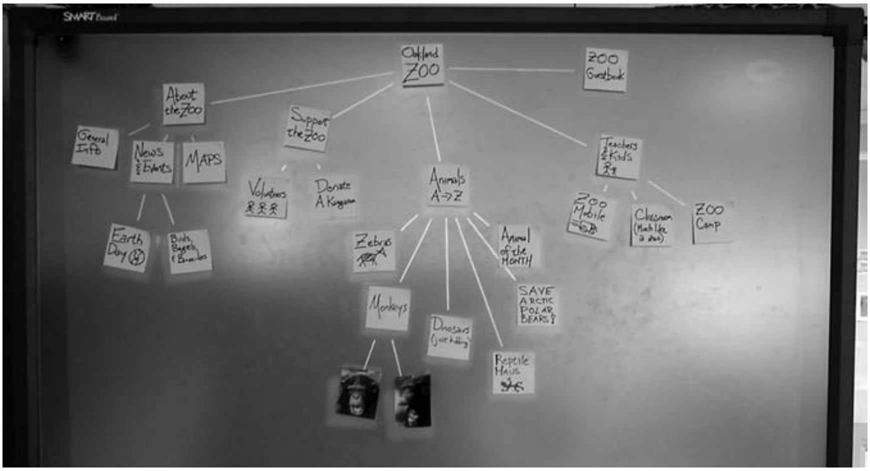
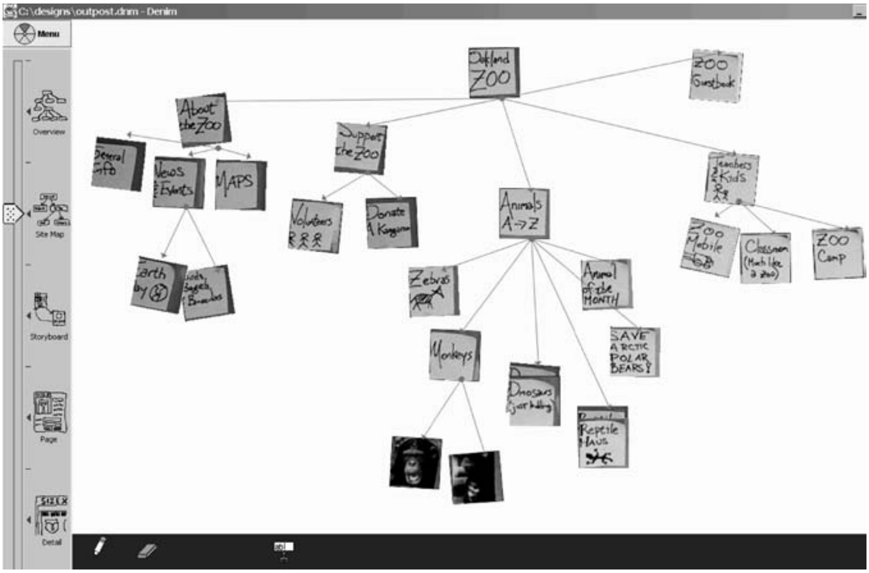


Figure 4. **DENIM, shown here in sitemap view, allows Web site design by sketching. As seen here, physical information spaces created in Outpost can be imported electronically into DENIM and then serve as baseline sitemaps.**



tional recognition-based interfaces. Unrecognized input—such as freehand ink and speech—embraces nuanced expression and suggests a malleability of form that is critical for activities such as early-stage design. In addition to Outpost, the initiative led to the development of informal design tools for graphical (Landay, 1996; Landay & Myers, 1995; Landay & Myers, 2001), Web (Lin et al., 2000; Newman et al., 2003), speech (Klemmer et al., 2000; Sinha, Klemmer, & Landay, 2002), multimodal (Sinha & Landay, 2003), and cross-device (Lin, 2003; Lin & Landay, 2002) user interfaces.

1.2. Background

Our research is inspired by fieldwork into Web design practice and the writings of designers reflecting on and prescribing effective design methods. We describe each of these in turn.

Web Design: Tools and Practices

The goal of Newman and Landay's investigation into Web design (Newman & Landay, 2000; Newman et al., 2003) was to enable the development of systems that better support actual practice. The study comprised interviews with 11 professional Web site designers from five different companies. Each interview consisted of asking the designer to choose a recent project and walk the interviewer through the entire project. The designer was asked to show examples of artifacts produced during each phase and explain his or her role in the process.

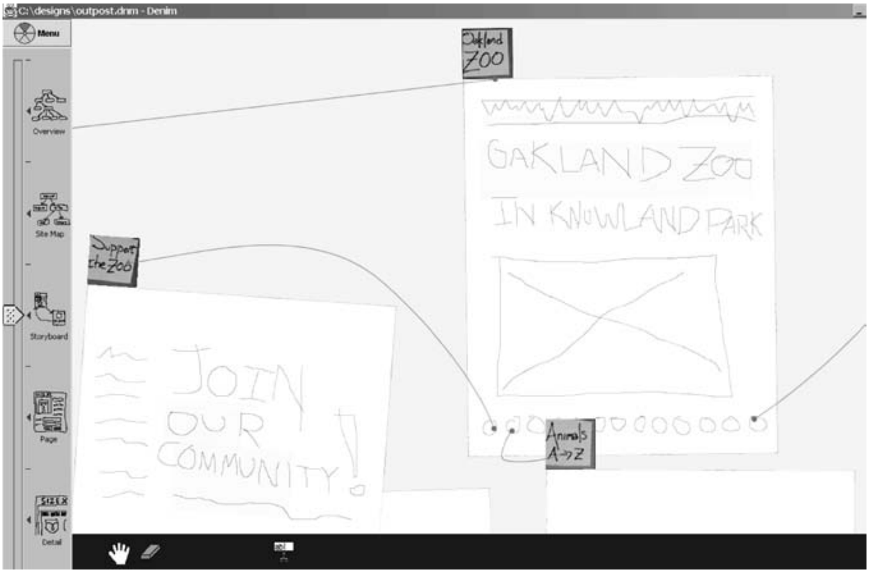
Three important observations were made during the course of this study. First, designers create many different intermediate representations of a Web site. Examples of pervasive and significant intermediate artifacts include sitemaps, storyboards, page schematics, and mockups. These representations depict the Web site at varying levels of detail, from sitemaps, which depict sites as related blocks of labeled information, to mockups, which depict individual pages in high fidelity. Second, the production and use of these intermediate artifacts dominate the day-to-day work practice for most of the design process. Third, Web design is comprised of several subspecialties, including information architecture and visual design, each of which has its own tools, products, and concerns. Whereas visual designers typically focus on interaction and graphic design, information architects are mainly concerned with the information and navigation design of a Web site. Newman and Landay (2000) found that information architecture is not well supported by current software tools; for example, sitemaps were regularly generated by placing Post-it notes on walls.

The results of these studies provided motivation for the creation of Outpost and provided the impetus for the development of DENIM, a sketch-based tool for information and navigation design of Web sites (Lin et al., 2000; Newman et al., 2003). DENIM supports sketching input, allows design at different refinement levels, and unifies the levels through zooming. In particular, DENIM supports visualizations that match the sitemap, storyboard, and page schematic representations of a web site (see Figure 5). DENIM also allows designers to interact with their site designs through a run mode, which displays the sketched pages in a limited functionality browser that allows the user to navigate the site by clicking active regions of the sketches and linking to other pages within the site. Although DENIM supports authoring sitemaps, it is best suited for storyboards and page schematics. In contrast, Outpost is ideal for sitemaps. In addition, DENIM was designed as a single-user interface, whereas Outpost was designed for collaborative work.

Affinity Diagrams

Our interest in researching computational support for information architecture was motivated by one specific design practice observed during the study just discussed. This collaborative practice—often called affinity de-

Figure 5. **DENIM's storyboard view displays images from the Post-It note information architecture created in Outpost (the two systems communicate using a common XML file format). The sitemap images from Outpost correspond to page titles in DENIM, shown at the top of each page. Designers can continue working with an Outpost sitemap in DENIM for sketching out the contents of a page, shown below the page titles.**

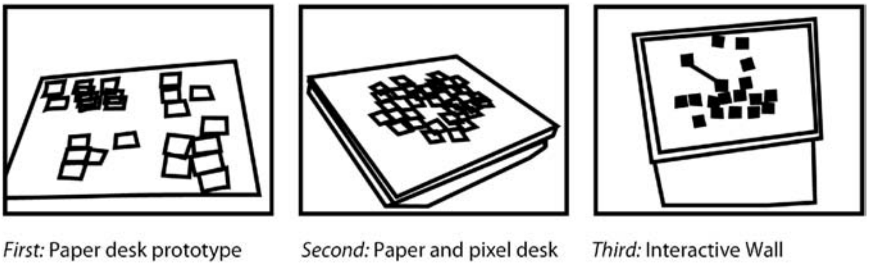


sign—consists of arranging Post-it notes on a large surface such as a wall, table, or desk in order to explore the information structure of a Web site. Designers write chunks of information on Post-it notes and stick them to the wall. They then move the notes into spatially proximate groups representing categories of related information. Groups are labeled and further grouped into hierarchies of groups. This hierarchical structure serves as a baseline for the structure of the Web site. Lines are drawn between notes and groups to indicate links. Early on, these links indicate organizational relationships (i.e., relationships that warrant navigation between the endpoints.) Later in the design process, these links represent hyperlinks that will appear in the finished site. Usability expert Jakob Nielsen also advocates a version of this method, using index cards to design the information hierarchy (see Nielsen, 1998).

2. INITIAL DESIGN STUDIES

We began the Outpost research with a series of low- and high-fidelity prototypes exploring the integration of physical and digital interactions (see Figure 6).

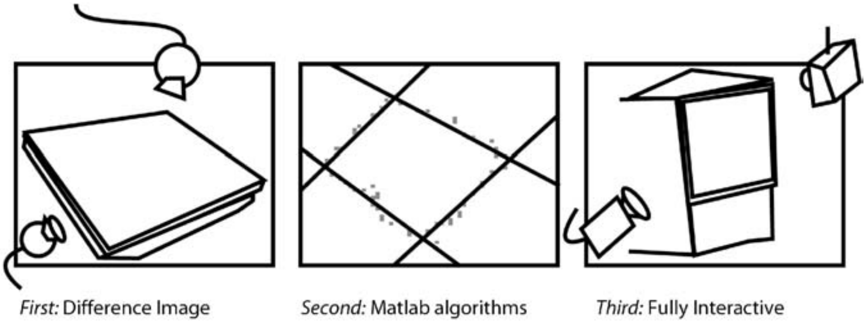
Figure 6. **The sequence of prototypes used in the three design studies.**


First: Paper desk prototype
Second: Paper and pixel desk
Third: Interactive Wall

In the first four subsections, we discuss the Outpost design studies in detail. We first evaluated the basic concept with a paper prototype study. Next, we built interface mockups that envisioned the combination of physical artifact state with interactive feedback. Finally, we created a wall-scale prototype for a set of participatory design sessions with 15 professional interface designers. Designers participated in evaluating the prototypes; they encouraged us to support freeform ink, digital annotations to sitemap pages, versioning of design artifacts, fluid transitions to other tools, and opportunities for collocated and remote collaboration.

After the initial paper prototype, it became clear that computer vision would be the optimal enabling technology for physical input. Thus, in parallel, we built a set of prototypes for the underlying vision system (see Figure 7). We describe these in section 2.5. These prototypes led us to difference im-

Figure 7. **The sequence of computer vision studies. The first prototype explored the difference image algorithm using the Java Media Framework and Webcams. The second prototype explored the expectation-maximization algorithm for line-fitting using Matlab. The final prototype integrated these techniques into a functioning system with a user interface.**


First: Difference Image
Second: Matlab algorithms
Third: Fully Interactive

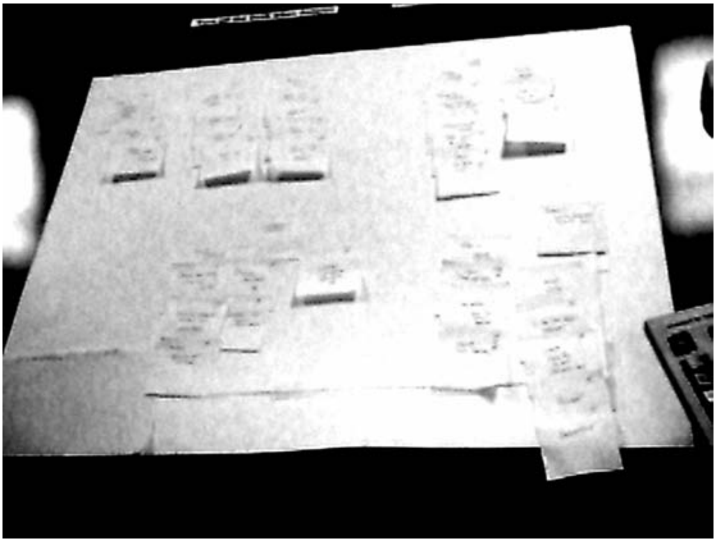
age-based recognition algorithms and a two-camera infrastructure: a rear-mounted video camera for capturing movement and a front-mounted high-resolution camera for capturing ink.

2.1. Low-Fidelity Desk: Design Study

We created our initial low-fidelity prototype using cardboard the size of an ITI VisionMaker Digital Desk (41″ diagonal) and evaluated this paper prototype with two individual participants. The participants wrote on a pad of 3 × 3″ yellow Post-it notes using a standard inking pen (see Figure 8). We asked participants to create the information architecture for a Web site about off-campus housing for college students. To start, we handed them six pages of notes from mock interviews with college students seeking housing. The task included chunking interview information onto Post-its, arranging the Post-its into related groups, and merging two previously saved versions of Post-its into a unified version. A wizard acting as the computer (Kelley, 1984; Maulsby, Greenberg, & Mander, 1993) gave verbal feedback about what the computer recognized as groups and which groupings were being selected and displayed widgets and dialog boxes when appropriate.

Participants often forgot to perform input steps when they were incidental to the immediate goal and the behavior they invoked happened asynchronously. The two actions that lacked an affordance or feedback were (a)

Figure 8. **The low-fidelity Designers' Outpost.**



underlining a note to designate it as the label for a group and (b) pressing an upload button after placing a note on the desk to add it to the system. This shortcoming suggested an interface where user manipulations of physical and graphical elements invoke actions clearly implied by their representation and where these actions are confirmed through visual feedback (e.g., an interface in which a new note is automatically added when it is placed on the desk, and a penumbra is projected behind it to confirm). Users were also confused by the need for three distinct input devices: the inking pen for writing on notes, the virtual stylus for authoring note relationships, and the keyboard for entering version names. As a solution, we removed the keyboard from our system design, which simultaneously simplified the input model and better matched current practice.

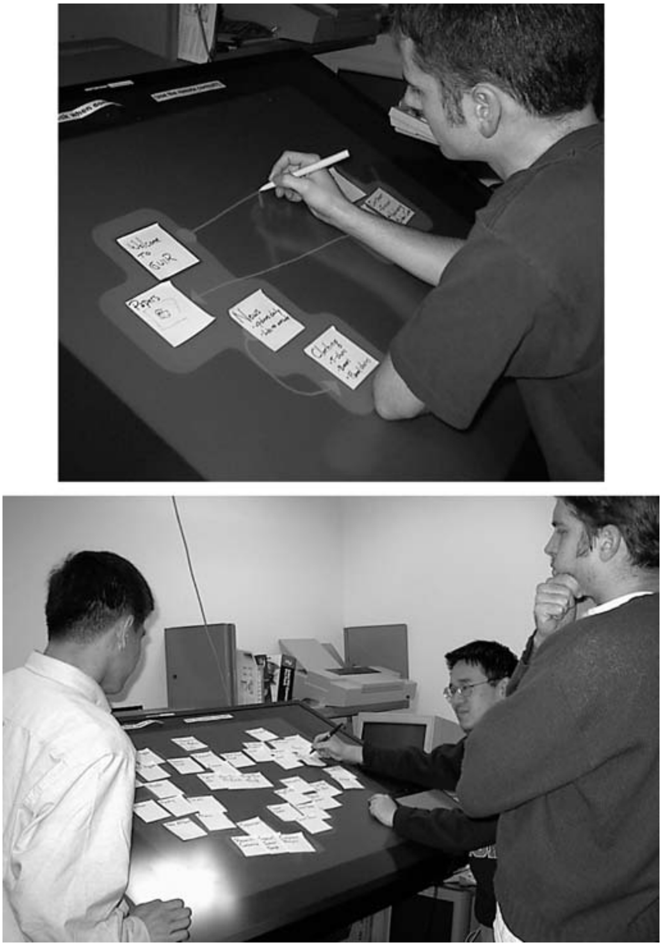
2.2. Pixel and Paper Mockup

Using our findings from the paper prototype, we created a mockup of our ideas for integrated physical–digital interaction. We created static images using Adobe® Photoshop® to prototype Outpost's interaction techniques and visual presentation. These images were displayed on an ITI VisionMaker Digital Desk, which is a rear-projected surface with the size and slope of an architect's drafting table. We designed the initial set of four interaction techniques to aid the affinity diagramming process: (a) Designers could create groups by placing notes near each other, (b) links could be drawn between groups using an electronic stylus, (c) groups could be given a name with a label, and (d) groups could be organized into hierarchies. The interface mockup showed physical Post-it notes and the corresponding electronic feedback for these initial interactions (see Figure 9). Through our experience with this mockup, it quickly became evident that a digital desk is too small a space for professional Web site information architecture. The desk's surface area allows for a maximum of 50 Post-its and two or three users. Information architects often use upward of 200 Post-its, and four to eight people may participate simultaneously in design sessions. To build The Designers' Outpost at a full collaborative scale, we moved our design to a SMART Board, a much larger rear-projected surface with a whiteboard form factor.

2.3. Outpost Interaction Techniques

The low-fidelity and mockup prototypes informed the design of the interactive prototype of Outpost. The first design study validated the general approach of integrated physical and digital interaction. It indicated a need to minimize the extra user effort required to use the tool and encouraged us to allow the interaction to be as freeform as possible. The second prototype

Figure 9. **Mockups of The Designers' Outpost—collaborating on an information hierarchy with Post-its on a digital desk.**



fleshed out the interaction techniques and showed that a drafting desk is too small to support professional-scale Web site diagrams. The designers working with an interactive wall-based prototype found constant interactive feedback distracting. They encouraged us to refocus our interface on supporting digital annotations to sitemap pages, fluid transitions to tools such as DENIM (Lin et al., 2000; Newman et al., 2003), versioning of design artifacts, and supporting collocated and remote collaboration. We also found this system to be more appropriate for information architects than for visual interface designers.

In this interactive prototype, physical, direct manipulation interaction techniques provide for authoring content with standard pens on Post-it notes. The system tracks notes as users physically add, remove, and move them around the board, but it does not attempt to recognize the content of the notes.

Outpost supports the following interaction techniques for working with paper on the board. We have combined these physical interactions with interactions that are better suited to an electronic medium, such as digital ink annotation and the use of virtual arrows to specify relationships.

