

DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice

Mark W. Newman
Palo Alto Research Center

James Lin, Jason I. Hong, and James A. Landay
University of California, Berkeley

ABSTRACT

Through a study of Web site design practice, we observed that designers employ multiple representations of Web sites as they progress through the design process and that these representations allow them to focus on different aspects of the design. In particular, we observed that Web site designers focus their design efforts at 3 dif-

Mark Newman is a computer scientist with interests in human–computer interaction and ubiquitous computing; he is a researcher in the Computer Science Lab of the Palo Alto Research Center. **James Lin** is a computer scientist with interests in human–computer interaction and end-user programming; he is a graduate student in the Electrical Engineering and Computer Sciences Department of the University of California at Berkeley. **Jason Hong** is a computer scientist with interests in human–computer interaction and ubiquitous computing; he is a graduate student in the Electrical Engineering and Computer Sciences Department of the University of California at Berkeley. **James Landay** is a computer scientist with interests in human–computer interaction, design and evaluation tools, end-user programming, and ubiquitous computing; he is an Associate Professor in the Electrical Engineering and Computer Sciences Department of the University of California at Berkeley.

CONTENTS**1. INTRODUCTION****2. INVESTIGATION INTO WEB SITE DESIGN**

- 2.1. Description of the Study
 - What We Asked
 - Validation of Our Observations
- 2.2. Specialization Within Web Design
- 2.3. The Story of a Design: A Software Tutorial
- 2.4. Design Process
 - Discovery
 - Design Exploration
 - Design Refinement
 - Production
 - Design Process Timeline
 - Individual and Collaborative Activity
- 2.5. Products of Design Practice
 - Site Maps
 - Storyboards
 - Schematics
 - Mock-Ups
 - Prototypes
 - Specifications and Guidelines
 - Presentations
 - Written Documents
- 2.6. Tools of Design Practice
 - Sketching on Paper
 - Design War Rooms
 - Computer-Based Tools
- 2.7. Implications for Web Design Tools
 - Use an Informal UI
 - Focus on Intermediate Artifacts
 - Support Multiple Representations
 - Integrate With Other Tools
 - Manage History and Variations
 - Integrate With Paper

3. DENIM: AN INFORMAL WEB SITE DESIGN TOOL

- 3.1. The DENIM UI
 - Unifying Representations Through Zooming
 - Gestures and Pie Menus
 - Creating Pages
 - Adding and Editing Web Page Content
 - Replacing Handwritten Text With Typed Text
 - Creating Arrows and Hyperlinks
 - Interacting With Designs in Run Mode
 - Exporting to HTML
- 3.2. Implementation Details
 - SATIN Overview

- Interpreting Ink Strokes in DENIM
 - Rendering and Semantic Zooming in DENIM
 - Executing Designs in Run Mode
 - 3.3. Evaluation and Usage Experience
 - Description of Study
 - Observations
 - Feedback
 - Other Usage Experience
 - 3.4. Discussion
 - 3.5. Current and Future Work
 - 4. RELATED WORK**
 - 5. CONCLUSION**
-

ferent levels of granularity—site map, storyboard, and individual page—and that designers sketch at all levels during the early stages of design. Sketching on paper is especially important during the early phases of a project, when designers wish to explore many design possibilities quickly without focusing on low-level details. Existing Web design tools do not support such exploration tasks well, nor do they adequately integrate multiple site representations. Informed by these observations we developed DENIM: an *informal* Web site design tool that supports early phase information and navigation design of Web sites. It supports sketching input, allows design at different levels of granularity, and unifies the levels through zooming. Designers are able to interact with their sketched designs as if in a Web browser, thus allowing rapid creation and exploration of interactive prototypes. Based on an evaluation with professional designers as well as usage feedback from users who have downloaded DENIM from the Internet, we have made numerous improvements to the system and have received many positive reactions from designers who would like to use a system like DENIM in their work.

1. INTRODUCTION

It is commonly known that designers of all types make use of paper-and-pencil sketches when designing a new product, building, machine, computer program, or advertising campaign. These sketches help designers think broadly during the early phases of conceptualizing a new artifact and keep designers and anyone with whom they share their early ideas from focusing on details and formalities, which are not yet relevant (Strothotte, Preim, Raab, Schumann, & Forsey, 1994). When we became interested in developing a new tool for Web designers, we wanted to see if there was still the need for informality and sketchiness in the early stages of Web design or if existing Web design applications and the pressures of “Internet time” eliminated

that need. So we took a fresh look at Web site design practice to determine what kinds of tools would be helpful to support designers. In this article we describe our observations of Web site design practice and introduce a system named DENIM that is aimed at supporting the early phases of the Web site design process.

A starting point for this project was the hypothesis that informal tools would fit well with designers' practices. By "informal," we mean tools whose user interfaces (UIs) are designed to support natural, ambiguous forms of human-computer interaction (Landay & Myers, 2001). Examples of interaction modes that informal interfaces support include speaking, writing, gesturing, and sketching. In other words, informal modes of interaction are precisely the modes of interaction humans use normally when communicating directly with other humans. However, in contrast to recognition-based systems, informal interfaces avoid early and excessive recognition and transformation of users' natural input, preferring to preserve the users' ambiguous intent.

We are interested in the exploration of informal interfaces in general, and in the Group for User Interface Research, we have developed informal applications to support graphical UI design (Landay & Myers, 2001), speech UI design (Klemmer et al., 2000), and group note taking (Davis et al., 1999; Landay & Davis, 1999). We know that designers in general employ ambiguous means of expression and communication, such as sketching on paper, when they are exploring design ideas (Erickson, 1995; Lawson, 1994; Robbins, 1994; Sano, 1996; Wong, 1992). Because Web design is an emerging field, the tools to support it are not yet mature. We believe that there is a real opportunity for improving the state of the art.

To understand how we might build a tool that supports the Web design process, we conducted an ethnographic study in which we observed and interviewed several professional Web designers. This study showed that the process of designing a Web site involves an iterative progression from less detailed to more detailed representations of the site. For example, designers often create *site maps* early in the process, high-level representations of a site in which each page or set of pages is depicted as a label. They then proceed to create *storyboards* of interaction sequences, which employ minimal page-level detail and focus instead on the navigational elements required to get from one page to another. Later still, designers create *schematics* and *mock-ups*, which are more detailed representations of individual pages.

The design process often includes rapid exploration early on, with designers creating many low-fidelity sketches on paper. These sketches are considered crucial to the process. Designers can quickly sketch the overall structure and navigation of a Web site without having to deal with unnecessary low-level details and without having to commit a large amount of time and effort to a single idea. Furthermore, sketches are important for communicating

ideas with other team members and gaining valuable feedback early in the design process. These uses of sketches are similar to what has been previously reported for GUI design (Landay & Myers, 2001; Wong, 1992).

These were the primary observations that led to the design and implementation of DENIM, a system to assist Web designers in the early stages of information and navigation interaction design. DENIM is an informal pen-based system that allows designers to quickly sketch Web pages, create links among them, and interact with them in a run mode. The different ways of viewing a Web site, from site map to storyboard to individual pages, are integrated through the use of zooming. An informal evaluation of this system has yielded positive comments and indicated that designers will find a system like DENIM useful in their work.

Aspects of this work have been reported in Lin, Newman, Hong, and Landay (2000) and Newman and Landay (2000). In this article, we expand on that earlier work by presenting much greater detail about both the Web design study and the implementation of DENIM. We also present a number of enhancements to the DENIM system, many of which were driven by the experiences of users both in classes at UC Berkeley and in the public at large who have downloaded DENIM since its release on the Web in May 2000.

The remainder of this article is structured as follows: In Section 2, we describe our study of Web design practice and the observations resulting from that study that guided the development of DENIM. We discuss observations concerning specialization within Web design, the Web site design process, common artifacts of the design process, and tools currently being used by designers. In Section 3, we describe the DENIM application, including the UI, details about the implementation, and the results of an evaluation with seven professional designers. Related work is described in Section 4, and we conclude in Section 5.

2. INVESTIGATION INTO WEB SITE DESIGN

Our study into Web design practice had several goals. We wished to learn about designers' current practices, to build a tool that would meet their needs. We wished also to test our hypothesis that an informal interface would fit well with designers' current practices. Finally, we wished to identify which aspects of Web site design practice could benefit the most from being supported by an informal tool.

2.1. Description of the Study

We interviewed 11 professional Web designers during the winter of 1998–1999. We conducted each interview in the designers' workplaces and

collected numerous artifacts of the design process, including sketches, prototypes, written documents, presentations, finished Web sites, and several other types of artifacts, some of which will be discussed later. Eight of the 11 designers we interviewed worked in situations in which they were typically contracted by outside clients to design sites or other interactive products. Of these, seven worked for four different firms, and one was a freelancer. The remaining four worked in the design department of a large Web portal.

A range of experience levels was present in the group of interviewees, with most (seven) having five years or less. Three others had between five and ten years of experience, and one had been a practicing designer for over twenty years.

Eight designers had educational or professional experience in graphic design for printed media prior to getting involved in Web design. The remaining three had varied backgrounds, including software engineering, computer science–interior design, and cognitive science–library science. All of the designers with more than five years of experience had been involved in designing UIs for software applications before getting involved in Web site design.

Though we did not know it when we began the study, we quickly learned that there are a number of subspecialties within the field of Web design, with the most important split coming between *graphic design* and *UI design–information architecture*. We will explore this split and explain the usage of these terms in a later subsection, but for now, we will simply note that the designers we interviewed represented both sides of this split. In particular, four were focused almost exclusively on graphic design, three were focused exclusively on UI design–information architecture, and four had responsibilities that were general enough to incorporate aspects of both kinds of design.

What We Asked

Each study participant was asked to choose a recently completed or nearly completed project and to walk the interviewer through the entire project, explaining what happened at each phase. The designer was asked to show examples of documents that he or she produced during each phase and explain the meaning of the documents with respect to the process as a whole. At the end of some of the interviews, the designer was asked to give copies of the documents discussed during the interview to the interviewer for the interviewer's reference. In this way, examples of design process artifacts were collected from four designers.

Examples of projects discussed include corporate identity and information sites, a state tourism site, a site for a major municipal aquarium, an online clothing catalog, a university site, an online software tutorial, and subsites of a large Web portal.

Validation of Our Observations

At the conclusion of our study, we returned to one of the design firms we had studied and presented our observations (described in the rest of Section 2). We also presented a series of sketches depicting our initial ideas for DENIM. Attending the presentation were one of the designers originally interviewed, three other designers, and the principal of the firm. We received valuable feedback from the firm members, most of which validated our observations. Several corrections were offered as well, especially regarding our use of terminology. We also received positive response to our initial ideas for DENIM.

2.2. Specialization Within Web Design

Designers were careful to use specific terms to refer to different areas of concern within the Web design space. Throughout the article we have attempted to use these terms in the same manner as the designers we interviewed. The references accompanying the terms' descriptions in this section are not meant to be definitive but are meant to reinforce the usage of the terms by designers. It is important to recognize that we are using the terms as they were used by practicing designers and not strictly as they were defined in the literature referenced.

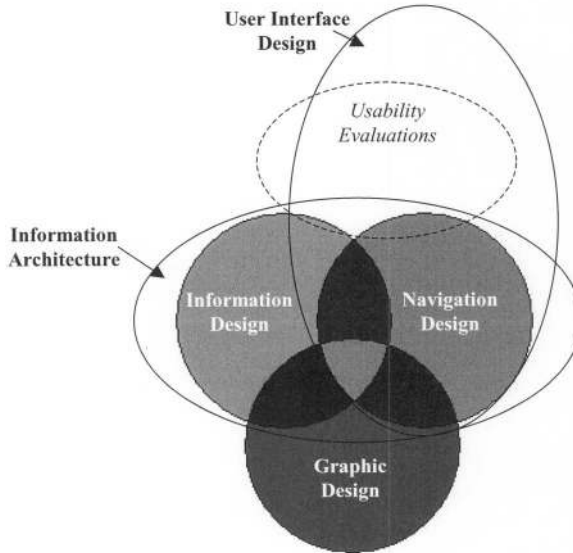
The term *information design* (Shedroff, 1994) was used to refer to the problem of identifying groups of related content and structuring information into a coherent whole. A closely related area, *navigation design* (Fleming, 1998), refers to the design of methods of finding one's way around the information structure. *Graphic design* (also called *visual design*; Cataldo, 1966) refers to the visual communication of information using elements such as color, images, typography, and layout. Whereas information and navigation design focus on the entire Web site and the relation between large-scale elements (such as pages) within the site, graphic design focuses primarily on the presentation of individual elements.

Information architecture (Rosenfeld & Morville, 1998) is an emerging specialty within Web site design that refers primarily to the combination of information design and navigation design.

The term *user interface design* (Lewis & Rieman, 1993; Shneiderman, 1997), when applied in the Web domain, refers primarily to the design of navigation systems, with some overlap into information design and graphic design. In addition, an individual specializing in UI design often has responsibilities extending to testing and verification of the overall usability of the site.

Figure 1 represents the relations among the different areas of design. There are many areas of overlap between different types of design. For example, the

Figure 1. Web design is subdivided into several focused specialties. In some organizations, different individuals focus on different specialties whereas, in others, a single individual will focus on different specialties at different times during the design process.



design of an individual page must take into consideration the information that is to be presented on the page, its relation to other information found elsewhere on the site, the support for navigation to other areas of the site, and the visual presentation of information on the page.

We said earlier that some of the study participants were specialists in one area or the other, and some worked across the boundaries. The level of specialization was mostly determined by the structure of the organization, with two of the five companies designating specialists as either “graphic designer” or “information architect–user interface designer” (in both cases the hybrid title was used). Perhaps not surprisingly, these were the two largest companies (the Web portal and a large design–development firm with just over 50 employees). The two mid-sized companies (roughly 5 and 15 employees, respectively) were the ones that had hybrid designers—individuals who tackled information architecture, UI design, and graphic design tasks at different points in the design process. The two smallest organizations—the freelancer, and a design “firm” that essentially consisted of one individual with additional, project-specific, short-term help—were each separately specialized in graphic design and UI design–information architecture, respectively.

In almost all cases, information and navigation design were done *before* graphic design. At the Web portal, the graphic designers preferred to have the

information structure worked out before the project reached their desks. In the firms at which a single designer would focus on different types of design at different phases of the process, he or she would switch to graphic design only after working out the information structure and obtaining approval from the client. One firm tended to work on graphic design ideas before (or sometimes in place of) working on information and navigation design. This discrepancy seems to have arisen from the firm's background in print advertising and their emphasis on novel, entertainment-oriented sites.

2.3. The Story of a Design: A Software Tutorial

Before presenting a general description of the design process, it will be helpful to ground the discussion with a look at a particular design project described in one of the interviews. The project described was a tutorial for a suite of software CAD tools. The tutorial was designed for deployment on intranets of companies using the client's CAD tools, remote access via the Internet, and distribution on CD-ROM.

This project was one of the shorter projects discussed in the interviews, although the overall process and the artifacts produced are representative of the projects described in other interviews. The durations of each phase of the design, however, should be taken with a grain of salt, as there was a great deal of variation among projects. The *relative* amount of time dedicated to each phase is consistent with projects described by the other designers.

The design team for this project consisted primarily of a designer, a creative project lead, and an account manager. The designer carried out most of the design work, in close consultation with the creative lead and with other designers in the firm. Other team members were concerned with other aspects such as client contact, budget, and schedule.