Adding Notes. Users can write on a note with a standard pen and add it to the board. Outpost's vision system recognizes the note and updates its understanding of the board. The system provides feedback that the note has been recognized by displaying a blue outline around it.

Creating Links. To link two notes, the user draws a line between them with the board stylus.

Removing Notes. To delete a note and its associated links, the user pulls the note off the board.

Moving a Note. To move a note, its links, and its annotations, the user picks it up and places it at a new location. This technique provides a lightweight means of coupling physical and electronic information. A simple timeout heuristic determines movement: If note removal and replacement occur within a specified time (Outpost uses 7 sec), the system interprets this as a move operation. This heuristic was sufficient for evaluating the prototype. A production implementation of Outpost should compare the image of the placed note to that of previously removed notes. This image comparison is tractable with current computer vision techniques (Kim, Seitz, & Agrawala, 2004). Shape Contexts offer the most promise for notes with ink content (Belongie, Malik, & Puzicha, 2002); pyramid-based techniques may be superior for general images (Forsyth & Ponce, 2003, §7.7).

Context Menus. Tapping a note invokes an electronic context menu, enabling the manipulation of the electronic properties embodied by physical objects. The Sticky command replaces a physical note with an electronic image of the note. Delete removes a note (which is useful if the vision system misses a physical removal).

With the feedback from our participatory design study, we added the following three physical tools for manipulating electronic content.

Freeform Ink. In addition to being a space for interacting with physical Post-it notes, Outpost supports freeform drawing using board styli.

Move Tool. A physical move tool provides a means of interacting with the system after the physical content has become electronic, retaining haptic direct manipulation.

Physical Eraser. Like a normal whiteboard eraser, the Outpost eraser removes ink on the board. It operates semantically, deleting each stroke it passes over.

Two primary benefits to structured, electronic capture of informal artifacts are later recall and export to other tools. Saving enables both of these.

Saving the Board. Users can press Save to save the board state to disk. They can then open it later in DENIM or Outpost. (With the introduction of the design history system described in section 3, the history's Bookmark replaced the save button.)

One important part of the Outpost visual design is that the board's background is black. Because the board emits light only where the user has authored content, it is not a giant glowing presence, thus affording the user a calmer interaction experience (Weiser & Brown, 1997).

2.4. Professional Design Study

Fifteen professional Web designers participated in a study of the basic Outpost functionality. The prototype for this design study was a Java application running on a rear-projected 72″ diagonal touch-sensitive SMART Board with a 1280 × 1024 resolution projector. With this prototype, we recognized the location of notes on the board using the board's touch sensor. Drawing a line from one note to another with the board stylus creates a link. The stylus is also used for creating freehand electronic ink on the board (see Figure 10). The context menu in the study prototype (see Figure 11) let users either delete the note or define it as the label note for its group. In the vision-backed Outpost system described later, removing a note from the board deletes it. (The Delete command was retained for the rare instance when the system fails to detect a remove event.)

Study Design

There were five design sessions with between two and five designers per session. In four of the design sessions, the designers were colleagues at the same company; the fifth session mixed designers from two companies. Two of

Figure 10. **The board's tool tray: Styli for drawing electronic ink, a clear plastic square for moving electronic content, and the eraser. (Only the pens were available during the design study.)**

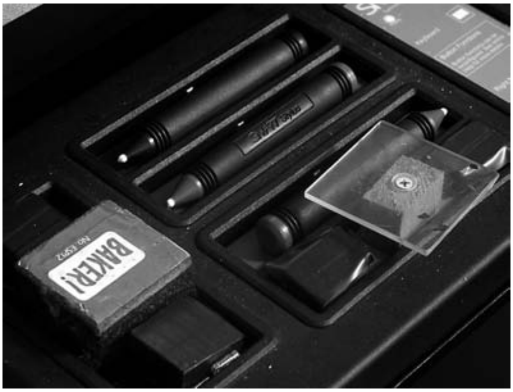
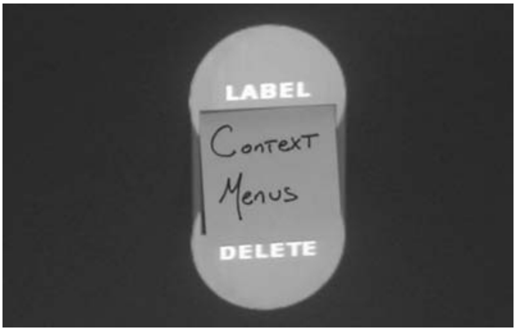


Figure 11. **Tapping on a note invokes an electronic context menu for physical content.**



the five groups were composed of information architects, two groups were visual designers, and one group had individuals performing both roles (see Figure 12).

Two researchers conducted each session. One was in charge of communication, explaining the system, and facilitating discussions. The other took written notes and videotaped the session.

Each session lasted roughly 2 hr (see Figure 13). We began the sessions with a high-level overview of the project and a brief demonstration of the existing prototype. We gave each team an information architecture design task to explore using the prototype; this task took 45 to 60 min. We conversed freely with the designers during the sessions. Throughout each ses-

Figure 12. **The Five Study Groups: Their Size and Primary Role.**

Role	Information Architect		Both	Visual Designers	
Group	A	B	C	D	E
Size	5	3	2	2	3

Figure 13. **The Time Breakdown of the Design Sessions.**

Overview and demo	15 min
Design tasks	45–60 min
DENIM demo	15 min
Discussion	30–45 min
Survey	10 min

sion, participants explained their actions and gave feedback and suggestions for improvement. This discussion was followed by a 15-min demonstration of DENIM and then a 45-min discussion of Outpost's utility and its relationship with DENIM and their current work practices. The study concluded with a 17-question written survey asking participants about their background and their opinions about Outpost and its relevance to their work (Klemmer, 2004, Appendix A).

Existing Practices

Our findings from this study offered insight into the designers' collaborative work processes and suggested an appropriate interactivity model. Every participant reported that they worked with groups on whiteboards early in the Web site design process. The information architects all said that they created sitemaps by placing Post-it notes on the board, whereas the visual designers reported sketching page designs directly on the board. Capturing whiteboard designs was highly valued by all five teams. Three of the design teams used a digital camera for documenting their work, one used a whiteboard capture device (the Virtual Ink Mimio), and one assigned a scribe to save information from design meetings. One of the groups using a camera also used an application called Whiteboard Photo (http://www.pixid.com) to rectify and filter out smudges, dirt, and lighting changes in whiteboard photographs. In addition, all designers said that they used either the Visio or Inspiration structured drawing software for creating sitemaps. Sitemaps can get quite large; designers from one firm said that 200 to 300 nodes is typical.

Structure of Activities During Study

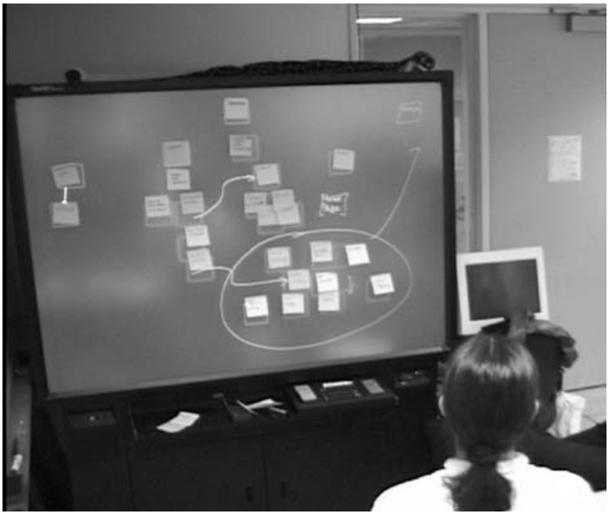
We observed the groups going through three general phases of design when using the interactive prototype. The designers stated that these same phases were part of their existing practice. (Two of the groups did not start the third phase during our lab sessions but said that it would be their next activity.)

Phase 1, Brainstorming. First, the designers brainstormed, quickly putting a large number of concepts on the board. One designer said, "Get all these things on Post-its." The notes simply represent ideas. Sometimes, similar information was placed close together. Designers did not eliminate ideas or link concepts together into any formal structure at this stage. One designer commented that Outpost would be "good for times with the client" because after a meeting the designer could continue to pare down and hone the artifact without having to start from scratch with a new tool. The designers were adamant about not wanting any system feedback during this phase. "We didn't do anything here that we couldn't do on a normal whiteboard." One team actually turned off the board.

Phase 2, Creating. In this phase, designers migrate from a loose federation of notes on the board to a high-level information architecture by clustering related information into groups, pruning unnecessary concepts, and linking notes together. The tool support in the interactive prototype was well suited to this phase. This compatibility was evident in the designers fluid work style and their enthusiastic comments while designing. It was also echoed on the posttest questionnaire, where several designers expressed interest in using Outpost to create top-level information architectures.

Phase 3, Drilling Down. After the designers created a rough cut of a sitemap, we saw work process differences begin to emerge. The visual designers began to work out basic page designs using empty board space and the board stylus. In contrast, the information architects fully fleshed out the page structure of the site, continuing to add notes. A key design implication taken from this phase is the need for associating freeform ink with individual notes (see Figure 14). The visual designers wanted to sketch the design details, and the information architects wanted to add annotations or properties. For example, one information architect said, "I'd like to be able to attach design rationale." Design rationale is a mechanism for asynchronous communication, in which the motivations for making decisions are embedded in the artifact (MacLean, Young, & Moran, 1989). The information architects also had a strong desire to use properties for project management. Two groups suggested

Figure 14. **A design team suggested that freehand ink would be useful for both unstructured annotation of the artifact and for performing operations on groups of notes.**



tagging objects with properties, such as an issue (e.g., “will it be possible to get copyright clearance?”), and later searching for issues across the design. Based on this feedback, we implemented annotation support as part of the design history interface.

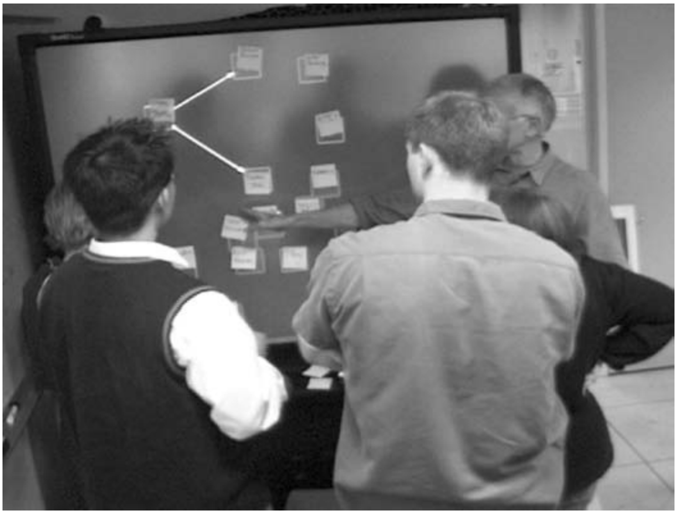
As reflects their discipline, the visual designers often talked explicitly about what pages might look like, whereas the information architecture groups actively discussed users and tasks at a more abstract level: “What does the user know here? What is the user trying to do?”

For all the teams, the site representation operated as the central shared artifact for discussion. Participants were actively working at the board only about half the time. In addition, for short periods (1 min or so), individuals or subgroups broke off from the main discussion to work. During these times, the board remained the anchoring reference point.

We observed two styles of interacting with the board. In the facilitator style, one person, usually the most senior individual, stands at the board (see Figure 15). The entire group discusses the site, but as the discussion progresses, the facilitator creates notes that synthesize the discussion content. One group also referred to this style as “gate keeping.” This style was the primary work practice in three groups, and the groups affirmed it to be their normal work practice.

The second style was open board. As with facilitator, all group members actively discussed the site. In open board, however, there is no central per-

Figure 15. **In the facilitator style, one person remains at the board guiding the group's process.**

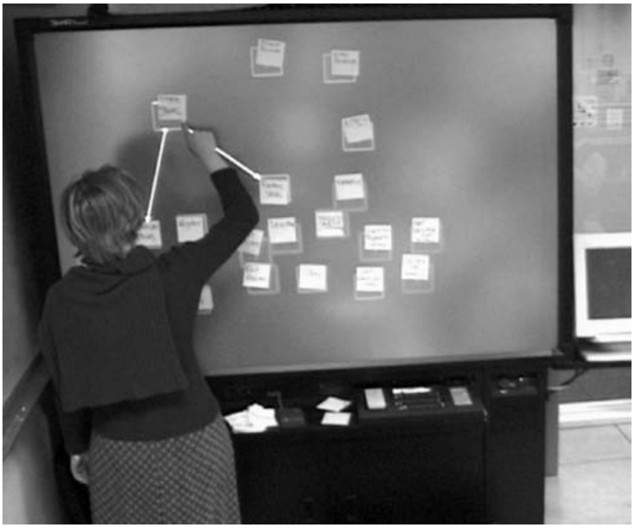


son; all participants have agency to create notes and directly express their ideas in the artifact (see Figure 16). We had started the sessions with a single pad of notes and a single marker next to the board. When the first design team requested one pad and marker per person, however, we provided them for that and all subsequent groups. In this paradigm each person has their own paper "input device"—a working style we had not considered but that the designers regularly employed. In adding content to the board, information moves from a personal creation space to a shared viewing space.

Several participants commented that they valued simultaneous input with a low-latency response. The SMART Board's touch sensor only supports one action at a time. (Newer SMART DViT Boards that use orthogonally mounted cameras—instead of a transparent resistive film—support multiple simultaneous touch points.) Concurrent use of the board has technical design implications for the note-sensing technology. This result encouraged us to complete a computer vision system. Vision lends itself both to simultaneous input and to rich sensing capabilities (e.g., object size, color, orientation, and capture of its contents).

The posttest questionnaire asked, "How likely is it that you would integrate Outpost as a regular part of your Web site design practice?" Participants rated their response on a 5-point Likert scale. Four participants rated the system the top value (*very likely*). Eight gave the second value (*somewhat*

***Figure 16.* In the open board style, all participants directly express their ideas in the artifact.**



likely). Three gave the fourth value (*somewhat unlikely*). One must be cautious about drawing strong conclusions from a participant's positive ratings: The positive rating may only reflect politeness toward the researchers. With this in mind, the most valuable information from these results is that 3 of the participants reported that they would be somewhat unlikely to regularly use the Outpost prototype. We believe the primary reason for these participants' negative feelings was the distracting visual feedback in this prototype; the current system is much calmer. In addition, in our research since the study, we introduced three substantial areas of functionality (transitions to other tools, support for design history, and remote collaboration) that provide important benefits unavailable with a whiteboard or other tools.

Enthusiasm for the prototype correlated directly with two variables: the percentage of the designer's work that was Web based, and how much the designer saw their role as an information architect rather than a visual designer. As one visual designer said, "We don't really do sitemaps so much. Our interfaces tend to end up with one or two screens." Information architects saw sitemap creation as the challenging process of designing the core structure of a Web site. The information architects praised our faithfulness to their current wall-scale work practices and were enthusiastic about the integrated physical-digital interaction.

2.5. Design Implications

This study underscored several important points about how calm (Weiser & Brown, 1997) an informal design tool must be; the system feedback should not interrupt the designer's flow state.

Smart Yet Silent

We originally felt that one benefit of the prototype was the system's ability to automatically recognize groups based on note proximity and provide visual feedback. However, the designers unanimously felt that automatic grouping was not useful, as they already knew the layout of the notes.

Furthermore, the group, note outline, and menu feedback was considered distracting during brainstorming. One designer said, "I'm totally disturbed while I'm trying to concentrate on what we are doing. There are too many things flashing." In hindsight, this result is consistent with the negative user opinion about automatic interpretation and immediate feedback in SILK (Landay, 1996; Landay & Myers, 2001), a sketch-based graphical user interface design tool. In redesigning the system, we removed the visual feedback for proximity-based groups because the visual structure of the design is readily apparent—at best it can be redundant, and at worst it can be wrong. In addition, we replaced the bright, crisp rectangles shown in Figures 14 through 16 with the dim, penumbral shadow as seen in Figure 3. We designed this shadow to be at the just-noticeable threshold of perceptual saliency.

Several participants valued the subtle visual relationships between notes: "Automatically arranging them would take away from my thinking." One designer said that she wanted "to work with this before it's turned on." A difficulty with automatic refinement of informal user input is that the refinement is distracting for users engaged in a creative brainstorming task. This correlation implies that only explicit user actions should cause visible system actions. In general, designers felt that interactive feedback and transformation should not be forced on them: It should be available, but not automatic. This suggests a usage model where, as designers move from brainstorming into more explicitly creating a sitemap, their use of the interactive features increases.

Integrating Physical and Digital Interactions

There are appealing aspects to both digital wall-scale interfaces (Moran, Chiu, van Melle, & Kurtenbach, 1995) and physical ones (Moran et al., 1999). The Outpost project aims to leverage the advantages of both interaction paradigms.

Our series of design studies provided insight into a sweet spot on the physical-digital spectrum. Working physically supports collocated collaborative processes. Direct manipulation of physical notes makes them easier to see, move, and share.

We reviewed the study videotapes to quantify the pace of interaction. We found that on average, a note was added to the board approximately every 25 sec. During active periods, a note was added every 3 to 5 sec. Often, there was no explicit interaction for minutes at a time. A good portion of the meetings happened off the board but referenced the board.

One facilitator began by authoring the sitemap virtually, sketching out square notes and their content. Working purely in the electronic domain has the advantage that there is no need to switch between an inking pen and a noninking stylus. However, working electronically was noticeably slower (top speed of one note every 7 to 10 sec) because (a) the designer had to create page boundaries rather than using the predefined pages torn from a pad, (b) authoring with plastic pens on a plastic surface is awkward for textual input, and (c) the projector ink feedback is much lower resolution than paper. These difficulties hindered the artifact creation process, discouraging descriptive input. For example, in one instance, one participant wrote "B" instead of "Business" when using electronic ink. Later, he started working physically, and the working pace and artifact quality picked up substantially.

In our designs, we were careful to preserve many of the successful aspects of working on a traditional whiteboard; the utility of these features became apparent in the study. Our system permits the representation of large, complex information spaces without the loss of contextual, peripheral information. One designer referred to our interface as "cross-cultural" because engineers, designers, and clients are all comfortable working informally on whiteboards.

Information appliances should be as easy to learn as physical appliances (Norman, 1998). When 2 participants showed up half an hour late, we were pleasantly surprised to see that the timely participant quickly brought her colleagues up to speed with Outpost's interaction techniques. After using the tool for only 5 min, she was easily able to communicate the conceptual model and the functionality of the prototype.

Extending the Existing Design Process

Every group mentioned that migrating the design artifact to other tools for further refinement is an essential advantage of the Outpost system. Many of the designers currently photograph meeting whiteboards even though this only produces a static artifact. They were very interested in the prospect of returning to their desk with an interactive site representation that they could continue to work on.