During the first 2 weeks of the project, the designer immersed himself in the background information for the project. This consisted mostly of reviewing the previous version of the tutorial, as this was a complete redesign of an existing product, and engaging in extensive discussions with the client to understand the content of the tutorial and get feedback about what was desired for the new version. During this time, he also sketched some ideas on paper, including representations of the structure and navigation of the previous version of the product, and new structures representing ideas about how to improve certain aspects. At the end of the 2 weeks, a written "needs analysis" document, detailing project goals, schedule, and general design directions, was delivered to the client.

A meeting with the client was scheduled for the week following the delivery of the needs analysis, at which initial ideas for the redesigned product were to be presented. The designer spent the week generating "initial design

variations,” which focused on the high-level structure of the tutorial and the basic means of navigating the structure. He first made about 20 sketches on paper representing the overall structure (see Figure 2), individual pages (see Figure 3), and specific interaction sequences (see Figure 6). To create something “presentable” for the client, he then created two variations of the site structure and navigation using Adobe Illustrator (Adobe, 1987), which he showed to the client as a large-format color printout. He also created a walkthrough of the structures. The walkthrough was created as a sequential presentation in Macromedia Director (Macromedia, 1989), consisting of images produced in Adobe Illustrator.

The images presented in the walkthrough were representations of individual pages in the tutorial. These representations were devoid of images and icons, used a simple color scheme consisting of three colors (blue, green, and black), and contained almost no typographic variation. The colors used for these representations were not intended to show the colors that the final pages would be but instead were used to differentiate different types of content. The designer said he chose blue and green for these initial images simply “because blue is different from green.” He intended to show that different regions of certain pages would be colored differently from each other to distinguish them, but he did not intend to propose what the final colors would be. Similarly, the bland typography and lack of images were not intended to represent decisions about the final product but were used intentionally to keep the focus on the “mental model” of the tutorial, that is, the overall structure and the means of navigating that structure.

After the presentation of the initial design variations, the designer had a week to prepare the first round of “visual design variations.” Whereas the initial design variations were intentionally devoid of graphic details, the visual variations were intended to address these details. In particular, high-fidelity mock-ups of the home page and one second level page were created (figures not available, but see Figure 8 for an example of a mock-up). These mock-ups contained images, icons, rich typography, and sophisticated color schemes, and these details of the visual presentation were meant to be taken literally.

To produce the visual variations, the designer made a few “very quick” sketches on paper, and then created mock-ups using the “Paint” window of Director. In addition, three other designers within the firm were asked to create mock-ups, to give the client a wide range of options from which to choose. All of the mock-ups were based on the initial design variations. As was done the previous week, a Director presentation was made to the client, this time showing electronic mock-ups of five different design ideas. The client selected two designs for further development, and a meeting was set for the following week.

The designer spent the next week refining and developing the selected designs using Director. The next presentation included not only the refined

Figure 2. This sketch, made by one of the designers interviewed for the study, depicts a portion of the overall structure of the CAD software tutorial described in Section 2.3.

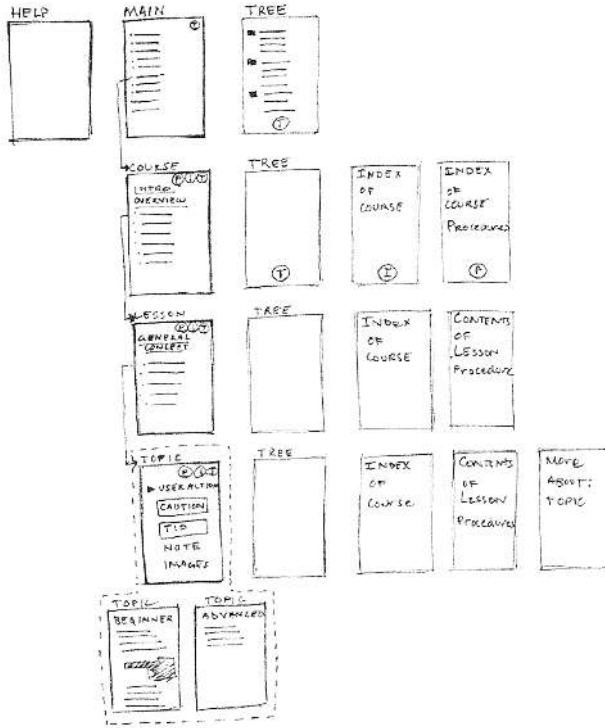


Figure 3. A single page within the CAD tutorial is depicted in this sketch, which was obtained from the same designer that provided Figure 2.



home pages and second level pages, but several other “content pages” as well. The goal of this presentation was for the client to select a single design for development into a prototype. It turned out that the client liked aspects of both designs, so the two were merged and the hybrid design was selected for further development.

At this point, the client announced that they wanted a prototype produced as soon as possible for an upcoming trade show in 3 weeks’ time. This shortened the amount of time that the designer could spend refining and developing the visual design ideas and forced an early transition into “production mode.” He worked on the mock-ups for a little bit longer before beginning to code the prototype in HTML. He said that his normal practice is to flesh out the mock-ups as completely as possible before starting to code, because he likes to “in Photoshop make this as complete as I can and then switch my mind from visual design into coding.” Once he begins coding, he does not work on the mock-ups anymore.

For the 2 weeks while working on the prototype, he used Adobe Photoshop (Adobe, 1989) to work on images and icons and Bare Bones Software’s BBEEdit (Bare Bones Software, 1993) to write the HTML. He also used Netscape Navigator (Netscape, 1994) to preview the prototype.

According to the designer, the development of a prototype is usually followed by the writing of guidelines or a specification to accompany and specify the prototype. Such a document would be handed off to whomever would develop the design into a working product. At the time of the interview, however, the guidelines had not been written. The client had not determined whether they wished to develop the prototype into a product, or whether the prototype was to be used to convince the client organization’s management to pursue a more serious redesign. Without knowing the ultimate fate of the design, neither the client nor the design firm thought it worthwhile to devote time and effort to producing guidelines.

2.4. Design Process

As was seen in the preceding story, designers follow a process of iterative refinement that moves the design from high-level and general to increasingly specific and detailed. Depending on the designer and the organization in which the designer works, the process that is followed may be less or more explicit. In the types of design firms studied in this investigation, the process tends to be explicit, largely because it directly structures the interaction between the designers within the firm and clients and other stakeholders.

Each phase of the design process is usually punctuated by a presentation to the client at which the designers obtain approval from the client (often called

“sign-off”) about the work that was performed during that phase. The explicit design process, which is often published on the firm’s Web site or made available to clients in other published forms, is also used to educate new and potential clients about how the firm operates and what they can expect. Only the Web portal and the freelance designer did not have explicit, published processes, though the designers at the Web portal claimed that they were in the process of developing one internally.

Presented here is a generalized design process, derived from the processes described by the designers we interviewed and refined in subsequent conversations with them and with other designers. This process has four phases: discovery, design exploration, design refinement, and production. The number of phases is consistent with the three to five phases found in a short survey of published design processes from several other firms (Evolve Design, 1999; Fire Engine Red, 1999; Studio Archetype, 1998; Young Ideas, 1999).

Moreover, the design process described here is not particularly unique; we make no claim to having “discovered” it. Crampton Smith and Tabor (1996) describe a generic interaction design process that consists of the phases of understanding, abstracting, structuring, representing, and detailing. Love (1980) outlines the phases of formal design processes in several domains, including mass production, construction, and book development, all of which have the same general shape and many of the same phases. He also describes a generic four-phase cross-domain process comprising concept, preliminary design, detailed design, and first production, the outputs of which are, respectively, concept, model, prototype, and mature design. Even more abstractly, Rowe (1987) reviews a number of staged-process models of design that were proposed in the early 1960s that followed the same pattern of iterative refinement.

Discovery

The purpose of the discovery phase is to determine and clarify the scope of the project, the desires of the client, and the characteristics, needs, or both, of the intended users. If the project is a revision or redesign of an existing site or product, the designers will carefully review and evaluate the existing version. It is common to perform a *competitive analysis* during this phase, which involves reviewing and evaluating competitors’ products for common features and opportunities for improvement and differentiation. Other techniques that might be applied at this phase include interviewing or corresponding with the client to clarify aspects of what is expected, and various techniques to discover the needs of the users, such as interviewing, observing, testing, or surveying.

Design Exploration

During the design exploration phase, possible solutions to the problems identified in the discovery phase are generated and explored. Information design, navigation design, and rough graphic design are often performed during this phase. Multiple rough design ideas and variations are generated. Initial designs generated at this point may or may not reflect ideas about color, imagery, and typography; often they do not. They often *do* reflect ideas about site structure and navigation, though this is not universal. Normally the goal of this phase is to quickly produce several designs and present them to the client, who is expected to select one for further development.

Design Refinement

After a design idea has been selected from the variations presented in the design exploration phase, the designers develop the selected idea further. During this phase the design is iteratively refined and detailed. Such aspects as the precise typeface of labels and body text, the exact sizes and appearances of images, and color schemes and palettes are determined. For most sites, it is not necessary to design every single page of the site, because the site will have been broken down into classes of pages (e.g., home page, second-level pages, and pages for specific types of content), each of which can be represented by an example or *template*. A fully detailed example of each type of page is usually considered sufficient to represent the design.

All of the design firms we studied incorporate user testing, formal usability inspections, or both into at least some of their projects. The Web portal also incorporates usability evaluations on a regular basis.¹ Whether or not usability evaluations are included depends on the client (who must authorize such work), the project scope and budget, and to some extent on the designers themselves (who can suggest whether to include evaluation in the initial project proposal). In the cases in which usability evaluations were performed, they were typically included during the design refinement and/or production phases. In most cases, an interactive prototype would be developed before user testing would be performed. None of the designers reported conducting user tests with low-fidelity prototypes (Rettig, 1994), though this subject was not probed in detail.

Production

When the design has reached a satisfactory level of detail, or when the deadlines and budget dictate that design should end and implementation be-

¹The freelance designer we interviewed focuses mostly on graphic design and does not himself conduct usability evaluations as a regular part of his design process.

gin, designers prepare the design for handoff to the people who will implement the site. *Production* refers to the creation of whatever artifact or set of artifacts will be delivered to the client (or to the software development team) to embody and represent the design. Such artifacts may include interactive prototypes, written descriptions, guidelines, and specifications.

Design Process Timeline

A rough timeline of the design phases, their durations, and the types of artifacts typically observed in each phase are seen in Figure 4. The specific durations were typical of the projects discussed in the interviews, but there were a few projects that took much longer overall. Even in the longer projects, the *relative* durations of each phase were close to those described; in other words, it was common that the design refinement phase was three to four times as long as the discovery or design exploration phase.

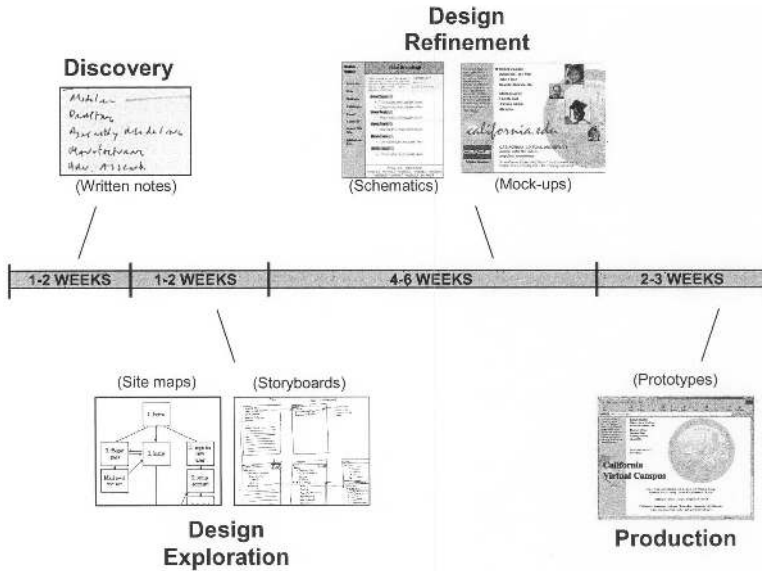
Individual and Collaborative Activity

All of the designers we interviewed did all of their work in the context of teams. Each project we discussed had a design team whose size varied from 2 to around 10, and whose composition was somewhat heterogeneous, including, in some cases, designers of various specialties (graphic, information, UI), account representatives (who managed contact with the clients), business strategists, and a project manager. In a couple of cases, more technical people, such as programmers or database administrators, would be involved from the outset, but this was rare.

Collaborative activity was present throughout the process, but it was more prevalent around the beginning and end of each phase or around the planning and delivery of milestones. For example, several project members would get together to brainstorm ideas about the information layout or navigation and would often sketch rough ideas on paper or on whiteboards. Responsibility for fleshing out various aspects of the rough ideas that were generated would be parceled out, and individual designers would work individually on the items that became their responsibility. Often these designers would generate a number of ideas and informally show them around to other team members for feedback. As a milestone or phase conclusion approached, collaboration would become more active again, as attempts were made to integrate the work of the various team members into a coherent whole to be presented to the client.

Also, designers would regularly draw on other designers outside their team, but in the same firm or organization, for feedback or assistance on their designs. This was sometimes done to take advantage of some specialized knowledge or experience that another designer might have, but was also of-

Figure 4. Reading from left to right, this timeline represents the ordering of the design phases and the kinds of artifacts likely to be produced in each phase. The specific durations indicated were typical of the projects discussed during the interviews, though some projects were much longer. The relative durations of the phases were fairly consistent, even for the longer projects.



ten done simply to get a fresh perspective from someone who was not immersed in the project and could see the work more objectively. As one would expect, the interactions among designers, both inside and outside the project team, were far more relaxed and informal than the much more structured and scheduled interactions with outside clients.

2.5. Products of Design Practice

Throughout the design process, the Web site being designed is represented as a set of intermediate artifacts, such as site maps, mock-ups, and prototypes, that help facilitate communication among the various individuals involved in the design project. Artifacts may support communication among team members, between designers and clients or other stakeholders outside the design team, between designers and implementers, or simply between the designer and herself. Often an individual artifact will support multiple dimensions of communication. In this section, we discuss some of the most commonly observed artifacts of the Web site design process and attempt to describe them using the terminology used by the designers themselves.

Site Maps

A site map is a diagram showing the structure of a site (see Figure 5). It is used primarily to reflect an understanding of the information structure of the site as it is being built and to a limited extent the navigation structure. In many cases, site maps are only used internally by the design team to organize work and obtain consensus on the goals of the project. In some cases, though, site maps are cleaned up and shared with clients. A site map might, for example, be printed out in a large color format to be shown to the client. Sometimes, site maps are published on the release version of the Web site to support users' navigation through the site, though these are often substantially different from the site maps used internally.