DENIM is an excellent complement to Outpost: It offers the ability to edit the information architecture, specify page level details, and create the navigational structures for a Web site. Its pen-based interface is intended for a single designer working at a PC. Outpost is most appropriate for creating sitemaps, whereas DENIM becomes more relevant when the design team starts to storyboard the specific pages and create schematics. The current Outpost system and DENIM read and write the same XML file format. This enables an individual to "save out a wall" from a collaborative design session and then flesh out the design on a personal computer or tablet. To support this, we augmented DENIM to handle images as page labels (see Figures 4 and 5).

Long projects magnify the benefits of having a sitemap artifact remain in use throughout the entire design cycle. For example, one design team we spoke with was in the midst of a redesign for a large Web site they had originally designed almost a year earlier. Through its electronic capture functionality, we hope Outpost will help design teams with such long-term projects.

Although an early objective for Outpost was to provide interactive support for information architecture design sessions, designers in the third design study found additional fruitful directions for our research. They encouraged us to refocus our efforts toward a more documentary interface, supporting digital annotations to sitemap pages, versioning of design artifacts, fluid transitions to tools such as DENIM, and supporting collocated and remote collaboration.

2.6. Computer Vision Prototypes

To illuminate the technology issues involved in building wall-scale tangible interfaces, we now briefly present the three computer vision prototypes we built. Outpost employs computer vision to precisely locate and capture Post-it notes and images that users place on the board. Computer vision is appropriate for this task as it provides untethered and untagged tracking and capture of artifacts from multiple users simultaneously. The first prototype was a simple system that computed the difference image between frames and analyzed this difference image to detect changes in the board state. In the second prototype, we used Matlab to prototype the full set of algorithms necessary to support the Outpost application. The third prototype was built on top of Intel's OpenCV library (Bradski, 2001) and implemented the computer vision algorithms at interactive rate, as well as a socket-based network connection for communicating with the Outpost user interface (UI).

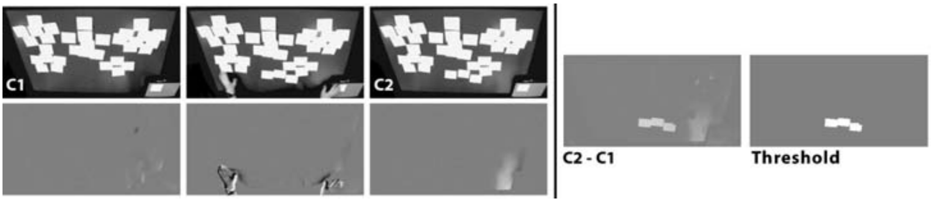
Difference Image Vision Prototype

The foundation of Outpost's recognition system is the difference image technique, computed by subtracting one camera frame from another (see Figure 17). The difference image expresses the change in board state between the two points in time.

We originally employed difference images for two purposes: (a) to ascertain when users are working on the board and when it is calm and (b) as an object detection primitive. A difference image is computed by subtracting consecutive frames (see the bottom row of Figure 17). The content of a difference image expresses the activity of the board. An activity metric a can be computed by summing the absolute difference values of all pixels in the image. Outpost considers the board to be active if this metric exceeds a specified threshold. Otherwise, the board is considered calm. In reality, the content of a single frame difference image also results from noise in the camera sensor array and from lighting changes in the world (e.g., someone walks between a light source and the board). The threshold used for this prototype was an absolute value change of 2.25 units per pixel, on a scale of 0 to 255. The goal of ascertaining activity was to find a calm frame that could be compared with the current frame to find changes in the board state. This activity metric is a fairly crude method that usually, but not always, filtered out both sensor noise and lighting changes. The method is crude because the decision of whether an image is calm, and thereby useful for a baseline, is binary. In the wall vision prototype, we replaced the activity method with spatial filtering and temporal averaging techniques that provide a more robust baseline image for comparison. Filtering and averaging obviates the need to make a binary decision about fitness, as the new baseline image is a moving aggregate of a number of frames.

This prototype also used difference images as an object-detection primitive. Outpost does not require that notes be tracked while they are moving, but it does require that the system is aware of a note when it is initially placed down, placed at a new location, or removed. When the board becomes active,

Figure 17. **Excerpts from an image sequence from our prototype steady state algorithm. Raw camera frames are shown in the top row, single frame difference images are shown in the bottom row. Raw and thresholded c2 − c1 difference images are shown at right.**



the system saves the last calm frame c1. When the board becomes calm again, it captures the new calm frame c2. Subtracting c1 from c2 and thresholding the result tells the system what has changed during that period of activity (see Figure 17). This thresholded difference image becomes the input for note recognition.

In building this first prototype, we realized that locating a note is a separate task from capturing a note. Dividing the vision task into these two distinct parts enabled us to realize that the system should use two cameras. To obtain an occlusion free view of the board for our difference image algorithm, we followed the metaDESK researchers (Ullmer & Ishii, 1997), mounting a video camera behind the board. Interactive frame rates are crucial for this camera. Because the notes are fairly large (3″ square), standard video resolution (640 × 480) is sufficient for location and orientation detection.

Ink capture has the opposite set of constraints: It requires high resolution for capture but not interactive speeds because the ink capture does not control the board feedback. These constraints suggest a high-resolution still camera. The two-camera approach eliminates the need for a mechanical pan/tilt/zoom camera and image stitching algorithms.

When we began this project in 1999, we found that consumer-grade Web cameras compressed images in ways that make computer vision difficult (Ulrich & Nourbakhsh, 2000). The situation has since improved substantially; today, consumer-grade cameras are generally sufficient for the types of vision that Outpost uses.

Matlab Algorithms Prototype

Using the difference image just described as a building block, we designed and prototyped the complete vision pipeline in Matlab before implementing it in an interactive system. This prototype introduced perspective correction, segmentation, and feature extraction. We used Matlab because the time required for software development using libraries such as OpenCV (Bradski, 2001) would have prohibited us from experimenting with design alternatives. The need for rapidly exploring multiple alternatives helped inspire the Papier-Mâché research (Klemmer, Li, Lin, & Landay, 2004). The pipeline performs the following set of operations on D(c2 - c1):

1. Rectify the perspective camera view of the board plane, bringing the board into a 2D plane using a 3 × 3 homography map matrix (Forsyth & Ponce, 2003, §13.1): A homography matrix describes an arbitrary projective transformation. Although more precise algorithms for camera calibration exist, we chose a homography because it is very fast.

2. Threshold the resulting image, producing a three-level image: Positive pixels are pixels that have gotten significantly brighter, neutral pixels have not changed much, and negative pixels have become significantly darker.
3. Segment the image using the connected components algorithm, labeling each positive and negative pixel with a cluster ID number: The system interprets note-sized clusters of positive pixels to be added notes and note-sized clusters of negative pixels to be subtracted notes.
4. Compute the center of mass and the orientation of the note: Inspired by Freeman's work (Freeman et al., 1998), we originally had the system determine orientation using a second moment algorithm. However, the second moment is undefined for squares, circles, and other objects that are symmetrical about both the *x*- and *y*-axes: for these shapes, all orientation choices yield an identical second moment. For this reason, we moved to an expectation-maximization (EM) algorithm (Forsyth & Ponce, 2003, §16.1) that finds the best-fitting square on the set of outline pixels of the note. This method is highly robust, even for a highly degraded image outline and small sample size. Theoretically, EM is a more expensive algorithm because it is iterative. In practice, we found that a small number of iterations (in our case, six) is sufficient for the solution to converge.

Interactive Wall Vision Prototype

We combined the system implications from our first vision prototype with the algorithms from our Matlab prototype to produce a wall-scale interactive prototype. This system employs three sensors: (a) a touch sensitive SMART Board, (b) a rear-mounted 640 × 480 industrial digital video camera, and (c) a front-mounted 3-megapixel USB still camera to achieve the multiple person, low-latency input and capture needs of the participants in our study. This prototype offers an interactive-rate solution for detecting the location of notes.

The Outpost vision system was written in C++ on top of the Intel OpenCV library (Bradski, 2001). The vision system and the user interface run as separate processes and pass semantic events (e.g., add [x, y, θ], remove [x, y]) through a socket network connection. Currently, both processes run on the same computer. A benefit of this socket architecture is that it allows the processes to run on separate computers without modification. The only reason to avoid running the vision and the UI on different computers is network latency.

When the system starts up, it automatically detects the corners of the board. It does this by projecting a white border on the board and capturing a frame, setting the projector completely black and capturing a frame, computing the thresholded difference image between the two frames, and finding the set of outline pixels. We use EM here as well, finding the best-fitting four lines on the set of outline pixels. We then use the four corners to automatically compute the homography transform. In this prototype, our camera distortion is small. We originally intended to apply a homography transform to every c2 - c1 difference image. We substantially improved the performance of the rear camera's vision system by applying the homography transform only to the logical coordinates of the detected note corners. Thus, we transform only 4 points per note found, as opposed to 640 × 480 pixels per difference image; a savings of roughly four orders of magnitude. This improvement is possible because the only relevant information from the rear camera is the location and orientation of notes, and this information is uniquely determined by the four corners. This optimization is not possible with the front camera because Outpost uses the actual pixels from the front camera to display electronic notes. However, it is not necessary to transform the entire camera image, only each subregion containing a note.

We also revised our mechanism for finding the location of a note. Because the notes have a sticky stripe across the top, the top edge is flush with the board, straight, and accurate. Sometimes the bottom and side edges curl away from the board, however. (This observation generalizes to pictures and other paper artifacts taped onto the board.) Outpost solves this by computing a four-line EM on the note outline and selecting the top line of those four to compute orientation and location. This shortcut allowed us to move beyond the vision details so we could address Outpost's user experience issues. A production implementation of Outpost should avoid this shortcut—by, for example, using a vision algorithm with a stronger notion of shape, such as Shape Contexts (Belongie et al., 2002).

For the most part, this prototype was successful; the main difficulty was process scheduling on a one-processor machine under Microsoft Windows 98® operating system. Often, either the vision or the interface process was given use of the processor for extended periods. Because both need to run interactively to achieve interactive performance, we moved the system to a two-processor machine running the Microsoft Windows 2000 operating system, which resolved the scheduling problems.

2.7. Current Implementation

The current Outpost system consists of two main components. The interface component handles stylus, physical tool, and touch input on the board and provides graphical feedback to the user. The computer vision component tracks and captures physical Post-it notes and pictures.

Physical Tools and Graphical Display

The physical tools input and graphical feedback are implemented in Java using SATIN, a toolkit for informal pen-based user interfaces (Hong & Landay, 2000), and the SMART Board SDK. In Outpost, we make use of SATIN's extensive support for ink handling, gesture recognition, and rendering. Free ink in Outpost is captured and saved as a stroke primitive. Outpost uses a tap interpreter for invoking context menus on existing notes, and a gesture interpreter for drawing links between pairs of notes.

The SMART Board's tool tray consists of four pen tool slots and one eraser tool slot. The hardware detects the presence of the tools via a photometer in each slot. The hardware defines the active tool to be the tool most recently removed from the slot. The tools themselves are passive. The SMART Board SDK uses callbacks to inform registered applications of the current tool. We use this mechanism to know when the move tool or the eraser is active instead of the pen.

Computer Vision Infrastructure

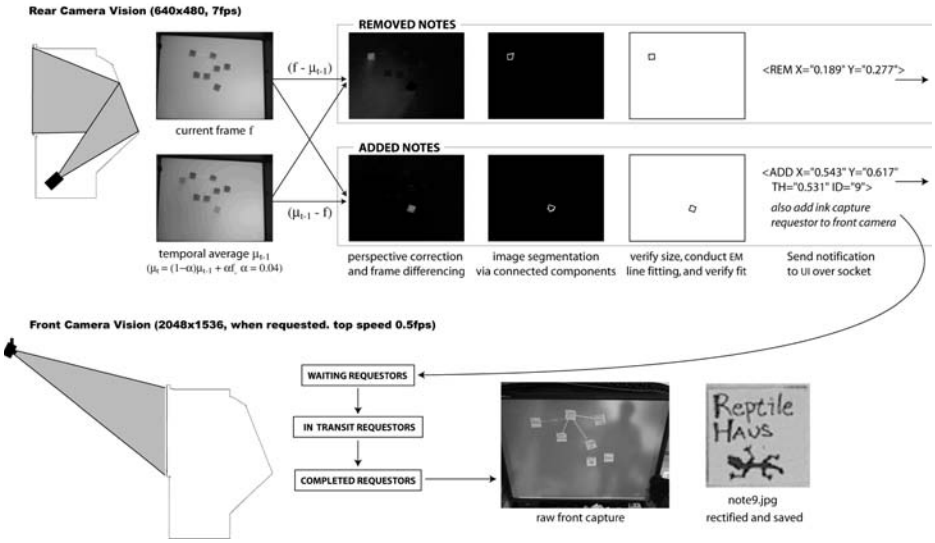
Outpost's vision system supports simultaneous input; essential for collaborative design. Our vision system is written in C++ on top of OpenCV (Bradski, 2001). The vision system runs as a separate process, passing semantic events to the Outpost UI through a socket network connection.

On a dual Pentium III, the two-camera architecture runs at interactive rates (approximately 7 frames per second) for detecting the location of notes with the rear camera, combined with background high-resolution capture (with a latency of approximately 1.5 sec) for virtual display and transitioning to DENIM. This design achieves the multiple-person, low-latency input and capture that our study participants needed. One way to think about the board capture is as a direct manipulation scanner. One operation—placing a physical document on the board—specifies both the location of the document and that the document should be captured.

There are several processing steps that we perform with each new image from the rear camera (see Figure 18). First, we employ spatial and temporal filtering techniques that help alleviate problems because of camera noise and lighting changes. This is a common and effective technique in many computer vision applications. Our temporal filtering computes an exponential weighted moving average image μt by recursively averaging in each new frame ft with weight α. In Outpost's case, the alpha is .04, which is a fairly typical value.

Each frame, we rectify the perspective camera view by bringing the board into a 2D plane using a projective transform matrix. Although there are more

Figure 18. **The Outpost vision pipeline at a frame where one note ("Reptile Haus") was added and another was removed.**


Rear Camera Vision (640x480, 7fps)
current frame f
temporal average μ_{t-1}
$(\mu_t = (1-\alpha)\mu_{t-1} + \alpha f,\ \alpha = 0.04)$
$(f - \mu_{t-1})$
$(\mu_{t-1} - f)$
REMOVED NOTES
ADDED NOTES
<REM X="0.189" Y="0.277">
<ADD X="0.543" Y="0.617" TH="0.531" ID="9">
also add ink capture requestor to front camera
perspective correction and frame differencing
image segmentation via connected components
verify size, conduct EM line fitting, and verify fit
Send notification to UI over socket
Front Camera Vision (2048x1536, when requested. top speed 0.5fps)
WAITING REQUESTORS
IN TRANSIT REQUESTORS
COMPLETED REQUESTORS
raw front capture
Reptile HAUS
note9.jpg
rectified and saved

precise algorithms for camera calibration (Forsyth & Ponce, 2003, chap. 3), we chose a simple perspective warp because it is fast and works well for our purposes.

Next, we construct two thresholded difference images. Objects placed on the board and people moving in front of the board cast a shadow on the board's surface. To the rear camera, areas with objects are darker than the empty board. When an object is removed, the area becomes lighter than it previously was. Added notes are found in the (μt-1 - f) image (darker areas are positive) and subtracted notes in the (f - μt-1) image (lighter areas are positive). We segment the two binary images using the connected components algorithm, finding note-sized components from changed pixels.

After segmentation, we compute the center of mass and the orientation of the note components. We use an EM algorithm (for a good overview, see Bishop, 1995) as a robust method for finding the best-fitting square on the set of outline pixels of the note. We then compute the homography transform from image coordinates into board coordinates.

As a final step, we require that added objects be found in the same place for two consecutive frames. We added this step to reduce false positives to a negligible level. At the completion of this vision pipeline, we send the semantic information about board state changes over the socket to the Java user interface.

To minimize our computational overhead, the front camera only takes a picture when the rear camera detects that a new note *n* has been placed on the board. For each *n*, we add a requestor to the front camera's request queue with [x, y, θ, ID] as the location to capture. As soon as the front camera is available, it takes a picture. The system corrects for perspective skew upon receiving the picture. For each requestor, the system saves the rectified area of the board as a jpeg file. This method ensures that note capture will complete soon after the note is placed on the board (capture completion time is bounded by twice the image transfer time). It also enables multiple notes to be captured from a single image.

Outpost can run in a calibration mode, where it automatically detects the corners of the board and saves the calibration parameters to a file. We do this by capturing a frame of an entirely black board, capturing a frame of the board with a projected white outline, computing the thresholded difference image between the two frames, and finding the set of outline pixels.

Discussion

We designed the vision system to be highly robust at finding notes. The occasional recognition errors fall into four categories:

Missed Actions. There are a few cases where the vision system misses an add or remove action (about 1% of the time). This omission is usually because a person is standing in front of the note, casting a shadow on the board. As visual feedback, the UI displays a faint shadow around recognized objects. When new objects are not recognized, the user must perform the add action again. Missed deletions can be fixed using the Delete option on the context menu.

False Positives. Rarely (about 2–3% of the time), the system reports an action that did not happen. This error is nearly always because the system perceives a user's closed hand to be a note. As a UI solution, we offer the Delete option on the context menu.

Location and Orientation Misreporting. In this system, there are two kinds of accuracy: resolution and calibration. Outpost performs adequately in both regards. As a point of comparison, most of the time our vision system is of higher accuracy than the board's capacitance sensor; a more sophisticated camera model could improve this further.

Occlusion of the Front Camera. Currently, the front camera takes a photograph of the entire board whenever it is requested to. The image-processing system then clips out the requested portion of the image, which corresponds to a photograph of the requested note. On occasion (about 2% of the time), a person is standing between the camera and the board, occluding the camera's view of the note. This happens only rarely because the camera is ceiling mounted and close to the board.

These issues can be remedied with improved vision algorithms (see e.g., Forsyth & Ponce, 2003). Perhaps more important, these errors remind us that intelligent interfaces make mistakes. Our interest in enabling fluid exploration of sensor input alternatives inspired the Papier-Mâché architecture, and the inclusion of edge detection as an algorithm in the Papier-Mâché library (Klemmer et al., 2004).

3. ELECTRONIC DESIGN HISTORY OF PHYSICAL ARTIFACTS

One of the primary advantages of a system that provides digital capture of physical content—such as the Outpost platform described in section 2—is that it becomes possible to catalog, and later replay, earlier states of the physical world. This is important because representations of an artifact's past help us form a deeper understanding of its present. This section presents Outpost's informal history capture and retrieval mechanism for collaborative, early-stage information design. An informal lab study with six professional designers demonstrated that Outpost's history system enhances the design process and provides new opportunities for reasoning about the design of complex artifacts.