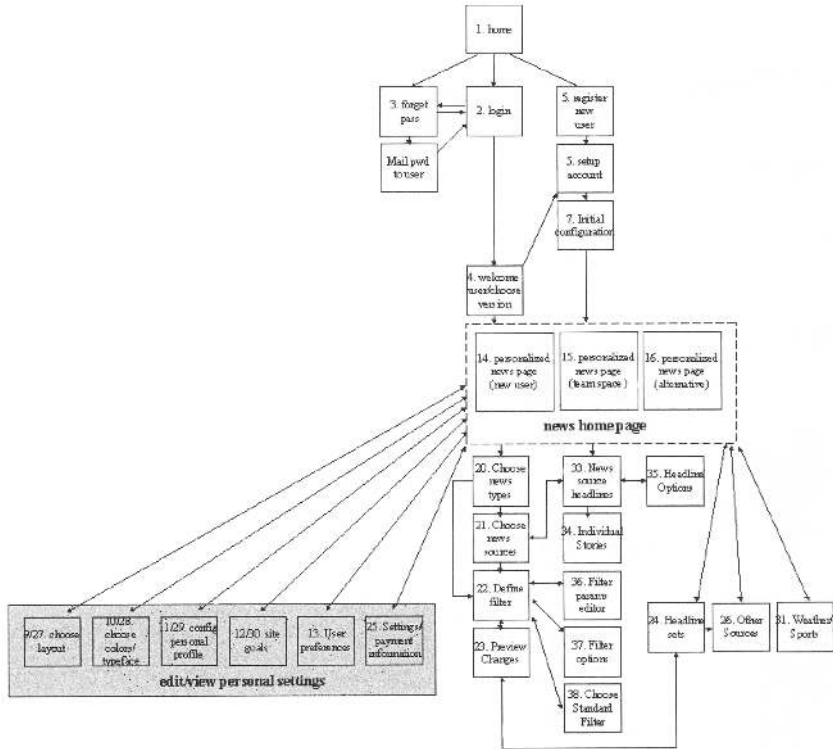
Site maps often evolve throughout the entire life of the project, being updated constantly to reflect new understandings of the site structure. Early in the design process, site maps will reflect the site's structure very broadly, and as time progresses, they will be revised to become increasingly detailed. In some cases, where site maps are used more extensively, they will evolve until they reflect every single page in the site. They can then be used to support project management, content management, and the generation of specifications. Site maps are the primary artifacts of information design, and in organizations that maintain information design specialists, the site map will be generated and updated by that specialist.

Site maps usually consist of labeled blocks and lines as in Figure 5, with some additional features to indicate certain kinds of groupings. The blocks represent individual pages and contain brief descriptions of the contents of the page, often only a short label. The lines and arrows represent navigational paths between pages. Often just the "primary" navigational paths are reflected in the site map. For example, even though it is common for users to be able to reach the home page of a site from any page on the site, this fact is not reflected on a site map such as the one in Figure 5—it is just assumed.

Storyboards

A storyboard is a representation of a particular interaction sequence. It is accompanied, either explicitly or implicitly, by a story or scenario about the task the user would be trying to accomplish via the particular sequence depicted. Storyboards reflect limited detail about the contents of each page in the sequence, and only the navigation links required to accomplish the task are represented. For example, the storyboard shown in Figure 6 illustrates an interaction sequence that a user might execute to access information within a tutorial system. It shows what would happen if a user started at the main page, clicked Begin Tutorial, then clicked Courses, and then clicked Modeling. One

Figure 5. Site maps are high-level visualizations of site structure in which Web pages or entire subsections of the site are represented by textual labels. This image was obtained after the conclusion of the study from a different designer but is highly representative of the site maps observed during the original interviews.



other possible sequence is shown: When the user clicks Cast Contents, she will be taken to the Main Menu page. It is clear that there are links on several of the pages depicted that would lead to other pages, but those interactions are not shown.

Like site maps, storyboards are primarily used within design teams to communicate ideas about site structure and navigation, and are not used to communicate with people outside the team, that is, clients. The idea of presenting a scenario to a client is quite common, only it is usually not done using storyboards. Rather, designers prefer what they call the *walkthrough*, which, like a storyboard, is accompanied by a story about what the user is doing and perhaps why. Whereas a storyboard is a document showing multiple pages at once and the transitions between them, a walkthrough is a mediated, sequential presentation of screens narrated by the designer with an explanation of

Figure 6. Storyboards represent sequences of interactions that a user would carry out to accomplish a task. This storyboard, drawn by the same designer that drew Figures 2 and 3, shows how a user would interact with a portion of the CAD tutorial system to find information on a specific topic.



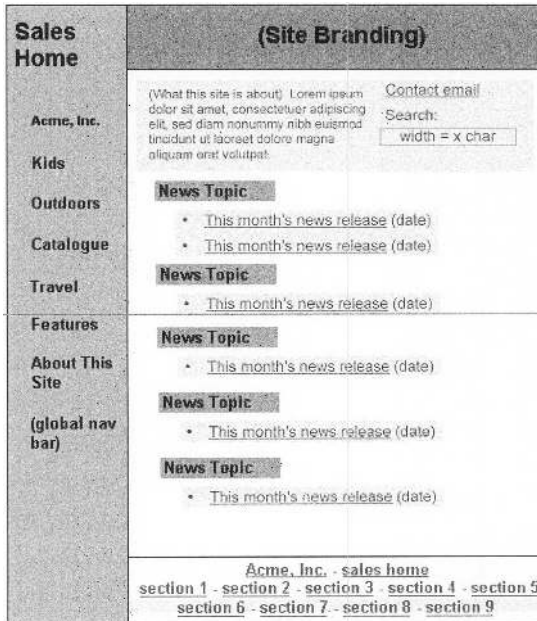
what the user is doing on each screen. A storyboard might well be used to design a walkthrough.

Schematics

Schematics are representations of the content that should appear on a particular page (see Figure 7). They are usually devoid of images, though they may indicate with a label where an image should be placed. Though schematics are not meant to show how color, typography, and graphics will be used on the page, they may themselves use simple color (often they are monochrome or grayscale), typography, and graphics to indicate other things about the page. For example, simple typographic variations may be used to show that a particular label is supposed to be larger and bolder than other labels on the page. Colors and lines may be used to separate regions of a page from each other and indicate that those regions should be made visually distinct from one another when the graphic design for the page is done. Schematics often mix actual page contents with annotations indicating the type of content that should appear in a particular region.

Even though schematics focus on individual pages, they fall into the domain of information and navigation design rather than graphic design. All of

Figure 7. Schematics show the types of information and the information groupings on an individual page. This example, which was modified slightly from one obtained during one of the interviews to disguise the name of the client, shows a schematic of a template for a page that will be filled in largely with dynamic content.



the information architecture specialists created schematics as part of their work, whereas none of the graphic design specialists did. This is because schematics represent the information organization on a given page and the navigational elements that must be included on the page (e.g., links to other pages, navigation bars, feedback about the page's location within the site). In each case where specialization among designers was observed, schematics were used as a means of communication between the information architect and the graphic designer: The information architect would specify the page contents using a schematic, and the graphic designer would determine how to present the contents in a clear and visually appealing manner. Designers in the organizations where specialization was not observed regularly produced schematics before working out the graphic design. We observed examples of schematics at all five companies we studied, though the freelance designer did not appear to use them.

Electronically produced page schematics are sometimes shown to clients during the early phases of design because they do not look like finished Web pages. They can be made to look aesthetically pleasing and professional with-

out appearing “finished,” so they are appropriate for client presentations during early design. Presenting too polished a representation encourages clients to focus on irrelevant details such as fonts, colors, and images when it is often desirable at this point to get feedback on the structure and organization of information (Wong, 1992). At the same time, some designers felt that presenting too rough a representation can seem unprofessional and unimpressive. For design firms working with new clients, it is often important that they make a positive impression early in the design process to reinforce that the client made a good decision in hiring the firm. Early presentations must strike a delicate balance between keeping the focus on basic, structural issues and making a good impression. Schematics were regarded by several of the designers interviewed as a good way to balance these demands.

A concept that was often discussed in conjunction with schematics was that of the *template*. A template is a representation of a *type* of page (e.g., “second level page,” or “product description page”) or a page whose contents will be updated dynamically. Though a schematic might be used to represent a template, the meanings of the two terms are subtly different. Whereas a schematic intentionally declines to specify the graphical presentation of the page, a template intentionally declines to specify the exact content of the page. An individual artifact may be both a schematic and a template, as is the case in Figure 7, but it is also possible for an artifact to be one and not the other.

Mock-Ups

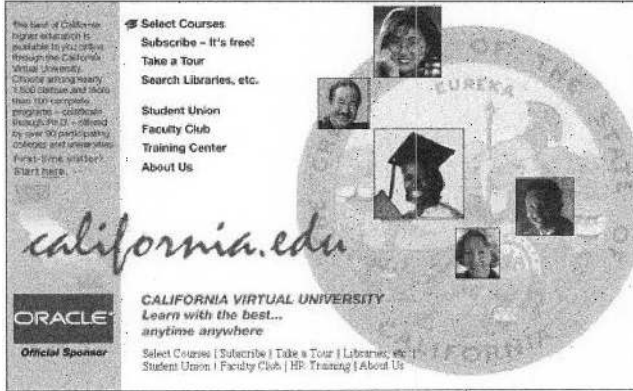
Web designers use the term *mock-up* to describe a high-fidelity representation of a Web page that shows exactly what the page is supposed to look like. They are usually produced using a graphics application like Photoshop and are not interactive. Unlike schematics, the graphic design of a mock-up is meant to be taken literally. The mock-up shown in Figure 8 is a literal representation of a site’s home page.

In some cases, mock-ups are the final deliverable of a design project, perhaps accompanied by written guidelines or specifications.

Prototypes

Though the term *prototype* could refer to anything that serves to represent the system as a whole and therefore is occasionally used to refer to a site map, a set of schematics, or a set of mock-ups, it is most often used to refer to an *interactive prototype*. Interactive prototypes are usually done in HTML or a prototyping tool like Macromedia Director, and allow the designer to demonstrate how the user will interact with the finished site. Prototypes are usually produced late in the design process, that is, during the production phase.

Figure 8. Mock-ups are high-fidelity representations of Web pages, and are meant to look as much like the intended final product as possible. This mock-up was created for one of the projects discussed in the interviews.



Specifications and Guidelines

Specifications are detailed documents that attempt to describe exhaustively and precisely the intent of the design. They usually accompany some kind of a prototype and refer to it explicitly. The intended audience for a specification is the developers who will implement the site. The specification tries to instruct the developers about how to extrapolate from the prototype to the finished product.

Guidelines are similar to specifications, though the term *guideline* implies something less rigid and detailed than a “specification.” Whereas a specification can be thought of as a set of exact instructions about how to build the site, guidelines are more like suggestions. Guidelines do not have to be as comprehensive, and they can leave more details to the discretion of the developers.

Although some designers use the two terms interchangeably, for at least one firm studied, the distinction between a specification and a guideline was considered extremely important. The principal of this firm said that there is a factor of 10 difference in terms of production effort and cost between a specification and a guideline. In one case, this principal reported, a client’s misunderstanding about the difference between a specification and a guideline had actually led to a lawsuit.

Several designers expressed a preference for *interactive specifications*, which integrate the specifications with the prototype. The precise form of the interactive specifications vary from firm to firm and from project to project, but generally they provide a way of accessing the specification information about a particular element of the site from the *element itself*, as it appears in the prototype.

Presentations

Especially in the design firms, presentations to the client were regarded by the designers as a significant part of the design process. Because interactions with the client may be limited and somewhat formal, presentations are often the only means available for designers to convey ideas about the design to the client. Designers at all four design firms described the process of creating client presentations as “a design process in itself.” The freelance designer expressed a similar sentiment. One firm had a “theater” for hosting client presentations: an elegant meeting room decked out to look like an old movie theater. The purpose of the room is to impress clients and increase the likelihood that they will react favorably to the designers’ presentations.

Presentations often require strategic planning to evoke the desired response from the client. One designer described some of the complexity of creating a presentation early in the design process. The design team truly wants the client’s feedback, and at the same time wants the client’s approval. It is particularly important at this early phase of the process that the client is not misled into thinking that the site is nearly finished, so it is desirable to make the images presented appear somewhat rough. Similarly, it is not useful to get feedback about irrelevant details that are not appropriate to the early state of the design, such as the fonts used or the background color. On the other hand, the client may be unfamiliar with the designer’s work and may have high expectations, so it is desirable to make a good impression with a polished design that shows off the designer’s strengths. These considerations are often in conflict and need to be carefully balanced when creating a presentation. One designer had worked with an outside contractor for three weeks nearly full time to produce a presentation that was to describe the results of the discovery phase to the client.

At all four design firms, presentations tended to punctuate phases of the process, especially in the early going. Later in the process, a higher comfort level could be achieved that would allow feedback and approval to be sought in less formal ways. For example, during later stages of the process some designers would post work to an extranet and allow the client to review it directly. Early on, however, presentations are frequent and involve the complex balancing act we have just described.

In terms of content, presentations may consist of any of the artifacts described in this section. Electronic mock-ups are the most common elements included, especially during the latter part of design exploration and throughout design refinement, but site maps and page schematics are frequently included as well. As mentioned in the discussion of storyboards earlier, one common way of structuring presentations is the *walkthrough*. In a walkthrough, the presenter leads the audience through a sequence of steps

that a user might take when interacting with a site, showing the pages that the user would see at each step.

At the Web portal, presentations were important but not as central to the design process as they were at the design firms. Because the portal designers were working entirely for “clients” within the organization, the interactions between designers and other stakeholders could be somewhat less formal. Sometimes, however, the internal clients would be powerful executives, or the proposed design would be especially controversial. In those cases, the importance of presentations to the design process was much more like what was experienced by designers at the design firms.

Written Documents

In addition to specifications and guidelines, many other written documents appear throughout the process. A great deal of information regarding work progress, requests for additional work, and requests for feedback, to name only a few of many types of information, is transmitted through e-mail. Additionally, several documents are often produced during the process, including reports on the results of the discovery phase, initial concept ideas (referred to at one company as the “creative brief”), reports on usability studies, market surveys, work schedules, and contracts. Many of these documents were prepared to share with the client. Some actually represented deliverables from the design process and required sign-off from the client. Other than e-mail, most written documents tended to be formal in nature, especially those shared with the client.

Generally speaking, written documents tended to be reports of some kind or textual descriptions of design ideas. In the latter case, the description would normally accompany a graphical representation of the design, which might be intertwined directly with the text or might be presented in a separate document.

2.6. Tools of Design Practice

The story of the designer working on the CAD tutorial illustrated the fact that designers use a wide variety of tools during the course of a project. He sketched with a pen on paper and also used an array of computer applications to accomplish his work. His pattern of use was typical of other designers.

Figure 9 shows which tools were used to produce which artifacts by the designers who participated in the study. This figure also shows the phases in which the artifacts were most prevalent and which design foci each artifact is most related to. Designers were observed to sketch while producing site maps, storyboards, schematics, and mock-ups. All of these artifacts were later

Figure 9. Each design artifact is more relevant in certain phases than in others, and some strongly focus on certain aspects of the site design rather than others. This table lays out the major classes of artifacts observed during the study, the phases in which they were most relevant, the aspect of design on which they focused, and the tools most commonly used to create them (italics indicate the most commonly used tools for each artifact).

Artifact	Phase	Focus	Tools
Site map	Discovery, exploration	Information, navigation	<i>Sketching, Visio</i> , Illustrator
Storyboard	Exploration	Navigation	<i>Sketching</i> , Illustrator
Schematic	Exploration, refinement	Information, navigation	<i>Sketching, Illustrator, Visio</i>
Mock-up	Refinement	Graphic	<i>Photoshop, HTML</i> , Sketch
Prototype	Refinement, production	All	<i>HTML</i> , Director
Specification	Production	All	<i>Word, HTML</i>

reproduced in a more formal state, except for storyboards. Only one example of a “formal” storyboard was observed; all others were in sketch form only.

Sketching on Paper

In keeping with our interest in informal modes of expression and communication, we paid special attention to the ways that designers currently use sketching. All of the designers did at least some sketching on paper, though not necessarily consistently or to the same degree for every project. When sketching did occur, it would take place early in the process, generally during the design exploration phase, and was employed for information and navigation design as well as graphic design. Examples of sketches done in support of information and navigation design can be seen in Figures 2, 3, and 6. At some point, the sketches would be converted into electronic form by recreating them from scratch using a tool such as Illustrator or Photoshop. In almost all cases, once the designer had converted his or her sketches into an electronic format, paper would be abandoned (except for marking up printouts, as discussed later).

Several designers indicated surprise that we wanted to see their sketches and were even mildly reluctant to show them. The presentation of the sketches was accompanied by a series of apologies for their “poor quality,” and disclaimers about how they were “really rough.” Some designers seemed to be somewhat ashamed of their sketches, or perhaps they had misgivings about showing them to a relative stranger. According to several designers, anything presented to the client must look “professional,” which meant at a minimum a color printout or photocopy of a high-resolution mock-up, and usually it meant a mock-up presented on a computer.

Several designers reported that they “used to sketch more.” Though it was not clear exactly what was behind this perceived reduction in sketching, one

designer said that he began working with Illustrator and Photoshop earlier and earlier in projects because he knew he would have to produce something to present to the client very early on. Knowing this, it was much easier to work in an electronic medium from the start. Several other designers agreed that early deadlines drove them to switch from paper to electronic media earlier in the project than they might have liked.

Another designer reported that she switched to working with computer-based tools when she thought she would be making incremental variations to a single general idea. She said,

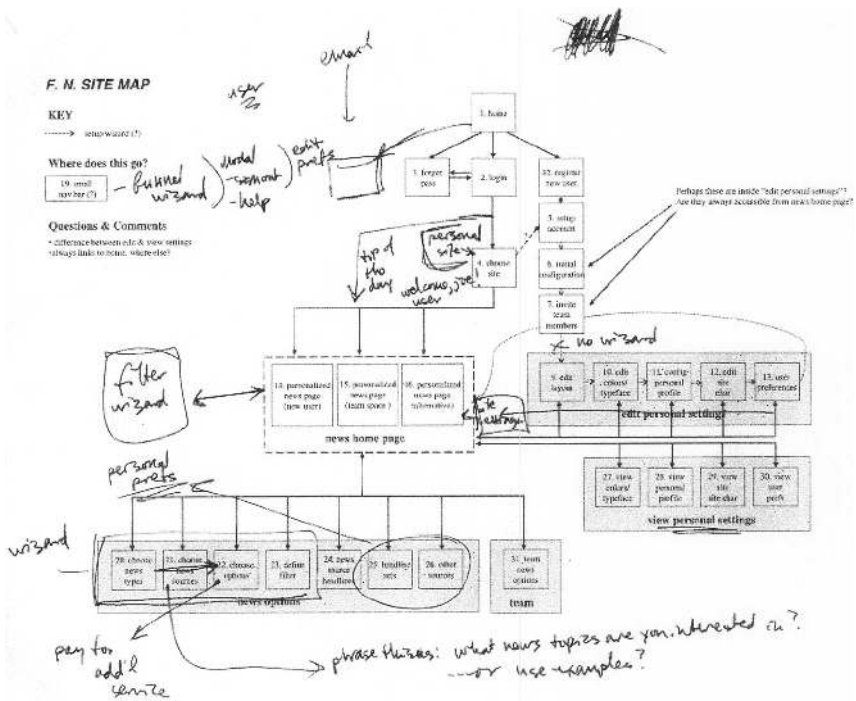
The beginning of each step I'll do on paper. As soon as I feel like I'm going to be starting any design revisions, then I'll move to [an electronic tool] ... because it's easier to make changes to these things.

Some other uses of paper were observed besides personal sketching to work out ideas. Several designers reported using paper and pencil when meeting with other designers. Spontaneous ideas and revisions were captured on paper in these settings. Paper was generally preferred to whiteboards because of its portability: After the meeting one can easily take it with them back to the desk. Designers would also give printouts of electronic sketches to colleagues for comments and they would be returned to them with handwritten annotations (see Figure 10).

Design War Rooms

Two of the companies employed paper in another way during the discovery and design exploration phases. In a process similar to the “affinity diagramming” technique (described in Beyer & Holtzblatt, 1998) for organizing data collected from users, designers would collect ideas about what should be in the site onto Post-It notes and arrange them on the wall into categories (see Figure 11). Whereas affinity diagramming is used specifically to organize and visualize data collected from customer interviews, the Web design firms we observed used the Post-It notes and war rooms in a more general way to collect and structure all kinds of information relevant to the eventual content and organization of the site. This technique amounted to a form of collaborative sketching to determine the site structure. At one of the companies that used this technique, other types of paper artifacts were also attached to the wall, such as annotated printouts of competitors' Web pages. In these cases, it seemed that paper was exploited primarily for its portability, ease of use, and low cost. It is relatively easy to fill a room with pieces of paper and move them around to suggest different associations. The use of large surfaces, such as

Figure 10. Paper is used to support collaboration in several ways. Real-time co-located synchronous sketching on paper is one way that collaboration happens. Another, asynchronous way is that documents are printed, given to other team members, and annotated, as shown in this example of an annotated site map. The site map shown is a different version of the one shown in Figure 5.



walls, allows a large number of complex associations to be represented at the same time and allows easy synchronous collaboration among team members.

Computer-Based Tools

The applications used by the designer of the CAD tutorial described in Section 2.3 were also regularly used by other designers. These designers relied heavily on some combination of Photoshop (Adobe, 1989), Illustrator (Adobe, 1987), and Director (Macromedia, 1989) for much of their work. They used Illustrator and Photoshop to create high-fidelity mock-ups as well as medium-fidelity schematics, whereas Director was used both for creating interactive prototypes and for creating presentations. Some designers used Illustrator for making site maps as well. This set of tools was most heavily used by the designers who spent some or all of their time focusing on graphic de-

Figure 11. This picture is typical of environments observed at two of the firms we studied. One or more walls would be covered with taped-up Post-It notes, sketches, printouts, and other paper materials. The arrangement of materials on the wall would come to resemble an early map of the structure of the site being designed (i.e., an early version of a site map). This photo was taken by Scott Klemmer during a later study of Web design firms (see Klemmer, Newman, Farrell, Bilezikjian, & Landay, 2001).



sign. In other words, the “hybrid” designers described earlier, who transitioned among information, navigation, and graphic design, tended to use these graphic design-oriented tools for most of their work, and had adapted their use of these tools to support each of their activities.

The UI design specialists, on the other hand, did not use the same set of tools. One of the UI designers did not use any graphics programs at all: Her diagrams were all on paper and most of her computer-based work involved writing reports using a word processor. Another UI designer made heavy use of Visio (Microsoft, 1992) for making site maps and schematics. She also used paper sketches to some extent and did a lot of word processing.

In fact, several designers used word processing software frequently, especially those charged with writing specifications and guidelines. None of the graphic design specialists fell into this category, but most of the other designers did. Designers at two firms reported having created interactive specifications in HTML, which involved somewhat less writing than word-processed specifications. However, even these designers still had to produce written specifications or guidelines at least some of the time. Most designers also used word processing software to produce reports (e.g., a report explaining the results of the discovery phase) and proposals (e.g., detailed project plans and schedules).

All of the designers, especially the more experienced designers, tended to be heavily invested in the tools they used. They admitted to using their pre-

ferred tools for tasks that might have been more easily accomplished with another tool. One designer did all of her diagrams, including site maps and schematics, using Microsoft Word's (Microsoft, 1983) drawing utilities. Another designer said he used Director's paint function for all his graphics needs, even though he knew that Photoshop would be better for some of the things he did. He simply did not have time to learn a new program. Similarly, the UI designer who used Visio for diagramming also used Visio for making page schematics, which she acknowledged might be easier to make, or at least more attractive, if they were made using a program with more graphics capability. Again, the potential gain from using a new program did not outweigh the inconvenience of having to learn it.

We were quite surprised to find that *none* of the designers we interviewed used Web development tools like NetObjects Fusion (NetObjects, 1996) or Macromedia Dreamweaver (Macromedia, 1997) on a regular basis. Many of them had never used them at all or had used them once or twice before abandoning them. It is possible that the learning curve on these tools discouraged the designers from adopting them, given their investment in their existing tool set. On the other hand, it is possible that these tools were not used because they focus on *development* of finished Web sites rather than on designing Web sites. In the cases where HTML-based prototypes were developed as part of the design process, most designers preferred to "hand code" the HTML. One designer said that WYSIWYG tools like Dreamweaver were unnecessary because it was relatively easy for him to write the HTML code once he had designed how he wanted the site to work and what he wanted it to look like. Some other designers never touched HTML at all, and worked only with image and diagram manipulation tools.

2.7. Implications for Web Design Tools

The motivation for this study was to guide the design of tools to support Web design. We conclude our discussion of the Web design study by looking at the implications of this study for the design of such tools.

Use an Informal UI

We found support for our hypothesis that an informal interface would be useful to designers. Because all of the designers sketch at least some of the time, and some designers sketch quite a lot, we believe that a sketch-based Web design tool would fit naturally into many designers' work practices. Many designers reported regretfully that they were converting to electronic media earlier in the design process than they had in the past. A tool that provides some of the advantages of an electronic medium (i.e., ease of manipula-

tion and replication) but preserves the ability to sketch may encourage designers to continue to sketch farther into the process. Other research has suggested that prolonging sketching, and therefore the ambiguous representations that are produced by sketching, will result in a broader exploration of the design space (Goel, 1995).

Informal interfaces leverage modes of interaction and communication that are already familiar to users. This means that a good informal interface should be relatively easy to learn and use. As described in the earlier discussion about computer-based tools, ease of learning and use will be critical to the acceptance of any new design tool.

Focus on Intermediate Artifacts

Before undertaking this study, our plan was to develop a tool to allow designers to design finished Web sites by sketching. Through this study, we learned a great deal about the intermediate products of the design process and all that happens between exploration of design ideas and the production of a completed Web site. The production of finished Web sites involves a mode of thinking and expression that is much more precise than the mode used by designers when exploring the design space. We now think it is important to concentrate on supporting creation of other artifacts, such as site maps, storyboards, and schematics, which are more relevant to the early design process than finished Web sites.

Support Multiple Representations

This study found that designers use multiple representations throughout the course of the design process. These representations depict the site at different levels of detail. A design tool should support a similar range of representations. Such a tool would be an improvement over the current state of the art, in which different representations are created using separate, poorly integrated tools. Several designers expressed a wish that the different representations could be tied together in a unified framework so that consistency and coherent project management strategies could be more easily maintained.

Integrate With Other Tools

Because the need to present polished design ideas to clients early in the process is one of the factors driving an early conversion to formal representations, a sketch-based tool should support the integration of sketches with more formal representations produced in other tools such as Photoshop or Illustrator. We plan to explore whether or not designers will take advantage of

the ability to integrate formal and informal representations to continue to sketch later in the design process.

Through this study, we were able to focus our understanding of where in the process an informal tool would fit best and which specific types of design (and designers) it would best support. We found it most appropriate to focus on the design exploration phase and on information and navigation design. Later design phases require greater precision and probably would not benefit greatly from a tool with an informal interface. These phases also place a greater emphasis on graphic design. Though an informal tool may not explicitly support later design phases, it should support transitions into them. We believe the right way to do this is through integration with other tools and representations as mentioned earlier. In addition to graphics applications like Illustrator and diagramming applications like Visio, we found that presentation and word processing software are especially prevalent in designers' work practices. A Web design tool should strive to integrate well with these applications as well.

Manage History and Variations

Designers expressed a desire to have a unified way of managing 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.

To keep track of project milestones and variations designers are forced to invent ad-hoc methods of their own, usually involving saving multiple versions of files and using complex, cryptic file names to encode the relevance of each version. Several designers were interested in having a tool that helped them keep track of project histories so that they could refer back to decisions made early in the process and better understand the context under which they were made.

Integrate With Paper

Paper has many affordances, independent of the affordances of sketching. We would like to explore ways to integrate paper directly with an electronic tool so designers can continue to use paper sketches while still gaining the advantages of an electronic tool. This integration could occur, for example, by incorporating scanned sketches or sketches done on a CrossPad (Cross Pen Computing Group, 1998) into the site framework. Another way that this integration could take place is to allow the spatial arrangement of paper sketches or handwritten notes (similar to the affinity diagramming technique mentioned earlier) and capture this arrangement via cameras. We briefly describe such a system later, in our discussion of current and future work.

3. DENIM: AN INFORMAL WEB SITE DESIGN TOOL

To address the implications raised by our study, we developed an informal tool to support early-phase information and navigation design of Web sites. To emphasize the informal nature of our tool, we named the system DENIM, which also conveniently stands for *Design Environment for Navigation and Information Models*. DENIM is intended for prototyping in the early stages of design but not for the creation of finished Web sites. For example, it does not output finished HTML pages. As we describe in the rest of this section, DENIM provides both an informal, sketch-based interface and the ability to view several representations of the site through zooming. The version of DENIM described in this section focuses mainly on addressing the need to support multiple representations, the desirability of focusing on intermediate artifacts, and the need to support informal interactions. Other implications from the study, such as the need to support transitions to more formal representations and the desirability of providing explicit support for design histories and variations, are important but are not addressed in the current version.

DENIM is designed to be used with a pen input device such as a Wacom Tablet (Wacom Technology Company, 2000) or a device running the Tablet PC operating system (Microsoft, 2002). We built DENIM in Java 2, on top of SATIN, a toolkit for supporting informal pen-based interaction (Hong & Landay, 2000). SATIN will be described briefly in a later section.

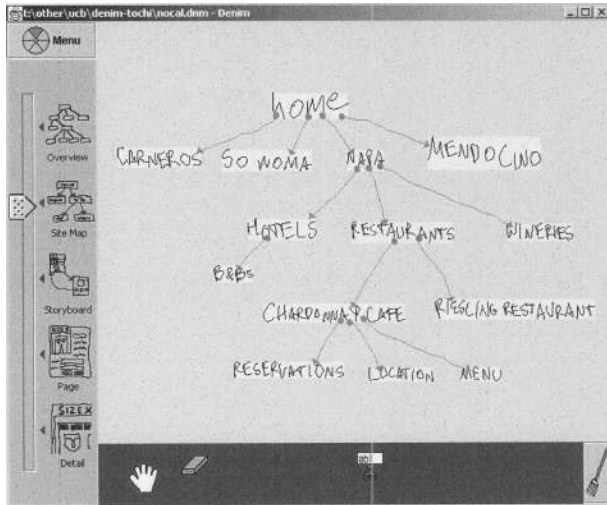
DENIM uses a semantic zooming UI (Bederson & Hollan, 1994) to integrate the representations most commonly used by designers when designing Web sites: site map, storyboard, and individual page. We envision a designer using DENIM during the design exploration phase to create and explore the information and navigation design of the Web site.

Here is one scenario of how DENIM could be used. A designer collects information during the discovery phase and brainstorms the site structure by writing words and phrases representing potential content areas for the site on DENIM's canvas in site map view (see Figure 12). She then moves the phrases around and connects them with arrows to flesh out the structure and to begin visualizing the navigation.