3.1. Motivations for History Support

To keep track of project milestones and variations of electronic artifacts, designers currently employ ad hoc methods, usually involving saving multiple versions of files and using complex, cryptic file names to encode the properties of each version. In the physical world, they must manually photograph, photocopy, or scan an artifact to save its current state, or abandon this state and keep working. People invested in understanding the trajectory of history from the past to the present include decision makers, students, designers, and their successors. These stakeholders engage history through creation, revision, and reflection.

Newman and Landay's (2000) study provided us with two important insights into design history tools. First, designers create many different intermediate representations of a Web site. Second, "Designers expressed a desire to have a unified way to manage different variations of design ideas. Variations play a key role during the design exploration phase, and it would behoove an effective design tool to help support their creation and management" (Newman & Landay, 2000, p. 273).

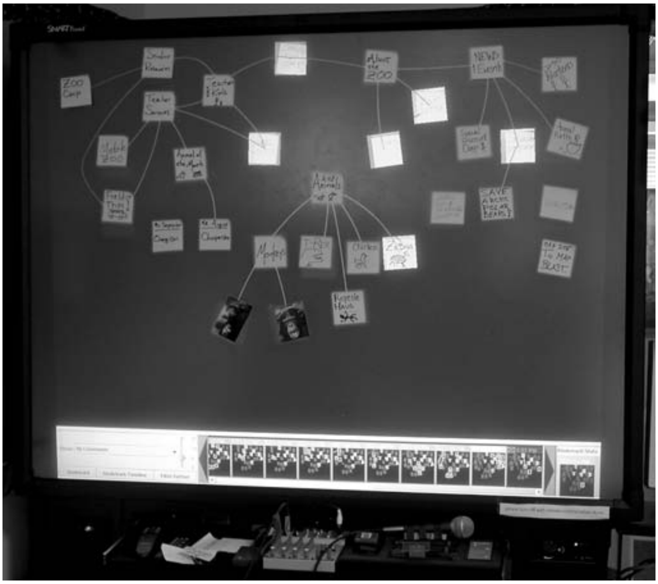
The importance of support for versioning and design history became clear in our initial studies: The participants stated that they often forgot the history of how some part of their design came to be, or they would alter their design and then realize that they preferred an older design. These studies both gave us insight into the working practice of Web designers and motivated our focus on better supporting design history.

We designed the history system around a set of scenarios that we distilled from design fieldwork studies. We present four here: (a) reaching an unproductive point and heading off in a new direction from an earlier point, (b) writing a summary of a design session, (c) finding the rationale behind a decision, and (d) creating a set of action items from a design session.

3.2. History Interface

The Outpost history system provides interfaces for the three primary uses of design history: a main time line presenting the history of the entire design artifact, a local time line of a particular element, and a synopsis view enabling annotation and postdesign review of key states. The main timeline is a visually navigable set of design thumbnails organized on a timeline (see Figure 19). This

Figure 19. **Users' view of the main history timeline (bottom) in The Designers' Outpost.**



view can be filtered by activity (By Actions, By Bookmarks, or By Meeting) or by inferred properties (By Time, By Note, or By Author). We employ a branched history, presenting the current branch to the user as a linear history. This linear history is annotated with stubs, indicating the existence and position of other branches. It is possible for users to jump to any point on the time line, including semantic places such as when an object was created. The local time line enables users to see, in the actual design, a history with just the actions relating to an individual object in the design. The synopsis view enables post-design review of key states that a user has explicitly bookmarked for later retrieval. These bookmarked states can be annotated with text and printed as hard copy for easy portability and sharing.

Outpost introduces a history interface that provides history as a central design organization mechanism (as in Timewarp; Edwards & Mynatt, 1997) while keeping it a relatively calm entity that does not distract from the task of design creation. It is easier, but not required, to work with design history when all of the information is electronic (see Figure 20). We describe the functionality of each of these interfaces and then illustrate their utility in the context of the scenarios. Although Web design is the setting that provided the inspiration and validation of our design history work, the techniques introduced here are in many ways transferable to other professional practices where an artifact is created over an extended time.

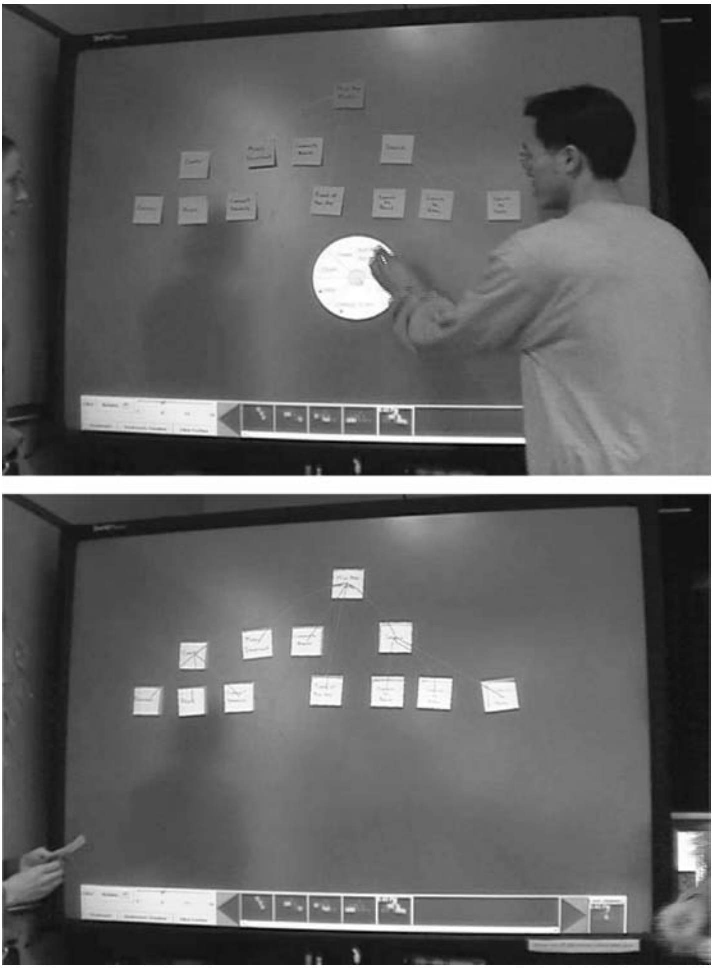
The main timeline displays a history of the design using thumbnails, as seen in Figure 21. Each thumbnail presents the contents of the board at the time of capture. To support the user in determining what has changed between adjacent frames, we highlight the elements that were altered in the most recent frame (see Figure 22).

Filtering Thumbnails

Designers use the time line display to choose the set of thumbnails to display. In the most detailed view, the time line displays all thumbnailsone thumbnail per single action the users have performed at the board, such as adding a note or moving a relation. View all is useful for local undo, but more substantive interactions mandate using the other filters.

There are two types of filters: explicit filters and implicit filters. The filters based on explicit user activity are By Actions, By Bookmarks, and By Meeting. For each of these, the user can select to have a thumbnail generated for every *n* bookmarks, actions, or meetings. Implicit filters produce a view using metadata properties of the design itself; they are By Time, By Note, and By Author. In the By Time filter, the user chooses to see only frames that correspond to actions carried out every *n* seconds. With the By Note filter, the user can select a note on the board to view only frames that correspond to actions

Figure 20. **Outpost's electronic capture enables replacing physical documents with their electronic images. A pie menu operation (left) makes all notes electronic (right). It is easier, but not required, to work with design history when all of the information is electronic.**



performed on or in relation to that note. Finally, when using the By Author filter, only frames that correspond to notes altered by the chosen author are displayed.

The By Author filter is one example of many possible context-sensitive history queries. Here, the system needs to sense who is the author of each operation. Although this could be implemented using an RFID tag and a reader be-

Figure 21. **The main time line at the bottom of the SMART Board. The pop-up pie menu lets users choose available filters. Bookmark adds the current state to the synopsis. Bookmark Timeline adds all states in the current view to the synopsis. Filter Further allows users to intersect filters.**

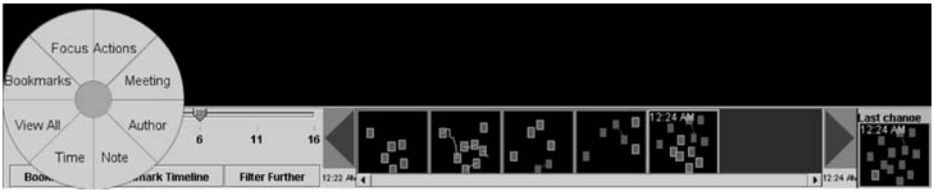
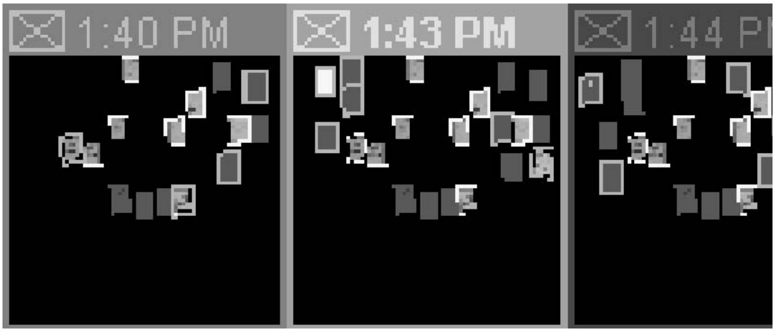


Figure 22. **Close-up of the global time line. Above each thumbnail is a timestamp. The main thumbnail is a scaled-down version of the board, with the changes highlighted in green. The frame around future thumbnails is dark blue, past is medium blue, and current light blue.**

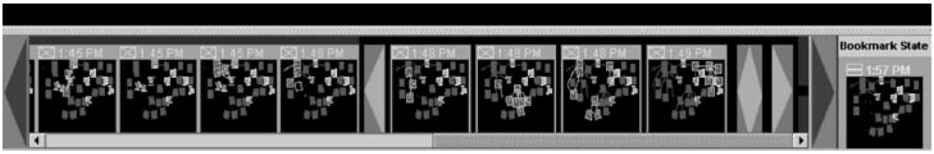


hind the board, we currently simulate this using a Wizard of Oz approach (Kelley, 1984; Maulsby et al., 1993): During design sessions an operator indicates the person currently at the board in a simple list of names shown on a secondary monitor.

Timeline Navigation

Thumbnails display information about the state changes of the board and provide a direct-manipulation interface for navigating the design history. Pressing any thumbnail causes the system to undo or redo all commands that have been issued since that point, restoring the board to that state. This multilevel undo/redo allows designers to experiment with the information design, as returning to a previous state is always possible (see Figure 23). Although the touch-based graphical interface works very well

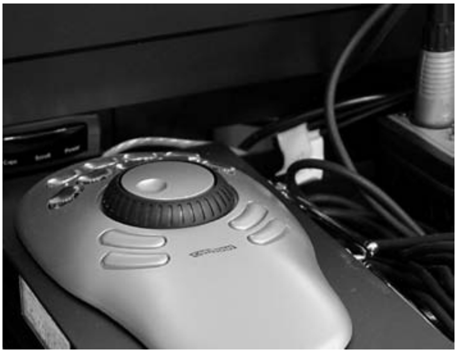
Figure 23. **The main time line, with an expanded strand containing a collapsed strand.**



for selection, we found soft (graphical) controls to be clumsy and slow for scrolling and browsing because virtual widgets lack tactile feedback indicating when the widget is acquired and manipulated. To support more fluid scrolling, we integrated a Contour Design USB jog dial (see Figure 24) for direct physical interaction with the history timeline. Snibbe et al. (2001) showed that jog dials with haptic feedback are highly effective for browsing time-based media. Presenting semantic information (such as branches) haptically would likely be of great benefit to users of design history systems.

Although our studies have shown many benefits to a paper-centric tangible interface for freeform design, the physicality of a design artifact becomes problematic when engaging its history: It is not possible for the system alone to perform an undo operation for all possible physical actions made by the user, such as adding or removing a note from the board. A combination of user actions and history manipulations can yield one of two degenerate cases: Either the current view calls for presenting an object without a physical presence, or a physical object is present that should not be. In the former case, we display the electronic capture of the object. In the latter, we give the object a red shadow to indicate it should not be present, hinting that it should be removed by the user.

Figure 24. **Physical jog dial for scrolling through history.**



Branched Time Visualization

As previously mentioned, the system supports multilevel undo/redo. Most contemporary software, such as Microsoft Office®, provides multiple undo/redo with the limitation that only one strand of actions is held in the history. If the user performs actions A, B, C, D, E undoes three times, and then performs actions F and G (the sequence shown in Figure 25); the current action strand in the history is A, B, F, G. The fact that C, D, and E used to follow b is lost in a linear history. Some undo/redo systems, such as the one in Emacs (Stallman, 1993), offer the user a truly branched history. However, branched histories have traditionally been difficult to navigate; the user is likely to get lost because it is difficult to build an accurate and complete mental model of the history tree.

Our goal was to preserve the entire history, with all constituent action strands, without introducing unwieldy complexity. We achieved this by merging the concept of a branched action history with the linearity of a single-stranded history. One possible way of presenting the branched history is as a branched tree, as in Figure 25. Although this way of displaying the multiple strands presents the whole history and its two constituent strands, the visualization rapidly becomes complex as the number of branches increases. This complexity creates too large a user burden for our domain of informal, early stage design, and it requires substantial screen real estate.

As a lighter weight alternative, we present the history as a linear list of actions, where inactive branches are represented by a collapsed strand, shown in the top row of Figure 26. This presentation preserves the temporal order of the actions; a frame presented to the right of another frame corresponds to an action that was issued after the other. It also scales well; multiple branches can be shown inside each other by nesting the stub parenthesis markers as shown in the bottom row of Figure 26. Users can open or close (collapse) any branch, choosing a presentation of the time line relevant to the objects of interest.

Figure 25. **Branched history: Actions A, B, C, D, and E form one strand; A, B, F, and G form the other.**

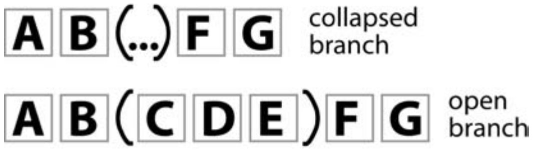
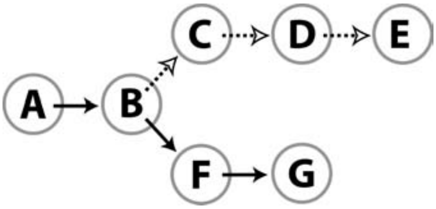


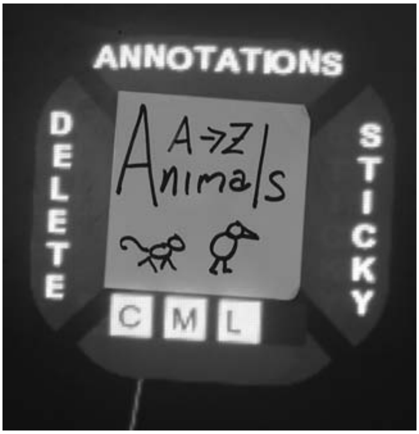
Figure 26. **Stub-branching history presentation: the top history fully displays the current strand; other strands are visualized as stubs. The bottom history displays the full history; states not part of the current strand are placed between parentheses.**



Local Time Line Visualization

The main time line visualizes the history for the whole board; the local time line provides a lighter weight history for an individual object. When selecting a note by tapping it, an object menu is displayed (see Figure 27). The object menu supports common operations such as deleting the note or making it persistent, as well as displaying a small note history along the bottom. This novel in situ time line offers the user more detailed information about a particular note without visually cluttering the entire board.

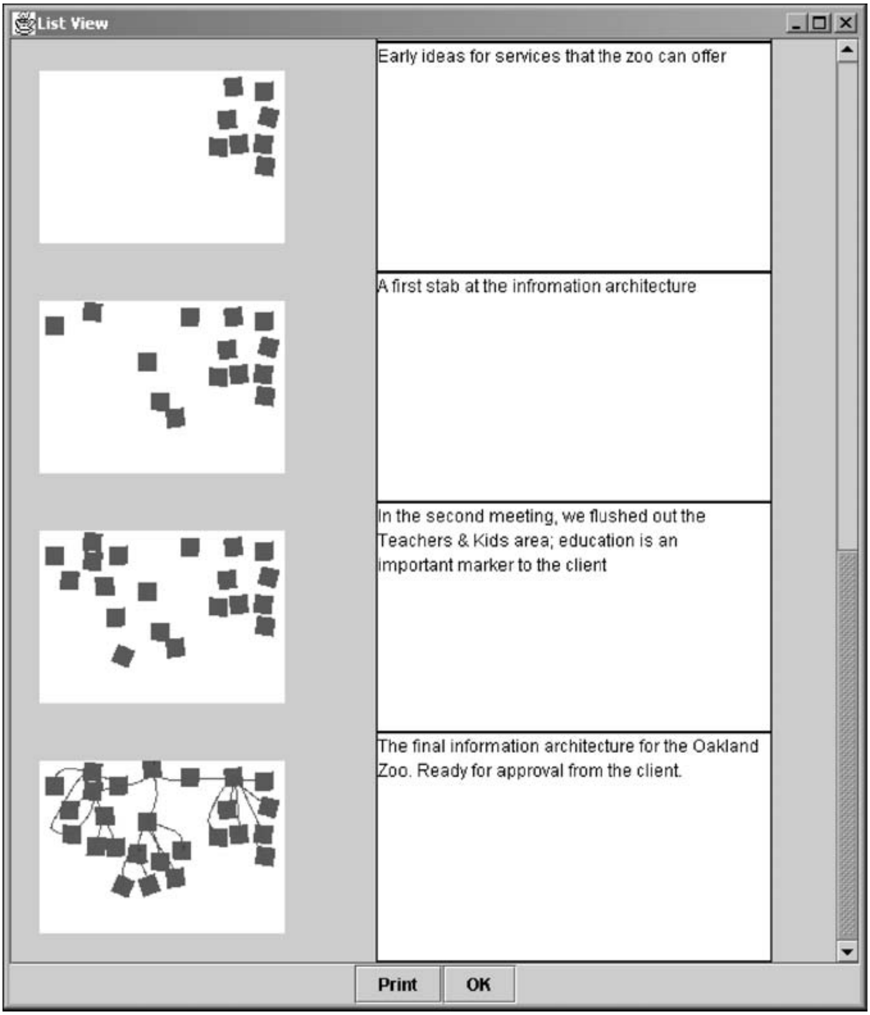
Figure 27. **The electronic context menu for physical objects. The bottom element in the menu is the local timeline. In this case, the note was created ("C"), then moved ("M"), and finally a link was drawn ("L"). This local timeline is display-only.**



Synopsis Visualization

One advantage of electronic capture is its ability to support radically different presentations of information. The synopsis visualization is an example. This visualization presents an annotated record of the set of changes that happened to a design. It provides the ability to work with this list on a secondary display (see Figure 28) or printed on paper (see Figure 29). Because users are

Figure 28. **The on-screen synopsis view.**



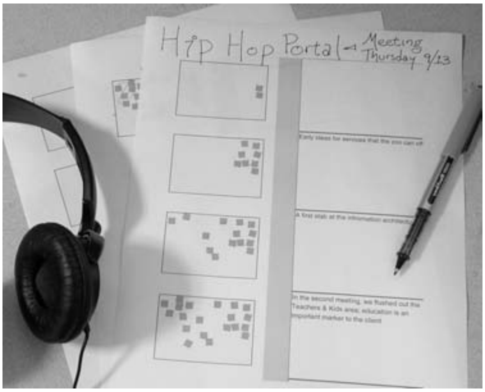
Early ideas for services that the zoo can offer

A first stab at the infromation architecture

In the second meeting, we flushed out the Teachers & Kids area; education is an important marker to the client

The final information architecture for the Oakland Zoo. Ready for approval from the client.