After working on the basic structure, she uses DENIM's page view (see Figure 13) to sketch out the contents of some of the key pages in the site. Zooming out to storyboard view (see Figure 14), she works out in more detail the navigation paths between pages in the site by redrawing the arrows between pages. Unlike the "organizational arrows" originally drawn while brainstorming the basic site structure, these redrawn arrows are "navigational arrows," meaning that they specify exactly which element in the page the user would click to jump to the target page.

The designer previews the user's interaction with the site by using DENIM's *run mode* (see Figure 25). After reflecting on her design and sharing

Figure 12. DENIM is an informal, sketch-based Web site design tool that allows designers to view and interact with a Web site at multiple levels through zooming. In site map view, depicted here, Web pages are represented by their labels, and connections between pages are depicted with directional arrows. A designer can create new pages by writing labels directly on the canvas or typing a label using a text insertion gesture, and can create connections between pages by drawing arrows.



it with her team members to receive feedback, she iteratively refines and revises her design using DENIM's editing facilities until she is satisfied with the site structure and navigation. When she eventually wants to formalize her ideas, as is likely to occur during the design refinement phase, she switches to a different tool such as Illustrator or Photoshop that is better suited to the task.

3.1. The DENIM UI

DENIM Beta 1 was released for free public download via the Web at the beginning of May 2000. Several beta releases followed, and eventually version 1.0 was released on November 21, 2001. Version 1.0 is the system described in this section and depicted in Figures 12–26. The primary DENIM application window is shown in Figures 12, 13, and 14. The window has three main areas:

- The center area is a canvas where the user creates Web pages, sketches the contents of those pages, and draws arrows between pages to represent their relationship to one another.
- On the left is a slider that reflects the current zoom level and allows the level to be set.

Figure 13. In DENIM's page view, a single page is visible and its contents are displayed at 100% resolution. The primary intent of this view is to allow the designer to view and sketch the contents of a particular page. It is also possible to create new pages by drawing a rectangle in a blank part of the canvas.

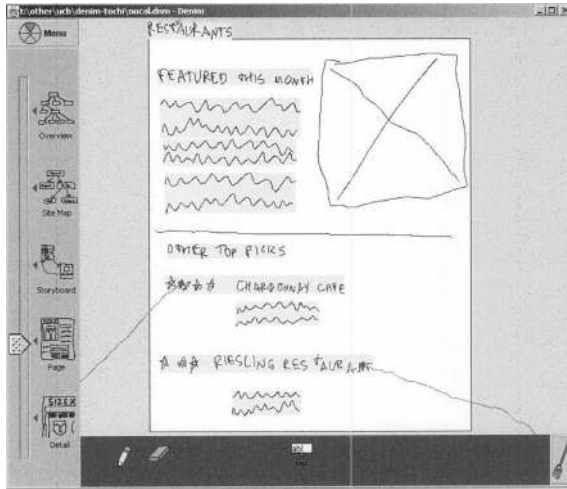
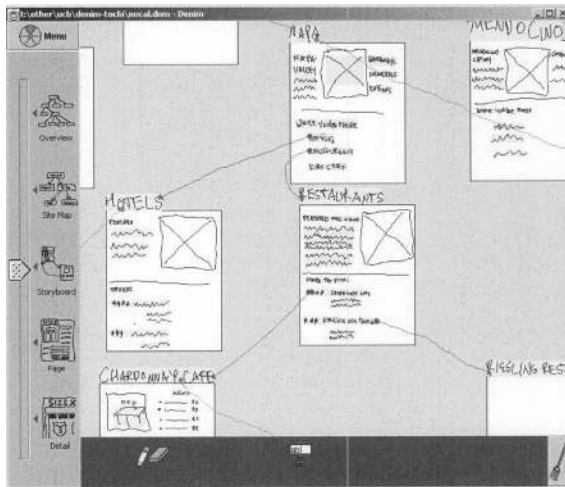


Figure 14. In DENIM's storyboard view, multiple pages and the links between them are visible. Designers can create new pages in this view by writing a label or typing text, as in site map view, or by drawing a rectangle in the size and shape of a page. It is also possible to create navigational arrows in this view, which indicate the links within a page that a user would follow to reach another page.



- The bottom area is a toolbox that holds tools for drawing, panning, erasing, and inserting text fields. The tools behave like local tools (Bederson et al., 1996): Users “pick up” a tool by tapping on it, “drop” a tool by tapping in an empty area in the toolbox, and “swap” tools by tapping on another tool. There are several tools: a hand tool for panning, a pencil and eraser for sketching and erasing the design, and a text field stamp for inserting text fields. A version of DENIM currently being developed includes more tools for inserting other UI elements, like check boxes and radio buttons, and to allow designers to create their own components; see Section 3.5.

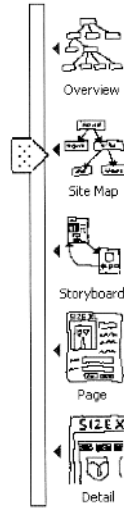
Unifying Representations Through Zooming

There are five main zoom levels in DENIM, which are identified on the zoom slider with icons representing the type of view available at that level (see Figure 15). There is also an intermediate zoom level in between each main level. Three zoom levels—the site map, storyboard, and page levels—map directly to the most common representations of Web site designs that we observed during our study of design practice. The *site map* level (see Figure 12) gives a view of the site as connected labels. The *storyboard* level (see Figure 14) allows the user to view several pages simultaneously and more clearly see the navigational relationships between the pages. The *page* level (see Figure 13) displays pages at “100%” scale and is intended to allow users to sketch the page contents.

In addition to these levels, there are two major levels at the extreme ends of the scale, with the *overview* level providing a more abstract, higher level representation of the entire site with only thumbnails of the individual pages shown, and the *detail* level providing a more fine-grained view of individual pages, for more precise sketching. These two extreme levels did not map directly onto any representations that were observed during the study and were added as an experiment to see if they would be useful in any way. In practice, we have observed that they are not used very much, and that when they are, it is often to fix a drawing mistake (*detail* level) or to find one’s place in the site map after having become disoriented at a closer zoom level (*overview* level).

To change the zoom level, the user either drags the slider’s elevator or clicks directly on one of the icons. Changing the zoom level initiates an animated transition from the current zoom level to the desired zoom level. The center point for a zoom operation can be set by tapping and holding on the background of the canvas. Such a tap causes crosshairs to be displayed at the point tapped, and any subsequent zoom operation will center on that point. Alternatively, if any objects are selected, the center of the selected object or objects is used as the zoom target.

Figure 15. The zoom slider controls the view in DENIM’s canvas. There are five major zoom levels, the middle three of which correspond to common views of Web sites used by professional designers (site map, storyboard, and page). The “overview” provides an alternate, zoomed-out view of the site map, and the “detail” is a close-up view that allows the designer to sketch inside part of a page. Discrete intermediate zoom levels are also available, equidistant from each of the labeled levels, and also including the very top and very bottom of the scale. Altogether there are eleven possible zoom levels accessible via the zoom slider.



Gestures and Pie Menus

Most commands in DENIM can be activated either through gestures² or through pie menus. To activate a gesture, the user presses the button on the barrel of the pen and draws a stroke. To activate the pie menu, the user taps with the barrel button depressed, that is, by gesturing a dot (see Figure 16 for DENIM’s responses to all possible pen actions).

When the user gestures, a thick gray translucent stroke is drawn during the gesture (see Figure 17a). If DENIM recognizes the stroke as a valid gesture, the gray stroke is replaced by an idealized version of the gesture in green (see Figure 17b). If it is not recognized, the gray stroke is flashed in alternating red and gray.

We have implemented gestures for panning, undo, redo, group select (select everything enclosed by a circular gesture), cut (shown in Figure 17), copy, and paste. A list of all supported gestures appears in Figure 18. Tapping and

²By *gesture*, we mean a stroke created by the pen that activates a command.

Figure 16. DENIM is designed for use with a pen input device with a single barrel button. When the pencil tool is selected, the system recognizes five unique pen actions, and treats a sixth (Button-hold) as equivalent to one of those. Tap and Drag are essentially equivalent as well, because both simply mark the canvas wherever the pen was placed. When other tools such as the eraser are in use, tap, drag, and hold all result in the tool being used. The other three actions retain their same interpretations.

Pen Action	Result
Tap	Draw dot
Drag	Draw stroke
Hold	Select object
Button-tap	Open pie menu
Button-drag	Gesture
Button-hold	Same as button-tap (opens pie menu on release)

Figure 17. Commonly used commands can be accessed through gestures. (a) The designer performs a cut gesture over the “Mendocino” page to remove it from the site. (b) DENIM shows that it recognized the cut gesture with an idealized cut gesture in green.

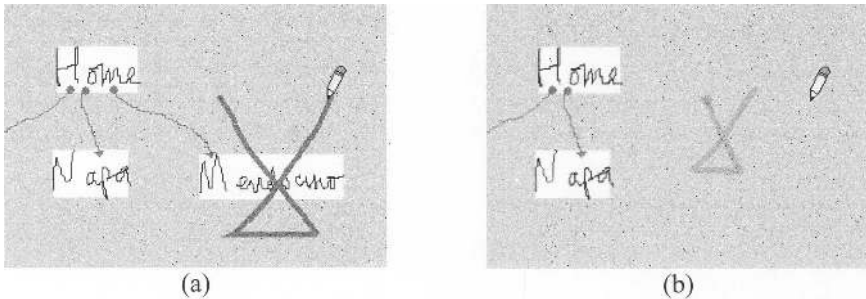
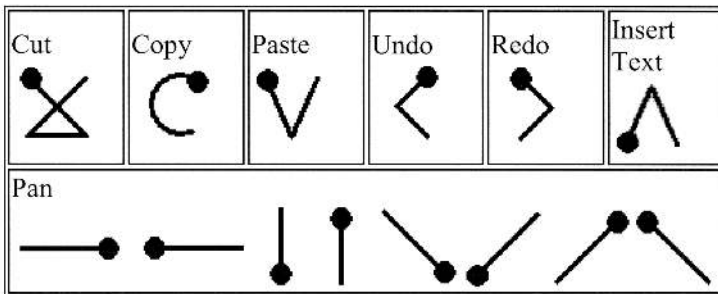


Figure 18. There are seven supported gestures in DENIM. The dot in the gesture indicates the starting point of the gesture. The “Pan” gesture results in the canvas being panned in the direction of the stroke.



holding on an object without depressing the barrel button selects or deselects that object. The selected object can be dragged to move it to a new location. Tapping and holding on the canvas, outside of any Web page, clears the selected objects and sets the zoom-center target, denoted by crosshairs.

Pie menus (Callahan, Hopkins, Weiser, & Shneiderman, 1988) are used to provide access to functions not easily mapped to gestures, as well as providing redundant access to certain commands (see Figure 19). Keyboard shortcuts are available for several commands, including cut, copy, paste, delete, undo, redo, pan, and zoom.

Creating Pages

When in site map view, there are two ways to create pages. The designer can write some words directly on the canvas. DENIM groups strokes that are close to each other into words and phrases. The designer can also type out the words after using a text insert gesture (discussed later). In either case, a new page is automatically created and its label contains the phrase that was written or typed (see Figure 20). Note that pages are not visible in site map view; the page becomes visible when the view is changed to a lower zoom level. Labels remain the same size throughout all the zoom levels (except the overview level), so that they can always be read.

In the storyboard and page views, a new page can be created by drawing a rectangle on the canvas (see Figure 21). If the rectangle is approximately the same size as a page's boundaries in the current view, it is converted to a page with an empty label. A page can also be created by typing out the new page's label, as in site map view.

Adding and Editing Web Page Content

Designers can create and erase drawings within a page by using the pencil and eraser tools in the toolbox (see Figure 22). They can also add text fields, for example, to allow end-users to enter search terms or other data, by picking up the text field stamp and tapping it at the desired location within a page. Currently text fields are the only "widgets" that can be stamped into pages, though future public releases will support more widgets (e.g., radio buttons, check boxes, and list boxes).

Replacing Handwritten Text With Typed Text

Typed text can be used in place of handwritten text in page labels and page contents. To insert typed text, the user performs a "caret" gesture at the desired location (see Figure 23). DENIM then prompts the user to enter text us-

Figure 19. Most commands in DENIM can be accessed through the pie menu. The pie menu is accessed by tapping anywhere on the canvas with the pen button depressed, or by tapping the “menu” button above the Zoom Slider. Several commands are context sensitive, for example, a “Run” command initiated from a pie menu that was invoked over a specific page will load that page as the first page in the DENIM browser. The same command initiated over a blank part of the canvas will load the site’s default home page as the first browser page.

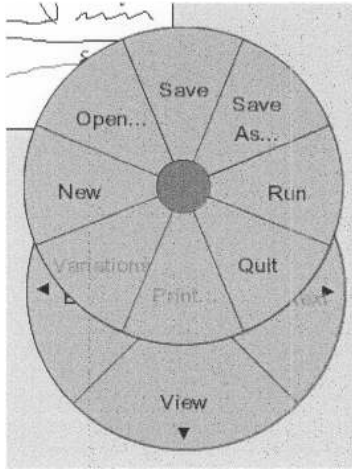


Figure 20. New pages can be created by writing words or phrases directly on the canvas when in site map view or storyboard view. DENIM automatically creates the label and a blank new page.

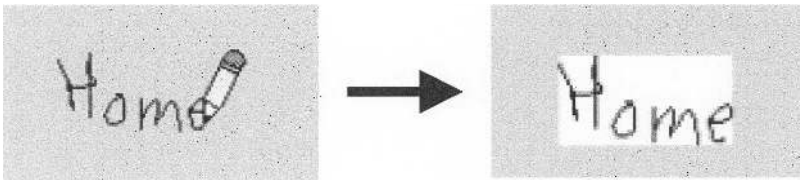


Figure 21. In storyboard and page view, new pages can be created by drawing a rectangle on the canvas. The new page is given a blank label.

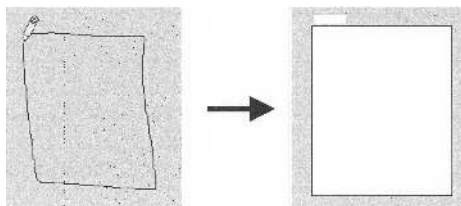


Figure 22. The toolbox holds tools for designers to edit and manipulate their designs. From left to right: hand (panning), pencil (drawing), eraser (erasing), and text field stamp (inserting text fields). The broom button on the right realigns the tools.

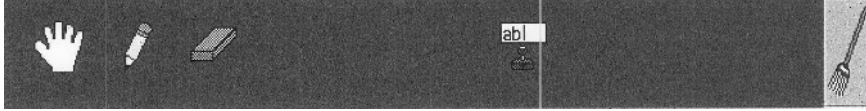
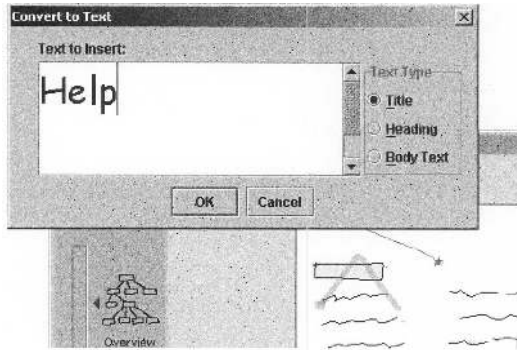


Figure 23. Typed text can be entered by gesturing in the desired location and typing in a phrase. In this example, the “greeked” text underneath the gesture will be replaced by the typed text.