Figure 29. **A print version of the same information.**


Hip Hop Portal → Meeting Thursday 9/13
Early ideas for services that the site can of
A first stab at the information architectu
In the second meeting, we fleshed out the Teachers & Kids area; education is an important marker to the client

not always at the board, a printout serves as a take-away design record for sharing, discussion, and annotation. The synopsis visualization meets all three of these needs.

A synopsis can be constructed in two ways. First, it can be constructed via explicit user bookmarks. Users can bookmark states as the design unfolds or by returning to an earlier state and bookmarking it. Users can view their set of bookmarks when viewing By Bookmarks. Users can also construct a synopsis from the set of states in the current filtered timeline by selecting Bookmark Timeline. These techniques can be combined to manually augment an auto-generated state set. For example, a user could begin a bookmark set with the states produced from the By Meeting filter, augmenting it manually with key points from the meetings. This combination of automatic and manual history echoes work by Kaasten and Greenberg (2001) on managing Web browsing histories.

When viewing the main time line by bookmarks, the control area at the left of the time line contains a button to bring up a synopsis view on a secondary display. The synopsis view presents each of the bookmarks vertically on the left side of the window. It provides a text box to the right of each bookmark for entering a description of that state. The synopsis view can also be printed for offline use.

3.3. History Usage Scenarios

Drawing on field studies of Web design (Newman & Landay, 2000; Newman et al., 2003), our initial laboratory studies, and on the related research literature (Bellotti & Rogers, 1997; Moran et al., 1997; Richter et al., 2001), we constructed four scenarios that reflect current and envisioned uses for Outpost's design history capture.

Del, Erika, Jared, and Ray are designing a portal Web site for hip-hop music and culture. This portal will enable site visitors to read music reviews, interviews with artists, and relevant news stories, as well as purchase music for download and find out about local concerts.

Reaching a Dead End

In the first design session, Erika, Jared, and Ray came up with a draft of an information architecture for the portal, which they are now trying to refine in a second session. After reviewing the initial design, Ray points out that the music reviews section is completely disconnected from the purchasing music section. If they are to profit from the site, the two should be strongly tied. After a while, Jared concludes that it isn't possible to alter the current design, so she creates an ink annotation on the main "music review" note explaining this. She taps the "music reviews" note to select it and then selects the By Note view on the main timeline. The first thumbnail in this view displays the board state when the note was created. She taps this thumbnail to revert the board to that point in time. From there, they redesign the site so that purchasing is easily accessible from the music reviews area.

Writing a Session Summary

After the design session, Erika stays in the project room. She uses the main time line to review the team's progress. As she rolls the time forward By Actions she bookmarks important states in the design. Upon reaching the end of the meeting, she opens the synopsis view (see Figure 28) and annotates the key states with text. Finally, she makes a print version (see Figure 29) for herself and the other team members. This portable, sharable summary serves as an overview of what has been accomplished and helps the team members communicate their progress to the client.

Find the Rationale Behind a Decision

Del missed the design session; he was helping friends set up for a show. When he returns, he views the electronic wall in the project room and notices the strong linking between the music reviews and the music purchase areas. Curious why this is, he taps on the "music reviews" note to reveal its local history, which he scans to understand the changes it underwent while he was away. From the note's context menu, he brings up the note's annotations. Reading the ink annotation written by one of the other designers, he quickly understands the rationale for the change.

Following Up on a Session

On Friday morning, the designers decide to perform a review of their work in the past week. Several design issues warrant further consideration; they bookmark each of these. They then annotate the bookmarks in the synopsis view and print it. The print view serves as a to-do list that the designers bring back to their personal workspace. In the afternoon they reconvene and discuss the issues that each has examined, yielding a much cleaner sitemap.

3.4. Implementation

At the core of the history system is a data structure that holds command objects (Gamma, Helm, Johnson, & Vlissides, 1995; Myers & Kosbie, 1996)—one command object for each action carried out by the users. A command object is a software design pattern that encapsulates an atomic action as a software object called a command. These objects are stored in a command log, facilitating robust undo and redo. Some command objects in the Outpost system are Add Note, Remove Note, Move Note, Add Link, and Add Ink Annotation.

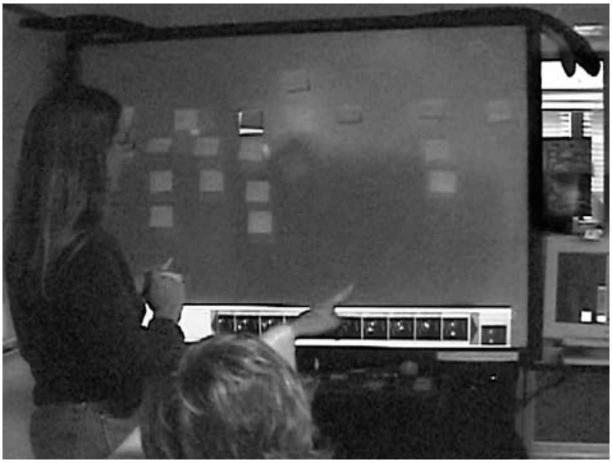
These command objects are stored in a tree-shaped data structure with branches. A new branch is added when the user jumps back to a previous state and then starts modifying the board from there. The actual restoration of the board's state from a given state *x* to the user requested state *y* is handled by first calculating the least common ancestor l of *x* and *y*, then the up-path from *x* to l and finally the down-path from l to *y*. Given these paths, the state is easily restored by following the path from *x* to l undoing each command on the way, and then the path from l to *y* redoing the commands found here.

Each of the thumbnails used to visualize the command history is calculated by asking the main SATIN sheet (Hong & Landay, 2000) to redraw itself into a new, thumbnail-sized graphics context. The entire set of thumbnails is redrawn each time the filter changes: Given a criterion, the whole tree is traversed and the whole visualization rebuilt. It would be computationally more efficient to cache some of this information, but we have found, as others have (Rhyne & Wolf, 1992), that for a research prototype rolling forward is not a substantial bottleneck. Computational efficiency was not the focus of this research, and this simple approach proved to be fast enough for medium-sized designs. A production implementation of this system would achieve faster performance by periodically caching state.

3.5. Design Study

We had six professional designers use the history system and offer their feedback (see Figure 30).When the history system was in an early state, we

Figure 30. **Two professional designers collaborate on an information architecture for the Oakland Zoo Web site during the study.**



brought in two designers from the same firm to talk with us about their current practices and try our system. In prestudy interviews with the pair, we learned that the participants currently had a difficult time managing history; their state of the art was to save “bookmarks” and “versions” simply as files with different names. When working with our system, they primarily used the main time line at a macroscale. Working physically and electronically occurred in cycles. They would add content for a while, work with it, then make the board electronic, and delve into the history. In addition to finding the history useful for reflection and design rationale, the pair commented that they would find value in using the history to make accountability from the client clearer. The pair’s collaborative work helped us to realize that knowing the author of content might be beneficial, leading us to implement the author wizard for the next group of participants.

With the software completed, we brought in four more designers in three groups: one pair of colleagues and two individuals. The format was similar to the study described in section 2.3. It began with an explanation of the system, continued with a design task lasting roughly 1 hr, and concluded with a 19-question survey (Klemmer, 2004, Appendix A). The participants were very enthusiastic about our bookmarking features and the ability to generate a synopsis view.

Participants found the View All filter distracting, reminding us of the need for calm interaction (Weiser & Brown, 1997). Viewing every single command is only useful for local undo, rare during fluid brainstorming. (The one time it proved useful was when the system misrecognized the users' input.) One finding from our previous study was that calm interaction is essential to an effective electronic whiteboard. Beyond its limited utility, View All is the antithesis of calm; it renders a new thumbnail for every command the user executes, which makes for a hyperactive electronic whiteboard. Based on this finding, we changed the default filter to be the By Actions filter. This provides visual locations the user can move back to at a coarser interval (the default is every six).

One participant commented that his favorite aspect of using computer-based tools was that easy saving enabled him to try new ideas and have different versions. After 3 of the participants had worked with the system, it became clear that save and bookmark should be integrated. We eliminated a separate Save button and included the save functionality as part of the bookmarking process for the last participant. He found this integration intuitive.

Timeline Usability

The participants were enthusiastic about the history's ability to easily capture different states. Having a simple, touch-based visual interface with the ability to negotiate the history of the board was highly appreciated as well. The participants used the history smoothly for the most part, but sometimes the presence of branches was confusing. As a solution, one designer suggested that it might be valuable at times to see the entire branch structure as a traditional graph.

Need for Visual Comparison and Merging

The designers encouraged us to provide facilities for simultaneous comparison and merging of history states. One participant said, "It is very important to view multiple versions in juxtaposition, at the same time and at a scale that we can make sense out of. Much of the impact is visual." The work by Terry and colleagues on supporting simultaneous development of alternative solutions provides an excellent set of UI techniques for viewing multiple alternatives simultaneously (Terry & Mynatt, 2002; Terry, Mynatt, Nakakoji, & Yamamoto, 2004). Outpost's history system would likely benefit from such techniques.

One designer commented that in his current practice, "When I'm working, I'll do the information architecture on Post-its, and draw links on the whiteboard. I'll take snapshots at different points in time. And then I'll project earlier states onto a wall, and go from there." This was a current practice uncannily similar to Outpost and its history facilities. (The other participants did not have this advanced a practice for dealing with history, possibly because such a practice is difficult with current tools.)

4. TANGIBLE REMOTE COLLABORATION

Having completed an infrastructure for time-shifting the content of walls, we then turned our attention to space-shifting physical content. A tension exists between designers' comfort with physical artifacts and the need for effective remote collaboration: Physical objects live in one place. Previous research and technologies to support remote collaboration have focused on shared electronic media. Current technologies force distributed teams to choose between the physical tools they prefer and the electronic communication mechanisms available. This section presents Outpost's remote collaboration functionality. We extended the system for synchronous remote collaboration and introduced two awareness mechanisms: transient ink input for gestures and an outlined shadow of the remote collaborator for presence. We informally evaluated this system with six professional designers. Designers were excited by the prospect of physical remote collaboration but found some coordination challenges in the interaction with shared artifacts.

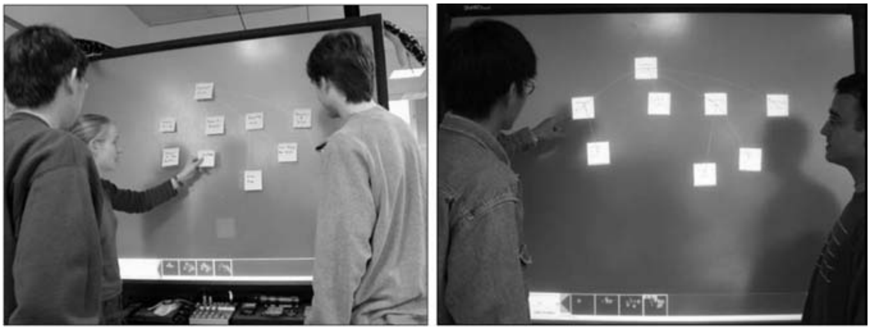
4.1. Motivation for Remote Collaboration

Many designers we have spoken with work in collaborative teams at multiple locations. When working with their remote colleagues, they are forced to choose between the physical tools they prefer and the electronic communication mechanisms available. The designers felt that consensus building was vital to their work process, as it establishes deep relationships, especially when participants have different backgrounds (Arias, Eden, Fischer, Gorman, & Scharff, 2000). However, it is more difficult to build relationships without a sense of physical presence.

This section discusses how structured capture enables fluid remote collaboration (see Figure 31). To better support remote collaboration, we introduce an interaction paradigm where objects that are physical in one space are electronic in the other space, and vice versa. This paradigm has the potential to enable more fluid design among distributed teams but must also overcome the problems of maintaining awareness between distributed groups.

We present and evaluate two mechanisms for awareness: transient ink input for gestures and a blue shadow of the remote collaborator for presence. The transient ink is a pen-based interaction technique for conveying deictic (pointing) gestures. Users mark up the board to suggest changes or relationships without permanently cluttering the workspace. Transient ink is dis-

Figure 31. **Remote collaboration with Outpost: notes that are physical in one place (see left) are electronic in the other (at right). The Outpost history bar at the bottom shows previous states of the board.**



played on both boards for a few seconds and then fades away. The mechanism for presence awareness is a blue shadow that represents the location of the remote participants with respect to the shared workspace. Users of the system can get a sense of the locations and intentions of remote collaborators without needing their physical presence.

The Designers' Outpost was originally a single location interface. We extended Outpost to communicate between two remote hosts. The shared communication consists of user actions (e.g., adding and moving notes) augmented with remote awareness information (a vision-tracked shadow of the remote users and transient ink).

4.2. Interviews and Fieldwork Informing Design

Previous fieldwork and design studies (Bellotti & Rogers, 1997; Newman & Landay, 2000) have found that designers often need to collaborate with colleagues and clients who are not in the same office or even the same city. We brought six professional designers into our laboratory to provide feedback on Distributed Outpost. We first asked them to discuss their current remote collaboration practices. The designers described several important collaboration tasks, including consensus building; concept mapping; user focused design solutions; and defining project features, function, and interaction.

Current Experiences With Remote Collaboration

Working with remote participants is a "nightmare," stated one designer. The designers expressed three primary frustrations with their current collaboration tools. First, they felt that their interactions with remote colleagues were impoverished. Second, they felt the tools well suited to collaboration (e.g., e-mail, telephone), were ill suited to design. Finally, all of the designers in our study had developed ad hoc methods when designing with remote colleagues. The study participants reported four different methods for working with remote collaborators.

Whiteboard, Video, and E-mail. One group maintained their physical practice of using a whiteboard with sticky notes at a central office. Remote participants can view the screen though a video link, however, their participation is severely limited. Distributed workers send e-mail to the facilitator when they have input. Thus, they are reliant on the facilitator for their participation in the design session, and there is a time lag between their contribution and its visibility to the rest of the group. As a group member stated, "This makes it almost impossible to have active participation of remote participants."

Two Whiteboards and Videoconference. Occasionally, both offices will have sophisticated videoconferencing technology. Designers work on two separate, manually synchronized whiteboards with Post-it notes. Each side has a remote controlled pan/tilt/zoom camera. The technology is adequate for viewing the distributed boards, and the resolution is high enough to view written text. However, there are significant pauses in the interaction while one side zooms the camera in to see a change, and there is trouble keeping the separate representations consistent.

Collocated Meetings (and Occasional Conference Calls). Another group was limited to only generating ideas when they were collocated. Once the ideas were generated, the potential design was typed into a computer for sharing with the remote clients. When meeting with clients in a conference call, each person had his or her own paper printouts on which the person recorded potential changes to the design. Later, these designs were synchronized by the designers in a discussion meeting to come up with the final design.

Visio and E-mail. Another participant developed designs alone with Microsoft Visio® software, a graphical diagramming tool. When it came time to collaborate, he would e-mail the document to another user, who would change it and e-mail it back. Some of his colleagues did not have this tool and thus worked on paper printouts and had him enter the changes into his document. This setup made real-time collaboration impossible and added significant lag to the design process.

User Needs for Remote Collaboration

One of the largest problems we identified was a lack of a shared workspace. For large, remote teams, it can be hard to maintain focus without a shared artifact to discuss. It is also difficult for remote users to gesture or convey spatial relationships when they do not have access to the items under discussion. The formality and constraints of current technologies also interrupt the flow, making design more difficult (Landay & Myers, 2001).

Many designers stressed the importance of establishing common ground with the people they worked with. "It's not the end, it's the means," one designer explained. Consensus is vital for moving forward in the project. When the participants have different backgrounds, it becomes especially important to establish deeper relationships.

The designers we interviewed found it difficult to establish a rapport with distributed participants. They said they felt disjointed from their peers working remotely. This remains a problem even with a sophisticated video conferencing setup. Latency, a lack of presence information, and out of sync artifacts remain barriers to effective collaboration.

4.3. Interaction Techniques

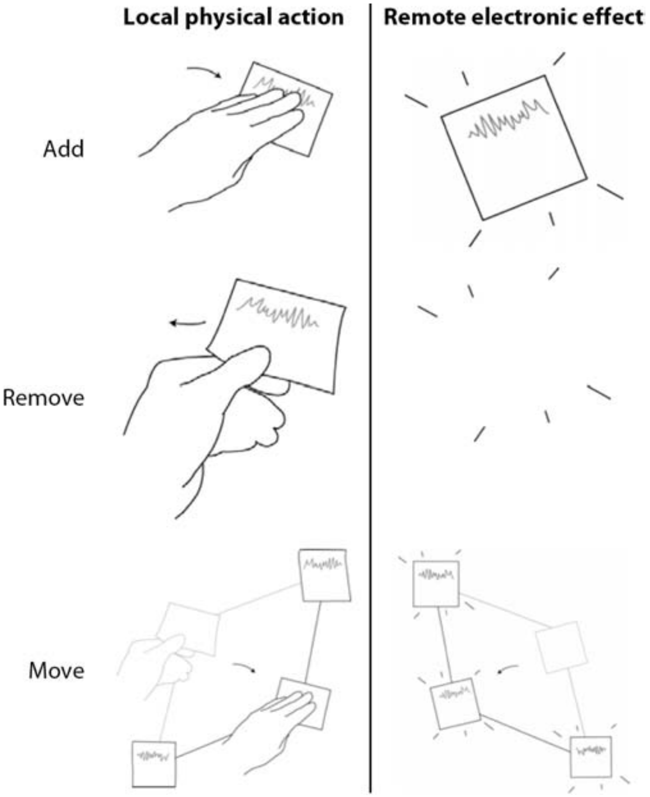
Outpost addresses designers' needs in two ways. It provides a unified workspace with support for spatial gestures between remote colleagues. It also provides presence and awareness mechanisms to help remote participants establish common ground. In supporting these requirements, we felt it was important to keep the physical interaction. Our earlier studies also demonstrated the importance of calm interaction; calmness was a design goal for representing remote presence, as it was with the AROMA system (Pedersen, 1998; Pedersen & Sokoler, 1997).

Shared Workspaces and Transactional Consistency

Our system consists of a shared workspace though which groups of designers can interact. Several designers can participate at once when working with the board. The computer vision system supports simultaneous input of several Post-it notes.