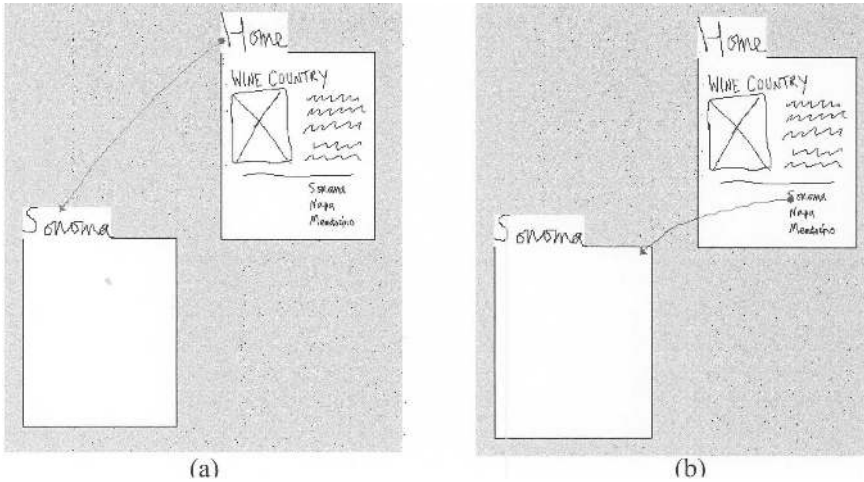


ing the computer’s keyboard. If the insertion gesture was performed over an existing phrase, regardless of whether it was typed or handwritten, the new text replaces the old text. This feature supports designers who prefer typing to handwriting for text input, as well as designers who prefer to start with entirely hand-drawn representations and iteratively refine them as the design process progresses. Both types of designers were encountered in our initial Web design study, as well as in the evaluation of DENIM discussed in Section 3.3.

Creating Arrows and Hyperlinks

An arrow between two pages represents a relation between those pages. We provide *navigational* and *organizational* arrows (see Figure 24). Navigational arrows represent hyperlinks in the HTML sense: They represent the reference from an item on one page (e.g., a word or image) to another page. Organizational arrows are used to represent a conceptual relationship between two pages; that is, the designer eventually wants to make a hyperlink from one page to another but does not want to fill in the details at this time.

Figure 24. Organizational arrows (a) originate from a page label or an empty region of the page indicate that there is a relationship between two pages but do not specify how to navigate from one to the other. They are gray in color. Navigational arrows (b) originate from a word or drawing inside a page and specify which link in the source page a user would click to reach the target page. They are green in color.

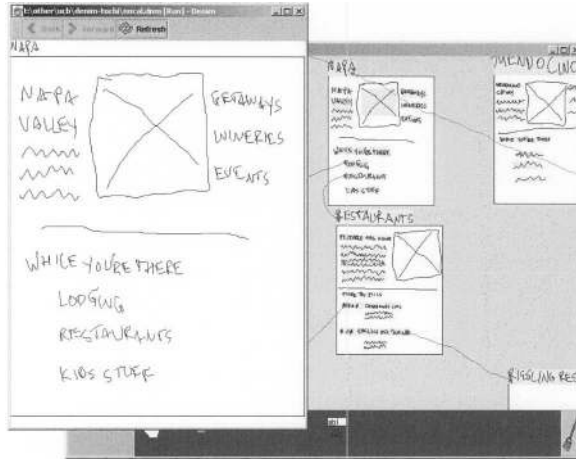


To create a link, the user draws a stroke between two pages. The system checks if the stroke is an arrow. Organizational arrows start on one page and end in another. This creates a *gray* arrow from the source to the destination. Navigational arrows start on a specific object on one page and end in some other page. This creates a *green* arrow from the source to the destination. When creating a navigational arrow, any organizational arrows from the source page to the destination page are removed. As additional feedback, the source of the navigational arrow becomes blue, like a hyperlink anchor in a Web page.

Interacting With Designs in Run Mode

After a number of pages have been sketched and navigational arrows drawn between them, it is possible to preview the interaction by entering *run mode* by choosing File→Run from the pie menu. In run mode, a simplified “browser” window appears on the screen (see Figure 25). The browser displays one page at a time, like a real Web browser, except the pages displayed are the sketches that the designer has created. If an element inside a page is the source of a navigational arrow, it is rendered in blue in the browser. Clicking on these elements causes the browser to display the target of the link, just as in a conventional browser. The browser has Back, Forward, and Refresh

Figure 25. The interaction with a site can be previewed using DENIM's Run Mode. In this mode, a limited-functionality “browser” displays page sketches and allows the user to navigate through the site by clicking hyperlinks, which are colored blue.



buttons, which also operate like their counterparts in conventional browsers do. With run mode, designers can test the interaction with a site that they are designing without having to create a full-fledged prototype. They can also test these early prototypes with target users and get useful feedback (Walker, Takayama, & Landay, 2002).

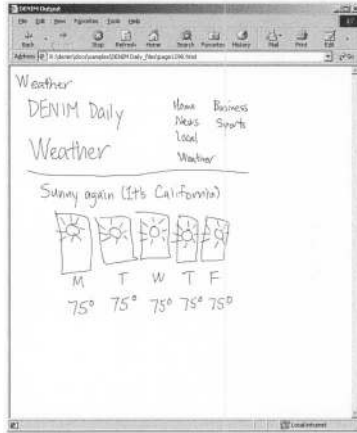
Exporting to HTML

DENIM allows designers to export their designs as a set of HTML pages, so that others can interact with the design without needing to have DENIM. The exported pages look exactly like the designer's sketches. One HTML page, containing one GIF image and an image map, is created for each page in the DENIM design. Objects in the page that serve as source objects for navigation arrows are rendered as active areas of the image map, such that clicking on them causes a transition to the appropriate page in the site. Figure 26 shows an exported page in a conventional Web browser. We are also working on exporting DENIM files to other formats so that designers can use them with other tools. Please see Section 3.5 for more details.

3.2. Implementation Details

As mentioned previously, DENIM was implemented in Java 2 on top of the SATIN toolkit, a toolkit intended to facilitate the creation of sketch-based

Figure 26. DENIM designs can be exported to HTML sites. The exported pages retain their sketchy, hand-drawn nature, but can be viewed and interacted with using a conventional Web browser. Hyperlinks are rendered blue, just as they are in DENIM. This is an HTML page from an exported DENIM design.



applications. SATIN provides three main facilities that are of critical importance to DENIM. First, SATIN provides a framework for capturing and interpreting ink strokes. Custom “interpreters” can be added to any graphical region of an application’s display, and these interpreters can receive and operate on any ink input directed to that region. Second, SATIN provides a scenegraph for manipulating and rendering graphical objects, including support for *semantic zooming* (Bederson & Hollan, 1994). Third, it provides libraries for processing strokes and GUI widgets optimized for pen-based interaction. SATIN was largely co-developed with DENIM, so many of its features fit well with the needs of DENIM.

SATIN Overview

SATIN’s scenegraph is a tree data structure that controls how graphical objects are displayed and how strokes are processed. A node in the scenegraph is called a *GraphicalObject*. SATIN provides several basic *GraphicalObjects*. A *Sheet* serves as the root of a scenegraph. A *Patch* is an arbitrarily shaped region of space somewhat analogous to windows in a windowing system. The simplest *GraphicalObject* is the *TimedStroke*, which is a set of points and the times that they were created. The timing information is stored to assist in sketch recognition.

Each *GraphicalObject* has one or more *Views*. Each view knows its visual boundaries and how to render itself. Each *GraphicalObject* has a default

view, which simply renders the graphical object as is. Application developers can define their own views if needed. SATIN provides several kinds of views, two of which are of special interest here. The first is the *SemanticZoomView*, which multiplexes between different views based on the current zoom level. The other is the *StickyZViewWrapper*. “Sticky Z” means that the view of the object does not change scale when zooming in or out—in other words, it stays at a fixed point on the z-axis relative to the viewport. The object does, however, move around in the x - y plane.

Each *GraphicalObject* also has one or more *Interpreters* for handling strokes. An interpreter is an application-defined object that receives ink strokes before they are rendered to the screen and decides whether to apply an interpretation to them. Each *GraphicalObject* has two sets of interpreters, one for handling gestures and the other for handling ink. When a stroke is dispatched to a *GraphicalObject*, the stroke is delegated to the appropriate set of *Interpreters* for handling. SATIN comes with a *SemanticZoomInterpreter*, which multiplexes between different interpreters based on the current zoom level.

When strokes are drawn, they are dispatched to *GraphicalObjects* similar to how mouse events and keyboard events are dispatched to individual widgets in modern windowing systems. Most windowing systems use a leaf-only dispatch, meaning that only the lowest object in the tree receives an event. For example, clicking on a button widget would cause only that button to receive the event. In contrast, SATIN uses a hybrid top-down, bottom-up approach, where dispatching starts at the root and recursively traverses down the tree, selecting only those *GraphicalObjects* that entirely contain the stroke, and then comes back up. This approach allows *GraphicalObjects* to try processing strokes as gestures first before processing them as ink. It also allows developers to provide global behavior for gestures (for example, by handling a stroke on the *Sheet*), as well as enforcing local behavior in *GraphicalObjects* (for example, straightening up lines in certain *GraphicalObjects* but not others).

Interpreting Ink Strokes in DENIM

DENIM has a global *GestureInterpreter* that uses SATIN’s *StandardGestureInterpreter*, which is based on a modified version of gdt (Long, Landay, & Rowe, 1999) and Rubine’s recognizer (Rubine, 1991) to decide whether ink strokes should be treated as command gestures (e.g., cut, copy, paste). When the *GestureInterpreter* determines that a stroke should be treated as a gesture, it executes the specified command instead of rendering the stroke to the screen. To keep strokes from being rendered to the screen, *Interpreters* have the option to *consume* the stroke, thereby preventing other *Interpreters* and the default rendering mechanism from knowing that the stroke exists.

Interpreters can be attached to the Sheet as well as to Patches. In DENIM, several Interpreters are attached to the *DenimSheet*, which is descended from SATIN's Sheet. In addition to the *GestureInterpreter* mentioned earlier, the *ArrowInterpreter*, *LabelInterpreter*, and *PanelInterpreter* are attached to the Sheet.

The *ArrowInterpreter* determines whether or not to interpret a given stroke as an *Arrow* between two pages. Its rules are simple: If a stroke originates inside a page or a label and terminates in or near another page or label, the stroke is considered an *Arrow*. It further determines whether to instantiate the *Arrow* as a *navigational Arrow* (if the stroke originated on or near a *GraphicalObject* inside the source page) or an *organizational Arrow* (if the stroke originated from an empty area or label of the source page). If the interpreter decides the arrow is a navigational arrow, it converts the source into a hyperlink by converting it into a *DenimHyperlinkInstance*. Within the *DenimHyperlinkInstance*, the interpreter associates a left-click event with the destination of the arrow.

The *LabelInterpreter* determines whether strokes added to the sheet should be treated as new page labels or be added to existing page labels. The *LabelInterpreter* contains a *ScribbledTextInterpreter* that determines whether adjacent strokes are part of the same word or phrase. Each *Label* contains exactly one *ScribbledText* object, so for two strokes to be contained in the same *Label*, they must belong to the same phrase. The *ScribbledTextInterpreter* uses a modified version of the algorithm proposed in (Chiu & Wilcox, 1998) for grouping ink strokes. It differs from Chiu and Wilcox's algorithm primarily in that it only takes into account spatial proximity when determining whether to group strokes together. Chiu and Wilcox's algorithm also considers temporal proximity because they are interested in grouping different input events (audio and ink) that do not have any spatial relationship to each other. Spatial proximity is sufficient for our domain.

ScribbledTextInterpreters are also attached to the *Patches* representing Web pages so that handwritten text inside the page will be grouped into *ScribbledText* objects that can then be operated on as single objects instead of collections of individual strokes. This means that, for example, a navigational *Arrow* originating from the printed word *Home* will treat the entire word as the source of the arrow as opposed to the individual stroke representing the letter *e*.

The *PanelInterpreter* determines whether a stroke is a rectangle representing a new Web page. Because Rubine's recognizer does not allow scale-independent gestures, the *PanelInterpreter* is actually implemented as two *Interpreters*, each configured to accept rectangles of a different size. The active interpreter is determined by the current zoom level, using SATIN's *SemanticZoomInterpreter* interface. New pages can be created by drawing rectangles only at the storyboard and page zoom levels. Using two interpreters in-

stead of one allows for improved recognition accuracy at each zoom level and reduces incorrect guesses.

Most of what users draw or write is rendered into the SATIN scenegraph without interpretation. Aside from aggregation into `ScribbledText`, strokes applied within the boundaries of a Web page are basically left alone. This is in keeping with the principles of informal UIs, which attempt to preserve users' intended ambiguity instead of imposing a formal interpretation on their input.

Rendering and Semantic Zooming in DENIM

Everything that is rendered to the screen in DENIM is descended from SATIN's *GraphicalObject*. We use SATIN's `SemanticZoomViews` for labels and pages so that, for example, labels do not appear at the overview zoom level and pages do not appear at the site map zoom level. Page Labels are rendered using SATIN's *StickyZViewWrapper*. Because Labels are rendered Sticky Z, they never appear to change size even though the view of their attached pages change size with the zoom level. This ensures that the labels are equally legible at all zoom levels.

As mentioned in the discussion of the `PanelInterpreter` earlier, SATIN provides the ability to apply different interpreters depending on the current zoom level. We also use this mechanism to change the way that editing gestures work with objects at different zoom levels. In the two broadest views, the overview and site map views, gestures work *shallowly*: You can select, move, or edit Web pages, but not anything inside of a Web page. Because these views focus on whole pages and the relationships between them, it follows that editing commands should operate on entire pages. In the two narrowest views, the page and detail views, gestures work *deeply*: You can select, move, or edit individual ink objects inside a Web page, but not Web pages themselves. These views focus on the contents of individual pages, so operations work on the page contents. The middle zoom view, the storyboard view, supports operations at both levels of detail. For example, the user taps and holds the page's label to select the page but can tap and hold any object inside a page to select that object.

Executing Designs in Run Mode

When the user opens his or her design in the Run window, one of the design's pages is cloned and added to the Sheet within the Run window, which is a subclass called *DenimRunSheet*. When users click within the Run window, the `DenimRunSheet` determines if they left-clicked on a `DenimHyperlinkInstance`. If they did, it tells the `DenimHyperlinkInstance` that it received a left-click event. The `DenimHyperlinkInstance` looks up which page is asso-

ciated with the left-click event and tells the DenimRunSheet to display that page.

DENIM's architecture is general enough to handle widget types and event types besides hyperlinks and left-clicks. For example, hyperlinks and text fields are actually subclasses of *DenimComponent*, and particular instances of hyperlinks and text fields in a DENIM design are subclasses of *DenimComponentInstance*. We will soon release a version of DENIM that allows designers to create their own DenimComponents and their own events. As we discuss in the Current and Future Work section, we use the generality of DenimComponents to add support for user-defined widgets (e.g., custom navigation bars) and reusable interaction sequences (e.g., a "checkout" sequence).