A note is created by writing on a physical Post-it note and placing it on the board. When a local user physically adds a note to the whiteboard, the remote system electronically displays a photograph of that object (see Figures 32 and 33). The vision system's rear camera locates the note and the front camera photographs it. When any user performs an action, both the local and the remote system are updated. Both teams can interact with any

Figure 32. **Interaction techniques for creating, deleting, and moving physical notes in Remote Outpost. The left column is the user's action with the physical note; the right column shows the electronic display on the remote board.**

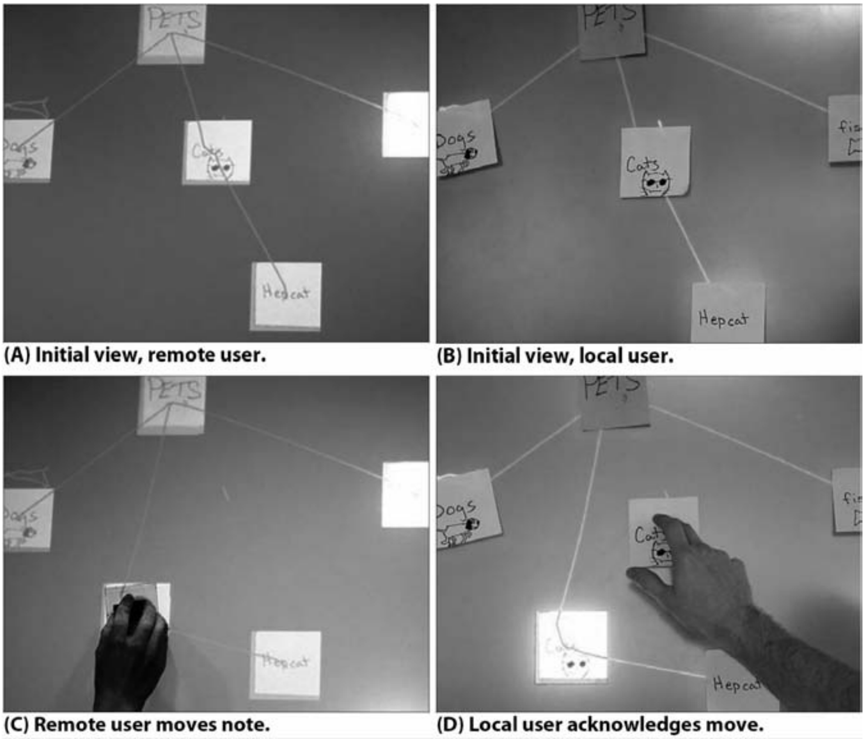


note, regardless of whether it exists as a physical object or remote analogue.

To delete a note, the user removes it from the board (see Figure 32, middle row). To move a note, the user picks it up and places it in the new position (see Figure 32, bottom row). The Outpost prototype does not recognize specific notes based on content; as a proxy heuristic, it assumes that the note is the same if it is replaced within 7 sec (see section 2.2).

Ideally, both teams should be able to edit and move all objects. When the objects are electronic (such as with links), this is easily facilitated. When the objects are physical (such as with Post-it notes), editing them from multiple sites introduces some difficulty. One option is to only allow the creator editing ability (Moran et al., 1999); that is not very appealing.

Figure 33. **Moving a note: A and B show the remote and local views before the move. In C, a remote user moves the electronic version of the "Cats" note with the move tool. D shows the virtual Cats note at the new location and the local user removing the out of date physical Cats note (marked with a red shadow).**



(A) Initial view, remote user.

(B) Initial view, local user.

(C) Remote user moves note.

(D) Local user acknowledges move.

We have taken an alternate approach. Post-it notes in Outpost cast electronic shadows as feedback to the user that the system is aware of their presence. When a note's physical state becomes transactionally inconsistent, the system casts a strong red shadow indicating to the user to remove the artifact (see Figure 33D). The red shadow identifies that the physical note is no longer an information handle to the virtual remote analogue. This feedback is lightweight; it provides awareness that the note is out of date but does not require the user take any action.

We originally introduced the red shadow feedback in Outpost's design history system. There, it identifies notes that are out of date with respect to time. Here, it identifies notes that are inconsistent with the remote users' board.

There are two user actions that make a physical note transactionally inconsistent: deleting and moving. If the note is deleted by the remote user, the faint recognition shadow is replaced with a red shadow (see Figure 33D). The local user could remove the note to dismiss the shadow or repost the note if they disagreed with its removal. When a note is moved, a red shadow displays behind the out-of-date physical note and a virtual note appears in the new position (see Figure 33). The local user could then remove any physical note with a red shadow.

When a note is virtual, the physical handles are missing and must be replaced with electronic controls (see Figure 34). In this case, a note context menu is available for deleting notes, and the physical move tool is available for moving the notes as described in section 2.2.

Flexible Deployment

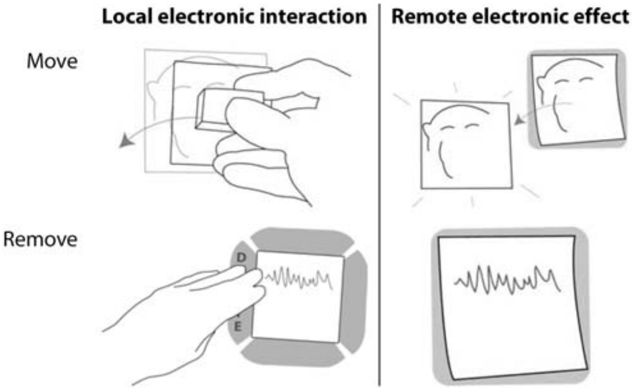
Distributed Outpost works on any hardware with a 2D input device, for example, an interactive whiteboard, a tablet PC, or a desktop PC. Although users benefit from the computer vision tracking and capture of physical objects, it is possible to create notes with a tap and draw on them with the stylus tool. Links can be added by drawing a line between two notes. Erasing and moving ink and links is supported with the stylus button.

Although the setup lacks the tangible advantages of Outpost, users can still work with the notes using pen-based design interaction. This setup is more cost effective and flexible for remote participants with limited resources.

Transient Ink for Deictic Gestures

An important affordance of remote collaboration systems is the ability to convey deictic gestures. Without this, it is difficult for users to understand

Figure 34. **Interaction techniques for moving (top) and deleting (bottom) electronic content.**



what their remote colleagues are communicating or to express their opinions on relationships.

When users want to draw their collaborators' attention to a particular spatial position or artifact, they need some way to convey this deictic gesture. We found that a simple remote cursor (U.S. Patent No. 4,317,956, 1982) did not convey enough information. For this reason, we developed transient ink as a richer interaction technique for enabling distributed users to convey deictic information to each other.

Users draw electronic transient ink on the board with the red stylus tool (see Figure 35). The ink is rendered on both displays for a few seconds, and then it fades away. This allows users to convey relationships, suggest links, and point to notes without committing their changes and permanently cluttering the board.

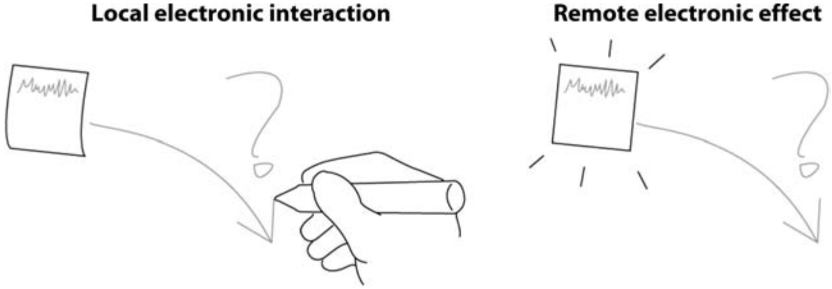
When using the board interface, a specific stylus is used to create transient ink. In the desktop setup, it is selected from a pie menu. In addition to transient ink as a mechanism for conveying gesture, Outpost employs computer vision techniques to provide stylized shadows of people to help provide a rough idea of remote collaborators' locations around the board.

Distributed Presence

A sense of presence is important to developing a working relationship with remote colleagues. However, the designers we interviewed did not feel that the currently available videoconferencing and audioconferencing technologies provide a sufficient sense of presence to establish a rapport.

Our presence shadow is inspired by Clearboard (Ishii, Kobayashi, & Arita, 1994) and VideoWhiteboard (Tang & Minneman, 1991). The seamless inter-

Figure 35. **"Should this note be moved down here?" Transient ink is used to convey pointing information and temporary graphical material by a remote user. The written ink fades away after several seconds. The writer's view is on the left; the receiver's view is on the right.**



action paradigm put forth in these systems is particularly appropriate to support awareness for our system. It is important that the presence mechanism be calm and nondistracting, allowing designers to focus on the task.

We extended the rear camera's vision processing, used for detecting notes, to detect peoples' shadows on the board (see Figure 36). As a person casts a shadow on the board, we determine if it is the appropriate size and darkness for a person. If so, the vision system calculates the shadow boundary. If more than one person is working at the board, the awareness will show multiple shadows.

An early version of the remote awareness, used for the feedback session, displayed a translucent blue oval based on the center point, width, and height of the detected shadow. We implemented the feedback in this manner because it was the simplest from an implementation standpoint. However, not surprisingly, our study participants found that this was not sufficiently expressive.

The current presence visualization is a stylized shadow outline of the remote users, displayed on the background of the design surface (see Figure 37). This shadow conveys the remote users' presence, gesture, and location in a lightweight fashion.

All content and presence information is sent using sockets over IP, unlike prior work (Ishii et al., 1994; Tang & Minneman, 1991), which required a dedicated video link. We present a shadow instead of a live video image of the user because it is calmer; live video is too distracting when interacting with whiteboard content. Our goal was to subtly display and communicate information that is not part of the user's primary foreground task.

Figure 36. **The view from the rear camera of two users, one of whom is pointing to a note on the board. The calculated borders of the shadows are drawn in white, on top of the raw pixel input.**

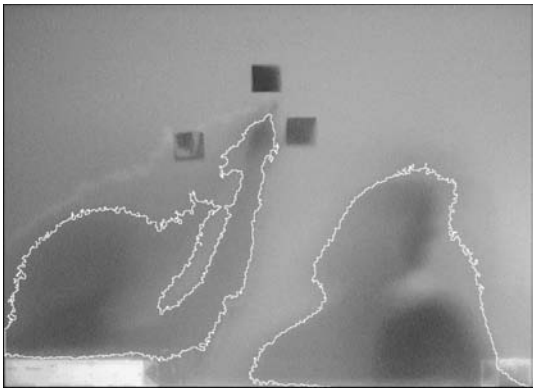
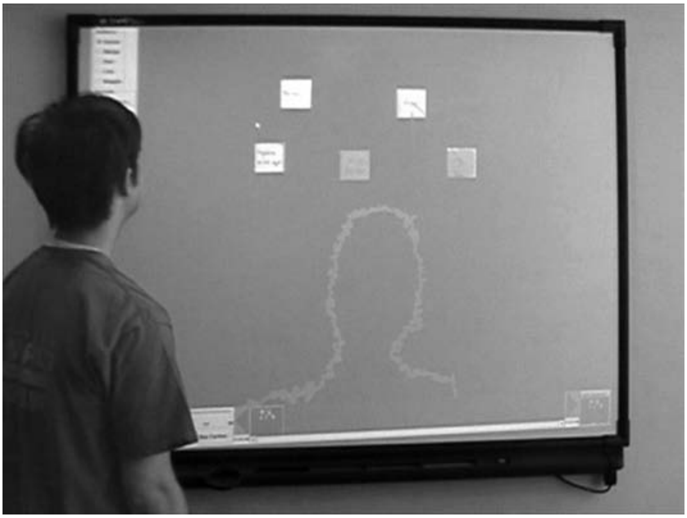


Figure 37. **The distributed awareness mechanism. A blue shadow outline in the background represents a remote collaborator.**



4.4. Software Infrastructure

Our remote collaboration system extends The Designers' Outpost. Two sets of additions were required. First, we extended the command object system to replicate state across a network for synchronizing shared applications in a manner similar to Berlage and Genau (1993). Second, we added functionality to the vision system to support distributed collaboration.

Data Transfer

Outpost designs are serialized to files as XML documents. It uses the same XML serialization mechanism for network communication. The system works as a peer-to-peer system with socket connections between machines; both endpoints replicate their commands, sending the corresponding command objects to the opposite endpoint. Each object has a unique global identification tuple, composed of its creator's hostname and an integer corresponding to its position in the local command queue. This identifier is used to refer to objects between hosts. We modified the SATIN command queue so that when a command is executed, it is also marshaled for serialization over the wire and sent to the remote host. Because most of the changes were made at the toolkit level, other SATIN-based applications can benefit from this infrastructure with minimal application-level change.

At present, photographs of notes are stored as JPEG files on a networked file server. They are accessed as needed over the file system by the hosts. This could easily be modified to use a Web server to support collaboration between distinct organizations.

Vision and Tracking

The Outpost vision tracking system uses the rear camera to track notes (see section 2.6). For remote collaboration, we extended the vision system to find shadows of users working at the board. The system processes the image using spatial and temporal filtering, corrects for perspective distortion, and computes a running average of the expected background image.

It constructs three thresholded difference images. Possible pixels from added notes are found by subtracting the current frame from the expected average image. Potential shadows are found the same way. They have a lower threshold than for notes, because a person's shadow cast from standing in front of the board is not as dark as the shadow cast by a note stuck directly onto the board. Potential removed notes are found by subtracting the expected average image from the current frame.

At this point, it segments the binary images using a connected-components algorithm. The found elements expected to be notes are subjected to size and shape restrictions using an expectation maximization algorithm before being classified as notes. The person objects are also subjected to size restrictions of 0.5% to 40% of the board.

The vision system runs as a separate process, passing events (e.g., add [x, y, θ, ID], remove [x, y], and addPerson [x, y, w, h]) to the local Outpost UI through a socket network connection.

4.5. User Feedback

To understand how Outpost's remote interaction techniques are used by the target community, we had six professional designers visit our lab: one group of three, one group of two, and one single user. We asked them to come prepared to design a Web site of their own choosing. Each session lasted 1.5 to 2.5 hr. The study began with an oral interview about participants' current remote collaboration practices. Then we introduced Distributed Outpost and had participants use it to design their site.

Because of the technical constraint of having only one full Outpost board setup, the groups worked with one Outpost Board connected to a VisionMaker digital desk. The input for the digital desk was an input-only Wacom Graphire pen tablet. (One participant used a mouse instead of a tablet.) Also, Outpost does not handle remote audio communication internally; it presumes that an audio conference infrastructure is already in place. For our feedback sessions, the board and the desk were located in the same room. The participants were allowed to speak to each other but unable to see each other because of a curtain.

The one difference between the study implementation and the final implementation is that, in the study, the shadow awareness provided was a simple oval, rather than a true shadow. Although clearly not optimal, we chose to conduct the study with this partial prototype to rapidly garner feedback. The shadow transmission was also unidirectional: It was only transmitted from the board (with cameras) to the desk (without cameras).

The digital desk users were seated in front of the slanted desk. Users could input notes and write on them using the Wacom stylus; no cameras were used on this side. Transient ink was available to them as an option through a pie menu accessed with a right click.

Qualitative Feedback

The users were very enthusiastic about the shared workspace. They felt that it would increase the value of working sessions with team members and clients. They appreciated seeing their colleagues' input in real time. They felt it improved their collaboration, as spatial relationships were visible in real time to everyone. One designer mentioned that she preferred Distributed Outpost to the whiteboard and videoconferencing setup because Outpost digitizes the information for later use and there is no pause in the work for zooming and panning of videoconferencing cameras. They liked the flexibility of the notes and the ability to collaborate and throw out ideas quickly.

Users also liked the concept of transient ink. One designer especially liked this concept because he could show relationships between elements without committing to the interaction. Designers found that Outpost's functionality made it easy to make changes and communicate their intent to others. About half of the participants found the transient ink useful; the others did not use it during the test. As one user commented, one may as well make marks with ordinary ink and then erase them. However, one of the participants rated the transient ink as being more important in remote collaboration than voice.

Half of the users found the presence awareness shadow compelling. They felt it was vital to provide a frame of reference for the remote participant. They could thus refer to data objects with an understanding of how the remote person viewed them. They felt this gave a better understanding of participation from the remote site.

Areas for Improvement. Although the users were generally enthusiastic about the system, there were some coordination difficulties. With a physical

Post-it note, it is clear when two collocated people wish to move or edit the artifact at the same time. With Distributed Outpost, there are no restrictions on who can change or edit notes. In our study, there were times when users at both ends edited the same element because they were working in the same area of the board. However, these conflicts were infrequent and easily corrected.

Even though the remote shadow was designed to be unobtrusive, some designers found it a bit jumpy and distracting. They requested feedback with smoother motion that provided more human characteristics, such as showing hesitation and acceleration: "things that one can translate into feelings." They also wanted more detail than the oval shadow provided; we have since implemented an outline shadow that provides this detail and smoothness.

Overall, the designers we interviewed were enthused by our system and felt the concepts would be helpful in increasing the interactivity of their remote design collaboration. This study also highlighted the need for integrated audio communication; this is an important area for future research in design tools for distributed collaboration.

5. RELATED WORK

In the 1970s, Xerox PARC transformed the computing world with the graphical user interface (Johnson et al., 1989; Smith, Irby, Kimball, Verplank, & Harslem, 1982). In 1988, another sea change began at PARC. Looking beyond the desktop, Mark Weiser and colleagues embarked on research into a future of ubiquitous computing. In contrast with virtual reality, which attempts to recreate the physical world electronically, ubiquitous computing provides "embodied virtuality": It embeds the electronic world in the physical one (Weiser, 1991). This PARC research group created "computing by the inch, foot, and yard." This yielded ParcTabs (Want et al., 1995; handheld computers), ParcPads (Kantarjiev, Demers, Krivacic, Frederick, & Weiser, 1993; tablet computers), and LiveBoards (Elrod et al., 1992; electronic whiteboards), respectively. The LiveBoard introduced pen-based interaction techniques like those of a traditional whiteboard to a rear-projected electronic whiteboard. Pens are an effective method for fast, fluid, informal input. Although this vision provided the catalyst for a worldwide research effort on off-the-desktop, highly networked computing, these early research prototypes provided direct stylus input for small-, medium-, and large-scale raster-graphics displays: The fundamental method for user input was largely similar to the desktop computing standard.

5.1. Origins of Tangible Interaction

Weiser's vision of integrating computation into the physical world also encouraged researchers to begin exploring interactions that integrate the physical and electronic worlds. Three seminal research projects in this direction are the DigitalDesk (Wellner, 1993), Bricks (Fitzmaurice, Ishii, & Buxton, 1995), and Tangible Bits (Ishii & Ullmer, 1997). We discuss each of these in turn.