3.3. Evaluation and Usage Experience

To validate DENIM's design and implementation, we conducted an evaluation of the system. We were interested in gaining feedback about the usefulness of the functionality of the tool and the usability of the basic interactions, such as creating pages, creating links between pages, zooming, panning, and interacting with a design in run mode.

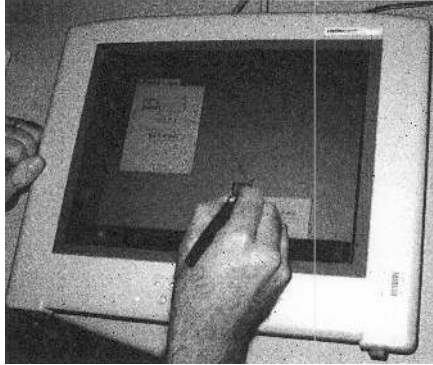
Description of Study

Seven professional designers participated in the study, one of whom had participated in the initial investigation into Web site design. Five of the participants said that Web site design projects constituted at least half of their current workload. The remaining two participants were a UI designer working on non-Web-related projects and a manager of a usability group for a large software company.

The system that we used for the evaluation consisted of an IBM ThinkPad 560Z (300 MHz Pentium II) laptop running Windows NT 4.0, and an ITI VisionMaker Sketch 14 display tablet (see Figure 27). The tablet had a 14-in. diagonal LCD screen and a pen (including a button) for input. The participants interacted primarily with the display tablet, although they could also use the keyboard for shortcuts.

The version of DENIM used for the evaluation was an early pre-release version that was available in August 1999. It was lacking some of the features of version 1.0, such as the ability to insert typed text, the ability to export to HTML, and support for complex widgets such as text fields. It also differed slightly in some aspects of the UI. For example, there were no functioning tools in the toolbox, and DENIM was always in "pencil" mode. Also, the textual labels now present on the Zoom Slider (designating the overview, site

Figure 27. This ITI VisionMaker Sketch 14 display tablet was used in the DENIM evaluation. It was connected to a 300-MHz Pentium II laptop running Windows NT 4.0 with Java 2 version 1.2 installed.

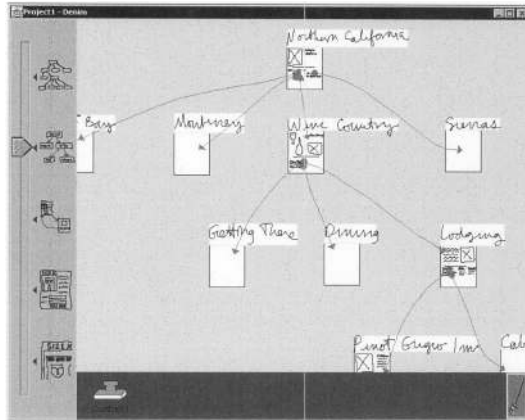


map, storyboard, page, and detail views) were not present, and neither was the button to bring up the pie menu. Finally, it was substantially inferior to version 1.0 in terms of performance and robustness. We opted to delay focusing on performance optimization and stability until after we had validated the desirability of the basic functions of the system through this evaluation. Throughout the following discussion of the study and our observations, we point out where specific observations led to improvements in the application.

One evaluation session was conducted per participant, and each evaluation session consisted of four parts. First, the participant was given a 10-min description and demo of DENIM, in which he or she was shown how to create pages, sketch their contents, connect pages with arrows, use the zoom slider to view the site at multiple levels, and interact with the site in run mode. Next, the participant was asked to add a few elements to a drawing in Microsoft Paint to become familiar with using the display tablet and pen. The second task was to get the participant used to interacting with DENIM. We loaded a previously created Web site design (shown in Figure 28) and asked the user to create a new page, link the page to the site, and then run through the site using run mode starting from the home page and ending at the page they just created.

The final part was a large design task, intended to be difficult to complete in the time allotted. We were interested in seeing how participants approached a realistic design task and how they used DENIM to help them. To motivate the participants to create the best design they could, we offered \$250 to the best design. The participant was asked to develop a Web site for a fictitious start-up company. The Web site was to help renters find places to rent and to help landlords find tenants. We provided an analysis of a competitor's

Figure 28. Each participant in the DENIM evaluation was given a 10-min demonstration of DENIM's major features. The demonstration showed them how to zoom to different levels, create pages, make links between pages, and interact with designs in run mode. This image shows the pre-release version of DENIM used during the evaluation, with the example Web site that was used for the demo. Note that the text labels are absent from the zoom slider, there is no button for invoking the pie menu, and there are no pencil and eraser tools (the cursor was always a pencil, and the tool in the toolbox was not functional).



Web site, market research on what renters and landlords said they wanted, and a description of what the client company required and desired. The participant had 45 to 60 min to come up with a preliminary site design, and then he or she presented the designs to us as if we were the rest of the design team.

While the participants performed the tasks, we paid attention to what types of actions they did (e.g., panning, drawing, and creating new pages) and at what zoom levels they performed those actions. This was to give us a sense of what features of DENIM they used and how well zooming supported the different design activities. We also recorded any critical incidents that occurred and their general comments and reactions.

After the participants were finished with the tasks, they filled out a questionnaire. We asked what they thought of DENIM in terms of usefulness, ease of use, and how they thought using it would affect their design process. The questionnaire also asked for basic demographics, primary job responsibilities, what tools they normally used, and how much Web design experience they had.

Observations

The size and complexity of the sites produced by designers varied somewhat: The smallest site design had five pages, whereas the largest had 29. Two

examples of designs created during the evaluation are presented in Figure 29. These examples highlight two distinct approaches to Web site design that were observed during the study. Some designers, like the author of the site depicted in Figure 29(a), took what might be termed an “architecture-oriented” approach, meaning that she brainstormed many of the pages that would be required for the site before sketching the contents of any one page. In fact, of the almost 30 pages she created during the session, only one of them contained anything besides a label. In contrast, the author of the site in Figure 29(b) took an “interaction-oriented” approach, meaning that he started with a specific interaction sequence—a potential tenant searching for an apartment—and sketched out in some detail the pages and links required to support that interaction. Of the eight pages he created during the session, six of them had at least some of their contents sketched. The two designers differed in their use of arrows and run mode as well. The architecture-oriented designer used organizational arrows almost exclusively, whereas the interaction-oriented designer did not use them at all. At the same time, the interaction-oriented designer made heavy use of run mode, previewing the interaction with the site several times, whereas the architecture-oriented designer did not attempt run mode until the very end of the session. We were encouraged to observe that both designers felt strongly that DENIM supported their respective design processes well.

All of the participants made substantial use of different zoom levels, with usage concentrated primarily in the middle three levels (site map, storyboard, and page). Several users verbally expressed that they liked the concept of the different zoom levels and liked the ability to maintain a unified representation of the site, while interacting with it at different levels of detail. It appears that users felt that the integrated view would help them iterate more quickly through different design ideas. One user highlighted the advantages of the integrated view by observing,

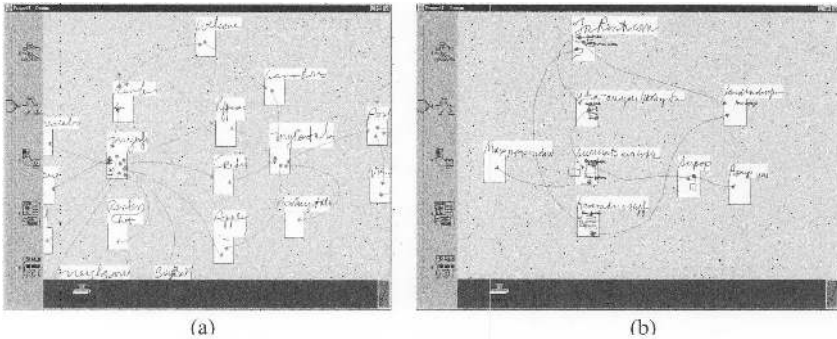
It’s not like “OK, that’s one idea,” then open a new file and work on a new [idea]. You don’t need to do that. The iteration goes on within this [tool] and I can see the relationships.

Another user described how she thought DENIM would improve her current process by remarking,

I usually [create site maps] in PowerPoint, then I go back to the navigational flow, then I go back to PowerPoint ... And here it would be so easy to do that iterative kind of thing.

However, the integration of these views through zooming sometimes proved to be problematic. Several of the users became frustrated navigating

Figure 29. These two Web sites, produced by participants in the DENIM evaluation, demonstrate DENIM's ability to support different design styles: (a) "architecture-oriented" (i.e., lots of pages with little or no detail and mostly organizational arrows) and (b) "interaction-oriented" (i.e., few pages, most of which contain detailed sketches of their contents, and mostly navigational arrows).



around their site designs and found that they often had to zoom out to a higher level to find their desired target and then zoom back in on that target. This has been previously reported as a problem with zooming UIs (Jul & Furnas, 1998).

Likewise, users had trouble creating navigational arrows between pages that they had initially drawn far apart on the canvas. It was difficult to find a view of the site that would include both the source and target page, yet have enough detail to be able to find the specific object on the source page that they wished to serve as the link anchor.

In response to these issues, we made two changes to DENIM, both of which were integrated before the release of version 1.0. We introduced auto-panning, which pans the screen when the user draws a line toward a side of the screen. This makes it easier to link two pages that are not visible at the same time. A user can start drawing from one page and draw until he or she sees the desired page. We also changed the display of pages at the site map level so that only their labels appear. The early version of the display in site map view is shown in Figure 28. The modified site map view that appears in version 1.0 (shown in Figure 12) encourages users to draw their initial site maps more densely, because the total size of each page is less. The resulting density of pages makes it more likely that the source and target page will be visible on the screen at the same time in the storyboard view.

In future versions, we also plan to explore focus + context techniques (Furnas, 1986) to address the navigation and linking problems we observed. Seeing more of the site in the periphery while zoomed in to a particular portion of the site could help reduce the difficulty of finding one's place in the

site. Similarly, compressing the distance between a source and target page, while maintaining a high level of detail in the source page, would help relieve the problem of linking pages that were originally drawn far apart from each other in the site map.

Users appreciated the informal mode of interaction provided by DENIM. One user compared the interaction to other tools with the following comment:

You draw a box in Illustrator or Freehand or Quark, and it's got attributes that have to be dealt with, and it interrupts the thought process. ... It's nice to be able to get rid of all the business with the pictures and all the definite object attributes. That is such a hassle.

At the same time, the free-form sketching interface provided some stumbling blocks. For example, handwriting on the screen was difficult, given the average performance of the application, the simple stroking algorithm used, and the lack of feedback from writing on a smooth screen. Also, printing (as opposed to cursive writing) was difficult because the automatic word-grouping algorithm was not very good. Two users experienced difficulty reading their own page labels. Another user wanted to type her page labels. Other users said that they like to handwrite while brainstorming, but would like the ability to replace handwritten labels with typed labels as their ideas become solidified. As described earlier, we added support for typed text in version 1.0. We also improved the performance of the application as well as the algorithms for capturing and displaying pen strokes and for grouping words and phrases.

Feedback

The responses to the posttest questionnaire, though informal, were instructive in several ways. Opinions about DENIM's perceived effect on the respondent's work practices were sharply divided based on the amount of the respondent's workload that consisted of Web design projects. The two individuals not involved in Web design ranked DENIM relatively low on factors such as "the perceived benefit using the tool would have on their ability to communicate with team members" and on "DENIM's overall usefulness" to them. The five Web designers, on the other hand, had generally positive opinions of DENIM along these lines.

Though the Web designers ranked the ease of use just above average (6.4 out of 10), they ranked the usefulness fairly high (9.0 out of 10). This seems to indicate that, despite the shortcomings of the current implementation in terms of performance and fluid interaction, users felt that the basic concepts were on target.

Also, the Web designers gave very high rankings when asked to rate DENIM according to its perceived ability to communicate with others involved in the design process. Those users rated DENIM better than 8.5 out of 10 in terms of ability to communicate with design team members (8.6), internal managers (8.8), and usability engineers and testers (8.8). They also gave similarly high marks to DENIM's improvement in their ability to express their ideas (9.0), iterate quickly through versions of a design (8.6), and overall efficiency (8.6). All users gave DENIM relatively low marks in terms of ability to communicate with clients (6.1 out of 10 overall), which we attribute largely to DENIM's inability to produce "cleaned-up" versions of sketches that would be acceptable to show to clients.

Other Usage Experience

After the initial evaluation, we promptly turned our efforts toward developing a robust and responsive version of DENIM that could be deployed to designers in the field and used over a long period of time. We felt that most of the core features were validated by the initial evaluation and we believed that a deployment in the field, followed by one or more longitudinal studies of DENIM being used in real settings, by real designers, on real projects, would reveal a host of other, more interesting issues. The effort to create a satisfactory version of DENIM for field use continues to this day, though feedback from our considerable pool of users seems to be indicating that we may well be approaching our goal.

Between DENIM's initial release on the Web (May 6, 2000) and the time this article went to press (June 17, 2003), it has been downloaded 31,944 times, with 27,615 of those downloads occurring since the release of version 1.0 on November 21, 2001. Whereas some portion of those downloads are probably by people merely investigating the tool out of curiosity, it is clear from e-mail feedback that at least some people are using DENIM to work on actual designs. Through this channel, we have received requests for increased support for widgets, support for exporting to HTML (earlier releases did not have this capability), and a version that works on Mac OS. We also received a number of messages expressing positive feedback and validating the existence of a need for a tool like DENIM.

DENIM has also been used in several UI design classes at UC Berkeley for low-fidelity prototyping assignments. The experience of the students in these classes again highlighted the need for increased widget support, as well as the need to improve the performance and robustness.

Finally, DENIM was used to carry out a study comparing users' reactions to formal and informal representations of Web sites (Hong, Li, Lin, & Landay, 2001). The experience of the authors of the study turned up additional prob-

lems, such as flaws in the implementation of Copy, Paste, Undo, and Redo; the difficulty of merging different files together; the need to improve the aesthetic appearance of drawn ink (through anti-aliasing); and (once again) the need to improve the performance. Thanks to these experiences, we were able to substantially improve DENIM in each of these areas before the release of version 1.0.

3.4. Discussion

DENIM supports many of the implications for a Web design tool enumerated at the end of Section 2, but some are not supported as of version 1.0. We said that informal tools are appropriate for the early stages of design, that tools focusing on the early stages should integrate well with more formal representations and the tools that produce them, that such tools should focus on the production of intermediate artifacts, that multiple variations should be supported, and that integration with paper should be supported. Let us look at DENIM in the light of these implications.

The sketch-based interface to DENIM supports an informal mode of interaction. This means that the tool should fit well into designers' current practice during the early phases of design. Currently, designers sketch extensively while initially exploring a design, and this mode of expression is explicitly supported by DENIM. Our evaluation with professional designers suggests that designers believe DENIM will, in fact, fit in well with their current practices. In addition to supporting current sketching practices, DENIM offers the advantages of electronic media, such as facilitating replication, incremental modification, testing, and distribution.

We also said that we believed that an informal tool would be easier to use and learn because it employs modes of interaction and communication that are familiar to users. In our evaluation, it was found that DENIM, though fairly easy to learn and understand, was not particularly easy to use. We do not know whether the ease of use problems stemmed from shortcomings of the hardware, the implementation, or from problems inherent in the design, but we believe at least some of the blame lies with the hardware. In particular, it appeared that the awkwardness of hand printing on the tablet, especially for users unfamiliar with the input device, caused serious difficulties. The poor performance of the application exacerbated this problem. On the other hand, problems with linking distant pages (inherent in our design) also caused serious problems. We believe our changes to the performance and the design have helped to resolve these issues, but we have not formally evaluated our improvements. We also believe that the introduction of the Microsoft Tablet PC platform will drive improvements in the hardware as more vendors strive to develop pen-based devices, and that it will also lead to wider familiarity with this type of interface.