Interacting With Paper on the DigitalDesk

Pierre Wellner and colleagues at Rank Xerox EuroParc introduced the idea of an interface that bridges the physical and electronic worlds. Their DigitalDesk system comprises a ceiling-mounted projector that projects onto a physical desk and an electronic camera that tracks the movements of user's hands on the desk. The camera also captures objects on the desk at standard video resolution, and a second camera is zoomed in on a special area of the desk, affording higher resolution capture. With this system, users can select areas of physical documents to be copied and pasted electronically. The video capture of the physical desk can also be displayed on a remote user's desktop. As a proof-of-concept demonstration, the authors use this for a game of tic-tac-toe between remote participants. This system helped inspire our interest in this area in general and our work on The Designers' Outpost in particular. Wellner's DigitalDesk used ceiling mounted cameras to track documents and hands on a physical desktop, with a ceiling mounted projector to electronically augment the real desk (Wellner, 1993). The DoubleDigitalDesk (Wellner, 1993; Wellner & Freeman, 1993) extends this augmented paper input paradigm to a pair of networked DigitalDesks. Content can be either physical (drawn on paper by one of the users) or virtual (information that is projected, such as remote content.) The DoubleDigitalDesk enables a user to electronically view and copy her remote colleague's physical content. The Desk does not allow her to move or delete this remote content. DoubleDigitalDesk also allows for spatial content selection, but objects have no semantic distinctive identity. Outpost continues in the direction the DigitalDesk began by augmenting physical paper with electronic information. Each object and awareness cue has a distinct internal representation in Outpost. As such, this information can be edited and displayed separately. Our mediation techniques and stronger semantic representation of content enable users to delete and move remote physical content. Last, although the Desk is intended as a pair-ware system, Outpost explicitly supports multiple users at each location.

Bricks: Laying the Foundations for Graspable User Interfaces

Fitzmaurice, Ishii, and Buxton introduced the idea of physical bricks that can be used as handles for manipulating electronic content (Fitzmaurice, 1996; Fitzmaurice & Buxton, 1997; Fitzmaurice et al., 1995), offering a true, direct manipulation interface. Fitzmaurice and Buxton observed that

> today an accountant, animator and graphic designer all use the same input device set-up (i.e., a keyboard and mouse) for performing their very diverse activities. This 'universal set-up' seems inefficient for users who work in a specific domain. The mouse is a general all-purpose weak device; it can be used for many diverse tasks but may not do any one fairly well. (p. 50)

They offered the distinction between space-multiplexed input and time-multiplexed input. An audio mixing board is space-multiplexed: There is a one-to-one mapping between controls and functions. A mouse is time-multiplexed: A single input device controls all functions, and the function controlled changes over time. Bricks exhibit properties of both paradigms. It is space-multiplexed in that multiple elements can be operated in parallel, and each has their own controller; it is time-multiplexed in that the mapping between controls and functions is reconfigurable.

The bricks work included explorative evaluations of physical prototypes. A design study found a graspable interface (a stretchable box) was roughly an order of magnitude faster than the traditional MacDraw interface for positioning, rotating, and stretching rectangles. The authors implemented a bricks interface to Alias Studio, a high-end 3D modeling and animation program. The evaluation found that

> all of the approximately 20 users who have tried the interface perform parallel operations (e.g., translate and rotate) at a very early stage of using the application. Within a few minutes of using the application, users become very adept at making drawings and manipulating virtual objects. (Fitzmaurice et al., 1995, p. 447)

The experimental psychology literature (e.g., MacKenzie & Iberall, 1994) has shown the impressive dexterity of human hands in working with physical objects. Bricks leverage this dexterity; the authors' direct comparison of bricks UIs and GUIs provides empirical justification that tangible manipulation can be both faster and more intuitively bimanual than graphical manipulation. In the commercial world, applications such as Adobe Photoshop are often used in a bimanual fashion: The nondominant hand selects tools using the keyboard and the dominant hand performs precision, positional input.

Tangible Bits

In 1997, Ishii and Ullmer introduced their Tangible Bits research agenda (Ishii & Ullmer, 1997). This article and the group's substantial subsequent work (e.g., Brave, Ishii, & Dahley, 1998; Jacob, Ishii, Pangaro, & Patten, 2002; Ullmer & Ishii, 1997, 2001; Underkoffler & Ishii, 1999; Underkoffler, Ullmer, & Ishii, 1999) offer a broad array of techniques for physically interacting with the digital world and was one of the inspirations for this research. Their work includes physical interaction techniques for both foreground (graspable) and background (ambient) interaction. Most related to Outpost is metaDESK (Ullmer & Ishii, 1997), a digital desk employing physical objects as the controls for an interactive map of the Massachusetts Institute of Technology campus projected onto the desk. Outpost employs the metaDESK insight of tracking objects with a camera mounted inside the display case.

5.2. Walls for Collaborative Design

Bellotti and Rogers (1997) conducted a study on Web publishing workflow. They too discovered a tension between paper-based practices and electronic practices. In particular, they found that people were often more comfortable working on paper but felt that electronic tools were beneficial for stronger communication and awareness among distributed teams. One site director commented,

> What I would love would be a flat panel I could hang on a wall. … For the tacked up paper and string setup we have, a video wall could be really useful, not just for the sake of more expensive equipment, but for working with remote group members, for ease of modification, and for keeping a better record of the evolution of the site. (Bellotti & Rogers, 1997, p. 284)

This study helped motivate our interest in combining the physical and electronic worlds to gain these benefits.

Outpost is a member of the spatial class of tangible interfaces (Klemmer, 2004; Ullmer & Ishii, 2001). In spatial applications, users collaboratively create and interact with information in a Cartesian plane. These applications include augmented walls, whiteboards, and tables (Jacob et al., 2002; Klemmer et al., 2001; McGee, Cohen, Wesson, & Horman, 2002; Moran et al., 1999; Rekimoto & Saitoh, 1999; Ullmer & Ishii, 1997; Underkoffler & Ishii, 1999; Underkoffler, Ullmer, & Ishii, 1999; Wellner, 1993). A majority of these applications use computer vision, often in conjunction with image capture.

Collaborage (Moran et al., 1999), a spatial tangible user interface (TUI), uses computer vision to capture paper information arranged on an ordinary wall, enabling it to be electronically accessed. These pieces of paper are tagged with glyphs (Hecht, 1994), a type of 2D barcode. The electronic capture of paper information enables remote viewing (e.g., a Web page view of a physical in–out board) but not remote interaction. Collaborage's computer vision capture of paper on walls inspired our work on The Designers' Outpost. One drawback of the Collaborage capture system is its long latency (8 to 10 sec on average). Outpost improves on this, with a location recognition latency of approximately 200 msec and an average capture latency of approximately 1.5 sec. This improvement is primarily because of our two-camera hardware approach and software built on top of OpenCV, a highly optimized vision toolkit (Bradski, 2001). Outpost also incorporates other forms of input using styli and physical tools.

McGee et al.'s Rasa system extends the physical wall-mounted map and Post-it note tools currently used in military command post settings with a touch-sensitive smart Board behind the map, gesture recognition on ink strokes written on the Post-it notes, and speech recognition on verbal commands (Cohen & McGee, 2004; McGee, Cohen, & Wu, 2000). They performed a comparative evaluation of Rasa and traditional paper tools with six users (McGee et al., 2002). To ascertain Rasa's robustness, the authors shut off the computer halfway through the study without informing the users. They found that users were able to keep working through a power failure and, with minimal effort, return the system to a consistent state after power was restored, showing "that by combining paper and digital tools, we have constructed a hybrid system that supports the continuation of work in spite of power, communications, and hardware or software failures" (McGee, Cohen, Wesson, & Horman, 2002).

However, the domain of technology support for military activity hides several shortcomings with recognition- and transformation-laden interfaces. These systems leverage existing practices of a precise, consistent grammar, and this highly constrained vocabulary and notation greatly eases recognition. As such, tool support for military activity has little bearing on the much more fluid and ad hoc practices of most professional and domestic life. Over the next decade, recognition will play a larger role in human–computer interaction, and our group's research agenda of informal interfaces (Landay & Myers, 2001) is part of this trend. Although informal interfaces incorporate more recognition technologies (e.g., gesture recognition, or vision recognition of Post-its) than WIMP interfaces, good informal interface designers work to minimize the recognition and, most important, minimize the transformation of users' input.

Technology Transfer of Spatial TUIs

Several spatial TUIs have received interest from industry. Adapx is a start-up based on the Rasa system. The CamFire product by SMART Technologies (D. Martin; http://smarttech. com) is a high-resolution digital camera for whiteboard capture. CamFire is based on ZombieBoard (Saund, 1998, 1999) and contains structured capture techniques inspired by The Designers' Outpost. As part of the capture process, CamFire uses computer vision techniques to perspective correct the capture and clean up the image. The addition inspired by Outpost is that the capture is segmented so that Post-it notes and other objects are semantically abstracted and become objects in their Smart Ideas electronic diagramming tool. The POLE project in Switzerland has used this system for urban planning, and discussed their use of it in a research paper (Breit, Kündig, & Häubi, 2004).

Electronic Walls

In addition to wall-scale tangible interfaces, several researchers have developed electronic wall-scale interfaces. The i-LAND system introduced interaction techniques for working with display surfaces embedded in rooms and furniture (Streitz et al., 1999). As part of their work, they developed the DynaWall, three adjacent electronic whiteboards that take input via hand gestures. The PostBrainstorm system introduced interaction techniques for large, tiled projector surfaces (Guimbretière, Stone, & Winograd, 2001). Rekimoto and Saitoh (1999) developed a system to integrate laptop computers, projected surfaces, and tagged physical objects. Other researchers have investigated interaction techniques for large electronic display surfaces and multimodal interaction with paper (McGee et al., 2002; McGee et al., 2000). This body of work on wall-scale interfaces motivates the concept that, for many tasks (especially in collaborative situations), manipulating physical objects on large surfaces is an intuitive and effective means of performing computer input.

5.3. Design History

Much of our thinking about design history is motivated by Design Rationale: Concepts, Techniques, and Use (Moran & Carroll, 1996). The 16 contributed chapters characterize the primary goal of design as giving shape to artifacts—design products—yet underscoring that "the artifact is a concrete form that does not (except in very subtle ways) manifest this process of creation."

A number of design rationale systems have been proposed in the past, such as the seminal ibis (Rittel & Webber, 1973) system in the 1970s and, more recently, QOC (MacLean, Young, Bellotti, & Moran, 1991). These systems employ semiformal syntaxes to capture design rationale in the form of argumentation surrounding decisions made during a design process. These systems have not achieved widespread use with designers, possibly because they impose a rigid structure on design thinking and burden designers with creating and maintaining a separate rationale representation in parallel with the design itself.

History-Enabled Applications

The visual design of our global timeline is inspired by the Chimera graphical editor (Kurlander & Feiner, 1992), which introduced the comic strip metaphor for displaying a history of changes. Chimera also used highlighting to focus on the parts that changed in each frame. Our time line extends the design in Chimera with numerous ways of filtering the displayed frames and with display of branched history.

The Time-Machine Computing (TMC) browser (Rekimoto, 1999) replaces a standard desktop browser, offering time as a unifying method for storing personal information such as documents, digital photographs, notes, and calendar information. Like the Outpost history system, it affords time-based browsing. In addition, like our local history system, the TMC browser allows users to select an object and return to the time it was created, a type of direct-manipulation query. An interesting personal information management feature in the TMC browser is that it enables users to place objects in a future point in time. The major difference between TMC and our system is that our system is intended as a collaborative design aid, whereas TMC is intended for personal information management. Outpost also offers support for branched history, which TMC does not. However, there are some techniques in TMC that might be beneficial to our system, such as a calendar view of information.

The WeMet system (Rhyne & Wolf, 1992) for distributed, collaborative drawing also automatically captures history. According to the authors, this allows users to "reconstruct the present" and allows parties that join work in progress to review the work performed by the other participants so far. WeMet inspired us to include explicit bookmarks in the history.

The VKB system (Shipman & Hseih, 2000) introduces the notion of "constructive time," which is the reader's experience of accessing a history in a hypertext. Our third scenario, finding the rationale behind a design decision, draws inspiration from the VKB notion of history being created for the benefit of an external viewer. Our By Meeting filter is implemented in a similar fashion to the VKB meeting discretization.

Capture and Access

Our focus on informal interaction (Landay & Myers, 2001) recognizes that forcing designers to heavily structure an artifact too early can distract from the main task of ideation; this led us to draw from related research on informal capture systems. Informal capture systems attempt to collect information from users in natural ways, that is, information that they produce in the normal course of their activities and attempt to structure it in useful ways for later retrieval.

The Classroom 2000 project (Abowd, 1999; Brotherton, 2001) captures information from multiple sources including audio and video of classroom lecturers, ink from students' notes and annotations, and lecturers' presentations slides. This information is then merged and indexed in order to support students' task of reviewing lecture notes.

The AudioNotebook (Stifelman, Arons, & Schmandt, 2001) and the Dynomite system (Wilcox, Schilit, & Sawhney, 1997) both focus on personal information capture and provide interaction techniques for browsing histories. Although they are concerned with audio, their methods for creating inferred bookmarks inspired us. AudioNotebook creates bookmarks based on pauses and changes in pitch, inspiring us to add inferred filters in our history system. Dynomite's ink properties inspired us to add author information to created notes. Outpost differs from these systems in that it is designed to support collaborative, rather than individual, practices. The CORAL system (Moran et al., 1997) captures and coordinates data from multiple sources such as audio and whiteboard notes to support meeting capture. These four systems share with Classroom 2000 a task-oriented focus on visualizations to support later retrieval.

5.4. Remote Collaboration

Our remote collaboration research draws on earlier work in media spaces for remote interaction. We now discuss this work and also look forward to the possibilities for remote actuation.

Distributed Media Spaces

In the mid-1970s, Krueger's videoplace art installation introduced the use of computer vision to track a user's shadow (Krueger, 1977, 1983). Computer feedback was generated based on the user's movement in an art space. The vision system was based on chroma key technology developed for broadcast news. Although the gallery was constrained to have a solid background that enabled the chroma key technique to work, this system was decades ahead of its time.

Over the last decade, there has been compelling research in distributed media spaces for visual collaboration tasks, such as shared drawing through electronic whiteboards. These researchers found, as we have, that users are interested in collaborating on design artifacts from different places. Clearboard (Ishii et al., 1994) and VideoWhiteboard (Tang & Minneman, 1991) are pair-ware systems that integrate visual drawings with video presence on a single display. Clearboard users draw on a glass board. The board is augmented with a live video projection, giving the appearance of "looking through the glass" at the remote participant's drawing, face, and upper body. The glass board and video camera setup is duplicated at each end. VideoWhiteboard works in a similar fashion, except that the video image is the shadow of a standing remote user's body. Both of these systems use a direct video feed, cleverly aligned, to transmit both the drawing and presence information.

Although the data transmission in these systems is a raw video feed, Distributed Outpost has a structured representation of the content. Its computer vision algorithms locate physical objects and users' shadows, building an internal representation of this content and awareness feedback. This semantic understanding of the information allows for more flexibility in presentation. For example, in Distributed Outpost the awareness display can be removed, modified in color, or shown as an outline only. Distributed Outpost provides more control over content changes, allowing objects to be erased or moved without affecting the rest of the display. In addition, all of the advantages carry over to Outpost's design history and ability to transition to other tools.

Our research goal is to bring together TUIs and distributed media spaces to create and evaluate an application that supports an existing design practice.

Remote Actuation

Our work with Distributed Outpost offers tangible input locally and electronic output remotely. An alternate approach is to use actuation for physically controlling a remote space. As these technologies mature, they may become feasible for design tools like Outpost.

InTouch (Brave et al., 1998) provides an identical set of cylindrical rollers to participants at two different locations. The networked rollers behave as though they are physically connected. This system provided a shared mechanism for synchronous awareness of touch. InTouch's compelling aesthetic experience encouraged us to explore richer awareness mechanisms for our design tool.

Reznik and Canny's Universal Planar Manipulator (Reznik, 2000) provides a view of the future where physical objects can be controlled remotely. The Universal Planar Manipulator is a rigid, horizontal plate, which vibrates in its own plane and moves generic objects placed on it because of friction. However, the technology is not yet mature enough to support large numbers of objects, and our system is based on vertical as opposed to horizontal surfaces.

6. CONCLUSIONS

We have presented The Designers' Outpost, a system that integrates physical and digital interactions for collaborative Web site information design. Its functions are informed by observations of real Web site design practice, providing many of the affordances of current paper-based practice while offering the advantages of electronic media. In particular, structured digital capture of physical walls opens up the possibility for time-shifting and space-shifting content. This insight enabled us to provide support for design history and remote collaboration.

At an infrastructure level, the Outpost vision system yields an interactive-rate solution for robustly finding notes on a large surface. These results and those of other researchers show that computer vision is an effective technology for informal, collaborative interaction with physical media on walls. Outpost's informal history mechanism comprises three novel history visualizations for collaborative early-phase design: a stub-branching main time line, an in situ object time line, and an annotated synopsis view. Distributed Designers' Outpost provides a shared workspace where the participants can edit any object, regardless of where it was created. We presented two novel awareness mechanisms: transient ink input for gestures and a vision-tracked stylized shadow for presence.

Over the 3 years of the project, we iteratively evaluated our design with 27 professional designers, showing that electronic whiteboards should be calm and that there is substantial merit in a system that is simultaneously tangible and virtual. The designers were enthusiastic about using Outpost. In particular, the most valuable aspects that Outpost's integrated interactions enabled were the fluid transition from artifacts on walls to single-user tools such as DENIM, lightweight support for design history, and its potential to support their current practices and increase their ability to work in distributed teams. The iterative evaluation also helped us flesh out design patterns for interactive walls. They consistently underscored the importance of a calm user experience, steering us away from garish interactions like constantly updating history thumbnails.

The Outpost research presents an instance of the larger paradigm of integrating physical and digital interactions. Traditional human–computer interaction focuses on purely digital interactions; the additional design variable introduced by this new class of system is the question of what aspects of the system should be represented physically and how. With Outpost, we chose for the initial content creation to be a physical process with Post-it notes, and for the relationships between the content—such as links—to be specified digitally. What this partitioning enabled was for the content interface to be "natural" in that it employs the same physical artifacts that designers have long used and that, because they are physical, the "interaction techniques" of creation, composition, and manipulation are readily apparent: The elements behave like everyday objects because they are everyday objects. An important property of everyday objects in this context that was underscored by the studies is that having multiple physical objects readily enables multiple-person interaction. Surprisingly to us, the physicality of notes also enabled fluid movement between a personal space away from the board and the collaborative space on the board surface. The advantage of having content relationships be digital—to put it more technically than our users might—is that as the representation of relationships is defined by constraints, such as the endpoints of a link, their digital nature enables them to be computationally updated as users manipulate the content that specifies the constraint values, such as moving a note.

Looking forward, this research area could benefit from integrating informal audio capture and access. We have found that in brainstorming sessions, the discussion among designers often captures information not expressed in the resulting visual artifact.

Although the Outpost user interface is highly effective, developing this system was difficult and time-consuming. This development required us to learn a substantial amount about computer vision algorithms and design software architectures for tangible interaction. At the time we developed this system, tools support for this development process was minimal. Although libraries such as OpenCV provide efficient implementations of image processing techniques, it is still the developer's responsibility to (a) understand the behavior of these techniques, (b) string together low-level image processing primitives to achieve the high-level user interface goal, and (c) create a software architecture for communication between the vision input system and the UI.

Through this evolution and extension of the Outpost code base, we gained an appreciation for UI tools and became aware of their absence for developing TUI input. Where appropriate, the design history support has been added to the SATIN toolkit rather than the Outpost application. This also helped motivate our interest in toolkits in general and Papier-Mâché in particular, and subsequent research on the Papier-Mâché toolkit (Klemmer et al., 2004) addressed these many of these development difficulties. Specifically, the difficulties involved in extending the vision tracking system in Outpost encouraged us toward the more flexible architecture that Papier-Mâché provides. For example, with Papier-Mâché, extending a vision system for capturing shadows is only a few lines of code.

Many professional practices feature the creation of an artifact by several individuals over an extended period. We hope that this work on exploring interfaces for collaborative wall-scale design will inspire work in other professional domains as well. Computers have been instrumental in allowing us to communicate quickly with people all over the world. However, we lose some of the advantages of meeting face to face. The Designers' Outpost begins to bridge the gap between the virtual and physical worlds and help both collocated and remote teams work more comfortably and effectively.

NOTES

Acknowledgements. We thank Mark Newman, Michael Thomsen, Mark Bilezikjian, Ryan Farrell, Robert Lee, and Ethan Phelps-Goodman for their insightful discussions and implementation assistance; we also thank Erin Milnes for copyediting assistance.

Support. This work was supported by National Science Foundation Grant IIS-0084367 and hardware donations from SMART Technologies and Intel.

Authors' Present Addresses. Scott R. Klemmer, Stanford University, Gates Computer Science 3B-384, Stanford CA 94305–9035. E-mail: srk@cs.stanford.edu. Katherine M. Everitt, University of Washington, Paul G. Allen Center, Box 352350, Seattle, WA 98195–2350. E-mail: everitt@cs.washington.edu. James A. Landay, University of Washington, Paul G. Allen Center, Box 352350, Seattle, WA 98195–2350.

HCI Editorial Record. First manuscript received September 6, 2005. Revision received July 21, 2006. Accepted by Gregory Abowd. Final manuscript received February 11, 2008. *— Editor*

REFERENCES

Abowd, G. D. (1999). Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM Systems Journal, 38*(4), 508–530.

Arias, E., Eden, H., Fischer, G., Gorman, A., & Scharff, E. (2000). Transcending the individual human mind—Creating shared understanding through collaborative design. *ACM Transactions on Computer-Human Interaction, 7*(1), 84–113.

Bellotti, V., & Rogers, Y. (1997). From Web press to Web pressure: Multimedia representations and multimedia publishing. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(4), 509–522.

Berlage, T., & Genau, A. (1993). A framework for shared applications with a replicated architecture. *Proceedings of UIST: User Interface Software and Technology.* New York: ACM Press.

Beyer, H., & Holtzblatt, K. (1997). *Contextual design: Defining customer-centered systems.* San Francisco: Morgan Kaufmann.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* New York: Oxford University Press.

Bradski, G. (2001). *Open Source Computer Vision Library.* Santa Clara, CA: Intel. Retrieved from http://www.intel.com/technology/computing/opencv/index.htm

Brave, S., Ishii, H., & Dahley, A. (1998). Tangible interfaces for remote collaboration and communication. *Proceedings of CSCW: Computer Supported Cooperative Work.*

Breit, M., Kündig, D., & Häubi, F. (2004). Project oriented learning environment (POLE-Europe). *Proceedings of International Conference on Computing in Civil and Building Engineering.* New York: ACM Press.

Brotherton, J. A. (2001). *Enriching everyday experiences through the automated capture and access of live experiences: eClass: Building, observing and understanding the impact of capture and access in an educational domain.* Unpublished doctoral dissertation, Georgia Institute of Technology, Atlanta.

Brown, J. S., & Duguid, P. (2002). Reading the background. In *The social life of information* (pp. 173–205). Cambridge, MA: Harvard Business School Press.

Cohen, P. R., & McGee, D. R. (2004). Tangible multimodal interfaces for safety critical applications. *Communications of the ACM, 47,* 41–46.

Covi, L. M., Olsen, J. S., Rocco, E., Miller, W. J., & Allie, P. (1998). A room of your own: What do we learn about support of teamwork from assessing teams in dedicated project rooms. *Proceedings of Cooperative Buildings: Integrating Information, Organization and Architecture: First International Workshop, Co'Build '98.* New York: ACM Press.

Csikszentmihalyi, M. (1990). *Flow: The psychology of optimal experience.* New York: Harper & Row.

Edwards, W. K., & Mynatt, E. D. (1997). Timewarp: Techniques for autonomous collaboration. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., et al. (1992). Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Everitt, K. M., Klemmer, S. R., Lee, R., & Landay, J. A. (2003). Two worlds apart: Bridging the gap between physical and virtual media for distributed design collaboration. *CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Fitzmaurice, G. W. (1996). *Graspable user interfaces.* Unpublished doctoral dissertation, University of Toronto, Toronto, Canada.

Fitzmaurice, G. W., & Buxton, W. (1997). An empirical evaluation of Graspable User Interfaces: towards specialized, space-multiplexed input. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Fitzmaurice, G. W., Ishii, H., & Buxton, W. (1995). Bricks: Laying the foundations for graspable user interfaces. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Forsyth, D. A., & Ponce, J. (2003). *Computer vision: A modern approach.* Upper Saddle River, NJ: Prentice Hall.

Freeman, W. T., Miyake, Y., Tanaka, K.-i., Anderson, D. B., Beardsley, P. A., Dodge, C. N., et al. (1998). Computer vision for interactive computer graphics. *IEEE Computer Graphics and Applications, 18*(3), 42–53.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software.* Reading, MA: Addison-Wesley.

Guimbretière, F., Stone, M., & Winograd, T. (2001). Fluid interaction with high-resolution wall-size displays. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Hecht, D. L. (1994). Embedded data glyph technology for hardcopy digital documents. *Proceedings of Color Hard Copy and Graphic Arts III.* New York: ACM Press.

Hong, J. I., & Landay, J. A. (2000). SATIN: A toolkit for informal ink-based applications. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Hymes, C. M., & Olson, G. M. (1997). Quick but not so dirty web design: Applying empirical conceptual clustering techniques to organise hypertext content. *Proceedings of DIS: Symposium on Designing Interactive Systems.* New York: ACM Press.

Ishii, H., Kobayashi, M., & Arita, K. (1994). Iterative design of seamless collaboration media. *Communications of the ACM, 37,* 83–97. New York: ACM Press.

Ishii, H., & Ullmer, B. (1997). Tangible bits: Towards seamless interfaces between people, bits and atoms. *Proceedings of CHI: Human Factors in Computing Systems.*

Jacob, R., Ishii, H., Pangaro, G., & Patten, J. (2002). A tangible interface for organizing information using a grid. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Johnson, J., Roberts, T. L., Verplank, W., Smith, D. C., Irby, C., Beard, M., et al. (1989). The Xerox Star: A retrospective. *IEEE Computer, 22*(9), 11–28.

Kaasten, S., & Greenberg, S. (2001). Integrating back, history and bookmarks in Web browsers. *Proceedings of Extended Abstracts of CHI: Conference on Human Factors in Computing Systems.* New York: ACM Press.

Kantarjiev, C. K., Demers, A., Krivacic, R. T., Frederick, R., & Weiser, M. (1993). Experiences with X in a wireless environment. *Proceedings of USENIX Symposium on Mobile & Location-independent Computing.* New York: ACM Press.

Kelley, J. F. (1984). An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Office Information Systems, 2*(1), 26–41.

Kim, J., Seitz, S., & Agrawala, M. (2004). Video-based document tracking: Unifying your physical and electronic desktops. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Klemmer, S. R. (2004). *Tangible user interface input: Tools and techniques.* Unpublished doctoral dissertation, University of California, Berkeley.

Klemmer, S. R., Li, J., Lin, J., & Landay, J. A. (2004). Papier-Mâché: Toolkit support for tangible input. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M., & Landay, J. A. (2001). The designers' outpost: A tangible interface for collaborative Web site design. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Klemmer, S. R., Sinha, A. K., Chen, J., Landay, J. A., Aboobaker, N., & Wang, A. (2000). SUEDE: A Wizard of Oz prototyping tool for speech user interfaces. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Klemmer, S. R., Thomsen, M., Phelps-Goodman, E., Lee, R., & Landay, J. A. (2002). Where do Web sites come from? Capturing and interacting with design history.

Proceedings of CHI: ACM Conference on Human Factors in Computing Systems. New York: ACM Press.

Krueger, M. W. (1977). Responsive environments. *Proceedings of AFIPS National Computer Conference.* New York: ACM Press.

Krueger, M. W. (1983). *Artificial reality.* Reading, MA: Addison-Wesley.

Kurlander, D., & Feiner, S. (1992). A history-based macro by example system. *Proceedings of UIST: User Interface Software and Technology.* New York: ACM Press.

Landay, J. A. (1996). *Interactive sketching for the early stages of user interface design.* Unpublished doctoral dissertation, Carnegie Mellon University, Pittsburgh, PA.

Landay, J. A., & Myers, B. A. (1995). Interactive sketching for the early stages of user interface design. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Landay, J. A., & Myers, B. A. (2001). Sketching interfaces: Toward more human interface design. *IEEE Computer, 34*(3), 56–64.

Lin, J. (2003). Damask: A tool for early-stage design and prototyping of cross-device user interfaces. *Proceedings of Conference Supplement to UIST 2003: ACM Symposium on User Interface Software and Technology: Doctoral Consortium.* New York: ACM Press.

Lin, J., & Landay, J. A. (2002). Damask: A tool for early-stage design and prototyping of multi-device user interfaces. *Proceedings of 8th International Conference on Distributed Multimedia Systems (2002 International Workshop on Visual Computing).* New York: ACM Press.

Lin, J., Newman, M. W., Hong, J. I., & Landay, J. A. (2000). DENIM: Finding a tighter fit between tools and practice for web site design. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

MacKenzie, C. L., & Iberall, T. (1994). *The grasping hand* (Vol. 104). Amsterdam: Elsevier Science B.V.

MacLean, A., Young, R. M., Bellotti, V., & Moran, T. P. (1991). Questions, options, and criteria: elements of design space analysis. *Human-Computer Interaction, 6*(3–4), 201–250.

MacLean, A., Young, R. M., & Moran, T. P. (1989). Design rationale: the argument behind the artifact. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Maulsby, D., Greenberg, S., & Mander, R. (1993). Prototyping an intelligent agent through Wizard of Oz. *Proceedings of INTERCHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

McGee, D. R., Cohen, P. R., Wesson, R. M., & Horman, S. (2002). Comparing paper and tangible, multimodal tools. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

McGee, D. R., Cohen, P. R., & Wu, L. (2000). Something from nothing: Augmenting a paper-based work practice via multimodal interaction. *Proceedings of Designing Augmented Reality Environments.* New York: ACM Press.

Moran, T. P., & Carroll, J. M. (Eds.). (1996). *Design rationale: Concepts, techniques, and use.* Mahwah, NJ: Lawrence Erlbaum.

Moran, T. P., Chiu, P., van Melle, W., & Kurtenbach, G. (1995). Implicit structures for pen-based systems within a freeform interaction paradigm. In *CHI: Human Factors in Computing System*s (Vol. 1, pp. 487–494). New York: ACM Press.

Moran, T. P., Palen, L., Harrison, S., Chiu, P., Kimber, D., Minneman, S., et al. (1997). “I’ll get that off the audio”: A case study of salvaging multimedia records. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Moran, T. P., Saund, E., van Melle, W., Gujar, A., Fishkin, K. P., & Harrison, B. L. (1999). Design and technology for collaborage: Collaborative collages of information on physical walls. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Myers, B. A., & Kosbie, D. S. (1996). Reusable hierarchical command objects. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Newman, M. W., & Landay, J. A. (2000). Sitemaps, storyboards, and specifications: A sketch of Web site design practice. *Proceedings of DIS: Symposium on Designing Interactive Systems.* New York: ACM Press.

Newman, M. W., Lin, J., Hong, J. I., & Landay, J. A. (2003). DENIM: An informal Web site design tool inspired by observations of practice. *Human–Computer Interaction, 18*(3), 259–324.

Nielsen, J. (1998). *Test it!* Jakob Nielsen’s UI Tips. C|net’s Builder.com. Retrieved from http://www.builder.com/Graphics/UserInterface/ss01a.html

Norman, D. A. (1998). *The invisible computer: Why good products can fail, the personal computer is so complex, and information appliances are the solution.* Cambridge, MA: MIT Press.

Pedersen, E. R. (1998). People presence or room activity supporting peripheral awareness over distance. *Proceedings of Extended Abstracts of CHI: Conference on Human Factors in Computing Systems.* New York: ACM Press.

Pedersen, E. R., & Sokoler, T. (1997). AROMA: Abstract representation of presence supporting mutual awareness. *Proceedings of CHI: Conference on Human Factors in Computing Systems.* New York: ACM Press.

Rekimoto, J. (1999). Time-machine computing: A time-centric approach for the information environment. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Rekimoto, J., & Saitoh, M. (1999). Augmented surfaces: a spatially continuous work space for hybrid computing environments. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Rettig, M. (2000, Summer). *Walls beyond whiteboards.* Artifact from the third Advance for Design summit, Telluride, CO. Retrieved from http://www.marcrettig.com/writings/rettig.walls.72dpi.pdf

Reznik, D. S. (2000). *The universal planar manipulator.* Unpublished doctoral dissertation, University of California Berkeley, Berkeley.

Rhyne, J. R., & Wolf, C. G. (1992). Tools for supporting the collaborative process. *Proceedings of UIST: User Interface Software and Technology.* New York: ACM Press.

Richter, H., Abowd, G. D., Geyer, W., Fuchs, L., Daijavad, S., & Poltrock, S. (2001). Integrating meeting capture within a collaborative team environment. *Proceedings of Ubicomp 2001.* New York: ACM Press.

Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences, 4,* 155–169.

Saund, E. (1998). *Image mosaicing and a diagrammatic user interface for an office whiteboard scanner* (Tech. Rep.). Palo Alto, CA: Xerox Palo Alto Research Center.

Saund, E. (1999). Bringing the marks on a whiteboard to electronic life. *Proceedings of CoBuild’99: Second International Workshop on Cooperative Buildings.* New York: ACM Press.

Shipman, F. M., & Hseih, H. (2000). Navigable history: A reader's view of writer's time. The *New Review of Hypermedia and Multimedia, 6*(1), 147–167.

Sinha, A. K., Klemmer, S. R., & Landay, J. A. (2002). Embarking on spoken-language NL interface design. *The International Journal of Speech Technology, 5*(2), 159–169.

Sinha, A. K., & Landay, J. A. (2003). Capturing user tests in a multimodal, multidevice informal prototyping tool. *Proceedings of ICMI-PUI: ACM International Conference on Multimodal Interfaces.* New York: ACM Press.

Smith, D. C., Irby, C., Kimball, R., Verplank, B., & Harslem, E. (1982). Designing the star user interface. *Byte, 7,* 242–282.

Snibbe, S. S., MacLean, K. E., Shaw, R., Roderick, J., Verplank, W. L., & Scheeff, M. (2001). Haptic techniques for media control. *Proceedings of UIST: ACM Symposium on User Interface Software and Technology.* New York: ACM Press.

Stallman, R. (1993). *GNU Emacs: The extensible self-documenting text editor.* Cambridge, MA: Free Software Foundation.

Stifelman, L., Arons, B., & Schmandt, C. (2001). The audio notebook: Paper and pen interaction with structured speech. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Streitz, N. A., Geißler, J., Holmer, T., Konomi, S. i., Müller-Tomfelde, C., Reischl, W., et al. (1999). i-LAND: An interactive landscape for creativity and innovation. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Tang, J. C., & Minneman, S. L. (1991). VideoWhiteboard: Video shadows to support remote collaboration. *Proceedings of CHI: Human Factors in Computing Systems.* New York: ACM Press.

Terry, M., & Mynatt, E. D. (2002). Supporting experimentation with side views. *Communications of the ACM, 45*(10), 106–108.

Terry, M., Mynatt, E. D., Nakakoji, K., & Yamamoto, Y. (2004). Variation in element and action: Supporting simultaneous development of alternative solutions. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Torok, G. P., & White, A. B. (1982). *U.S. Patent No. 4,317,956.* Washington, DC: U.S. Patent and Trademark Office.

Ullmer, B., & Ishii, H. (1997). The metaDESK: Models and prototypes for tangible user interfaces. *Proceedings of UIST: User Interface Software and Technology,* 223–232.

Ullmer, B., & Ishii, H. (2001). Emerging frameworks for tangible user interfaces. In J. M. Carroll (Ed.), *Human–computer interaction in the new millennium* (pp. 579–601). Reading, MA: Addison-Wesley.

Ulrich, I., & Nourbakhsh, I. (2000). Firewire untethered: High-quality images for notebook computers. *Advanced Imaging Magazine,* January, 69–70.

Underkoffler, J., & Ishii, H. (1999). Urp: A luminous-tangible workbench for urban planning and design. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.

Underkoffler, J., Ullmer, B., & Ishii, H. (1999). Emancipated pixels: Real-world graphics in the luminous room. *Proceedings of SIGGRAPH: Computer Graphics and Interactive Techniques.* New York: ACM Press.

Want, R., Schilit, B. N., Adams, N. I., Gold, R., Petersen, K., Goldberg, D., et al. (1995). An overview of the PARCTAB Ubiquitous Computing Experiment. *IEEE Personal Communications, 2*(6), 28–43.

Weiser, M. (1991). The computer for the 21st century. *Scientific American, 265,* 94–104.

Weiser, M., & Brown, J. S. (1997). The coming age of calm technology. In P. J. Denning & R. M. Metcalfe (Eds.), *Beyond calculation: The next fifty years of computing* (pp. 75–85). New York: Copernicus.

Wellner, P. (1993). Interacting with paper on the Digitaldesk. *Communications of the ACM, 36,* 87–96.

Wellner, P., & Freeman, S. (1993). *The DoubleDigitalDesk: Shared editing of paper documents* (No. EPC-93-108). Cambridge, UK: EuroPARC.

Whittaker, S., & Schwarz, H. (1995). Back to the future: Pen and paper technology supports complex group coordination. In *CHI: Human Factors in Computing Systems* (Vol. 1, pp. 495–502). New York: ACM Press.

Wilcox, L. D., Schilit, B. N., & Sawhney, N. (1997). Dynomite: A dynamically organized ink and audio notebook. *Proceedings of CHI: ACM Conference on Human Factors in Computing Systems.* New York: ACM Press.