DENIM focuses on and supports the tasks of the early phases of design, that is, the discovery and design exploration phases, when informal modes of interaction and communication are most appropriate. Information and navigation design are commonly the most active types of design during these early phases, and these are the types of design supported by DENIM. We support information and navigation design by facilitating the creation of associated artifacts and visualizations, namely site maps, storyboards, and schematics. Unlike current tools used by Web site designers, DENIM ties together these different representations into a unified framework. Designers who participated in the evaluation felt that this was a significant improvement over their current tools and practices.

Some observers have noted that the “storyboard” label for the middle zoom level might be misleading. This zoom level offers some of the features of conventional storyboards, such as the ability to view multiple pages at medium detail and to see the navigational interrelationships among those pages. On the other hand, this view does not support other characteristics of storyboards that were present on some designers’ sketches observed during the Web design study. For example, one cannot easily focus on a single thread of interaction that connects pages distant from one another in the site map. This is partially supported in a soon-to-be-released scenario-recording feature, which will allow a designer to capture and view threads of interaction (see Section 3.5), but there is still no visualization available in DENIM that precisely parallels the storyboards observed in the study. The text labels on the zoom slider were introduced before the first beta release but after the usability evaluation was conducted, as we found it difficult to write documentation without having visible textual labels as reference points. Because the labels were not present during the usability evaluation, we do not have data about whether this label is confusing, but it has not come up among our user population.

Though it is appropriate that DENIM focuses on the intermediate artifacts produced early in the design process, the current implementation does not support integration with other tools or transitions to more formal representations. This means that DENIM does not meet the need to integrate with later design phases, nor does it support an important task faced by designers in the early phases of design: presenting designs to clients. Because designers were clear that they did not wish to show sketches to clients, an early stage tool like DENIM should assist designers in producing cleaned-up versions of sketches for presentation. Otherwise, designers will turn to other tools to create presentable versions of their designs, and as with the transition from paper sketches to more formal representations, it is unlikely that designers would then return to the informal, “sketchy” representations.

Ideally, we would like DENIM to support the creation of “presentable” versions of design ideas while still allowing the designer to work with sketchy

versions of unfinished ideas. Allowing the creation of multiple variations of sketches and sites could allow both formal and informal representations of the same elements to exist side by side and remain associated with each other.

Alternatively, we would hope that a tool like DENIM might help persuade designers to push back against the trend toward premature formalization that many of the designers we interviewed complained about but felt helpless to combat. Designers know, and studies have shown (Black, 1990), that overly formal design representations cause viewers to focus on details that are distracting or irrelevant to the main issues that the representation is supposed to address.

In general, variations are important during the design exploration phase. Currently, DENIM does not support the generation, comparison, and management of multiple variations, other than through saving files explicitly, but this is a feature we plan to support in a future version.

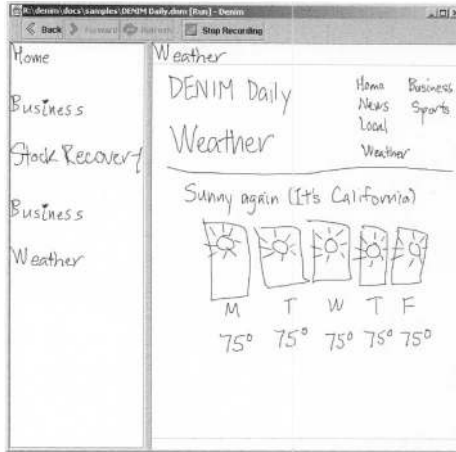
Finally, we said that even a sketch-based tool could not replace paper completely. Paper is still unrivaled for its inexpensiveness, portability, and feel. At present, DENIM does not support integration with paper, either as input or output. We have experimented with integrating paper for input by allowing sketches done on a CrossPad to be associated with a DENIM Web page. Another approach would be to allow scanned sketches to be imported into DENIM pages. Allowing printouts of both sketches and formalized representations is also important.

3.5. Current and Future Work

Our intent in releasing DENIM on the Web for free download was to allow designers to begin using it in their projects. We hoped that this real-world usage would provide us with feedback on how well DENIM supports designers' work. As mentioned before, there have been several thousand downloads of DENIM, and we have received a lot of feedback about its strengths and weaknesses. However, we are currently planning a more formal survey of the people that have downloaded DENIM to elicit feedback on its effectiveness and utility, as well as longitudinal studies of DENIM being used in the field.

We have designed and partially implemented the concept of scenarios, which we will add to a future version of DENIM. Scenarios will allow designers to create a "slide show," consisting of a sequence of pages the designer believes a user will visit to accomplish a particular task. To create a scenario, the designer will first go to run mode and press the Record Scenario button in the Run window's toolbar. The designer will then add pages to the scenario by clicking on the links that bring up those pages. The labels of the pages will be added to a list along the left side of the Run window. When the designer is done, he or she will click on the Stop Recording button in the Run window's toolbar (see Figure 30). The designer will then be able to name the scenario in a dialog box.

Figure 30. Using the scenarios feature, designers will be able to record sequences of page transitions. The list of page labels down the left-hand side indicates the sequence of pages visited so far in this scenario.



The designer will then be able to playback a scenario by opening up the pie menu, choosing View→Scenarios, choosing a scenario, and clicking Play. A Run window with the first page of the scenario will open. Clicking anywhere within the page will bring up the next page in the scenario.

We have also developed a more advanced visual language for DENIM (Lin, Thomsen, & Landay, 2002). One part of this visual language lets designers specify their own re-usable components. These components can be as simple as a new kind of widget or as complex as a template for a Web page. An example in which this is useful is with navigation bars. Instead of redrawing the same navigation bar for each Web page, a designer can factor out common behavior into a navigation bar component. This visual language also supports conditional behaviors as well as a wider range of events beyond just left mouse click, such as right clicks, double clicks, timeouts, and rollovers. The goal here is to extend DENIM so that it can also be used to design more sophisticated Web interaction as well as standard GUIs.

A closely related project our research group has developed is The Designer's Outpost (Klemmer, Newman, Farrell, Bilezikjian, & Landay, 2001), a tangible UI that combines the affordances of paper, large physical workspaces, and electronic media to support information design of Web sites. Designers interact with the system by writing on physical Post-it Notes, arranging them on an electronic whiteboard in related groups, and drawing links between them. The system tracks the Post-its using computer vision and captures links among Post-its and groups with an electronic stylus. Site maps created in Outpost can be ex-

ported to DENIM, where they can be evolved further. Outpost also supports the creation and use of design histories (Klemmer, Thomsen, Phelps-Goodman, Lee, & Landay, 2002) and has been extended to support remote collaboration (Everitt, Klemmer, Lee, & Landay, 2003).

Another closely related project our research group has developed is WebQuilt (Hong, Heer, Waterson, & Landay, 2001; Hong & Landay, 2001), a tool for logging and visualizing Web usage. The overall goal is to help designers see problems that people encounter in a Web site. Logging is done through a proxy, overcoming many of the problems with server-side and client-side logging. Captured usage traces can be aggregated and visualized in a zooming interface that shows the Web pages people viewed. The visualization also shows the most common paths taken through the Web site for a given task, as well as the optimal path for that task as designated by the designer. We have also created more sophisticated visualizations and layouts for organizing, filtering, and drilling down into the collected data (Waterson et al., 2002).

We are also looking into ways to support the generation of medium-fidelity prototypes from low-fidelity sketches. As noted before, such a feature could allow designers to give more “professional” presentations, while staying with sketching longer. Presentations themselves could be generated directly from DENIM as well. We are considering extending the scenario recording and playback mechanism to support exporting of scenarios as PowerPoint (Microsoft, 1987) presentations containing walkthroughs.

We would like DENIM to work with existing Web design tools so as to fit more naturally into the entire Web design cycle. This includes generating HTML and other artifacts that can be imported by other tools. Ideally, DENIM would also be able to import files from other tools, so that designers can smoothly move back and forth in the design process. We are currently working on adding support for Scalable Vector Graphics (SVG) into DENIM, which would allow it to interoperate with tools such as Adobe Illustrator.

To make DENIM scale for large Web sites, we plan to explore additional visualizations and interactions that operate on higher levels of abstraction. Note that both of the site maps in Figures 2 and 5 use lines and shading to show certain groupings of pages. DENIM should support this aspect of the site map visualizations. It would also be desirable to allow the designer to identify subsites and collapse and expand their representation. The overview zoom level in particular could be used to support interactions with the overall site structure.

4. RELATED WORK

Several work practice studies have appeared in the literature that are especially relevant to our study of Web site designers. Sumner and Stolze’s study

of speech application designers (Sumner & Stolze, 1997) and Bellotti and Rogers' study of editorial staff at several publishing companies (Bellotti & Rogers, 1997) showed that designers and editors use multiple intermediate representations of products during their creation, some of them similar to the representations found in this study.

A certain amount can be learned about Web design practice by reading the growing body of literature that covers it (Fleming, 1998; Nielsen, 1999; Rosenfeld & Morville, 1998; Sano, 1996). Unfortunately, much of this literature is prescriptive rather than descriptive in nature and may not accurately reflect what designers are actually doing in the field. To learn what designers do, there is no substitute for direct contact. We elected to conduct a first-hand investigation into Web design practice through field visits and interviews with professional designers. Our approach was inspired by the methods proposed in, for example, Blomberg, Giacomo, Mosher, and Swenton-Wall (1993); and Hughes, King, Rodden, and Andersen (1995).

Sketching and writing are natural activities used by many designers as part of the design process. DENIM captures this activity with an informal ink-based interface (Landay & Myers, 2001). Using an informal interface is a key aspect of DENIM, as it allows designers to defer the details until later and focus on their task without having to worry about precision. Many research systems have taken this direction in recent years, either by not processing the ink (Davis et al., 1999; Schilit, Golovchinsky, & Price, 1998; van de Kant, Wilson, Bekker, Johnson, & Johnson, 1998) or by processing the ink internally while displaying the unprocessed ink (Gross & Do, 1996; Landay & Myers, 2001; Moran, Chiu, & van Melle, 1997; Saund & Moran, 1994).

DENIM is most closely related to SILK (Landay & Myers, 1995, 1996, 2001), a sketch-based UI prototyping tool. Using SILK, individual interface screens can be drawn, with certain sketches recognized as interface widgets. These screens can be linked to form storyboards (Landay & Myers, 1996), which can be tested in a run mode. DENIM takes many of these ideas and extends them to the domain of Web site design. However, DENIM de-emphasizes the screen layout aspects of SILK, focusing instead on the creation of whole Web sites. Furthermore, instead of the separate screen and storyboard views in SILK, all of the views are integrated through zooming. Also, SILK attempts to recognize the user's sketches and display its interpretation as soon as possible. Seeing the results of the eager recognizer's interpretations can be disruptive of the "flow state" (Csikszentmihaly, 1990) many designers identify as one of the major advantages of sketching (Edwards, 1989). For example, when the user's input is incorrectly interpreted, it is very difficult for most users to ignore the incorrect guess and continue sketching. The user will usually stop sketching to correct the application. DENIM intentionally avoids doing much recognition to support more free-form sketching.

DENIM's use of storyboarding to illustrate behaviors is similar to SILK. Other systems that use storyboarding include Anecdote (Harada, Tanaka, Ogawa, & Hara, 1996) and PatchWork (van de Kant et al., 1998).

WebStyler (Hearst, Gross, Landay, & Stahovich, 1998) is another sketch-based tool for prototyping individual Web pages. However, DENIM addresses more aspects of Web site design, including designing the site structure and being able to interact with the sketches.

Others have noted that designers often sketch basic designs in the early phases of solving a design problem (Wagner, 1990; Wong, 1992). Sketching has many advantages over traditional UI design tools that focus on creation of high-fidelity prototypes. Sketches are inherently ambiguous, which allows the designer to focus on basic structural issues instead of unimportant details (Black, 1990). The ambiguity also allows multiple interpretations of the sketch, which can lead to more design ideas (Goel, 1995). Sketching is quick, so designers can rapidly explore different ideas, which leads to a more thorough exploration of the design space. Rapid sketching also encourages iteration, which is widely considered to be a valuable technique for designing interfaces (Gould & Lewis, 1985).

There is a lack of early-stage prototyping tools for the Web. Our ethnographic study showed us that Web designers use other tools to fill this gap. Macromedia Director (Macromedia, 1989) is often used to assemble storyboards, whereas Visio (Microsoft, 1992) is used for modeling the high-level information architecture of a Web site. However, Director is a multimedia-authoring tool, and Visio is a general purpose diagramming tool. This makes using them for such high-level Web site design awkward at best, because they are not designed for those tasks.

Currently, the most popular tools for creating Web sites include Microsoft FrontPage (Microsoft, 1995), Adobe GoLive (Adobe, 1997), Macromedia Dreamweaver (Macromedia, 1997), and NetObjects Fusion (NetObjects, 1996). These tools focus on designing page layout rather than the site architecture. Admittedly, each of them has a "site structure view" of a Web site. However, this view often constrains any edits so that the tree structure remains intact. Furthermore, the site structure view and the page layout view are not as closely integrated as they are in DENIM. Most importantly, these tools focus on producing high-fidelity representations, which is inappropriate in the early stages of design. These are all important issues that we chose to address in DENIM.

5. CONCLUSION

The development of DENIM was motivated and guided by an ethnographic study of Web designers that provided insight into design practice. The study showed that designers commonly sketch on paper to explore

and communicate early design ideas. It also showed that designers employ multiple representations of sites at different levels of detail throughout the design process. Moreover, it suggested that there was little or no support for early stage information and navigation design in existing tools. We relied on these observations in our design and development of DENIM.

DENIM is an informal sketch-based tool that supports information and navigation design for Web sites. The focus of DENIM is on the artifacts and practices associated with the design exploration phase of the design process. Different site views are integrated through zooming, with special emphasis placed on the most commonly observed site visualizations, that is, site map, storyboard, and individual page views. Designers can use DENIM to sketch out Web pages and site structures, and can then interact with their site designs in *run mode*. Thus, designers can get a rough sense of the user's experience using a potential site design with a minimum of investment in creating the design. An informal evaluation of DENIM's features and UI indicates that DENIM will be useful to designers. Designers who participated in the study believed that DENIM would improve their ability to communicate with team members, their expressiveness, and their efficiency. Based on observations made during the study, as well as feedback from a substantial number of people who have downloaded and used DENIM since its release on the Web in May 2000, we have implemented a number of new features and improvements.

We believe DENIM fills a gap that currently exists in Web site designers' tool belts. In particular, it fills a gap currently present in the early design phases. We plan to further improve the usefulness of DENIM by extending it to integrate with other tools and with the practices and concerns of later design phases.

DENIM is available for download at <http://guir.berkeley.edu/denim>.

NOTES

Background. This article is based on work carried out by the Group for User Interface Research at UC Berkeley. Mark Newman has subsequently taken a position at PARC.

Acknowledgments. Liane Beckman, Michael Bina, Lisa Chan, Eric Chung, Andrew Cuneo, Carol Hu, Peter Khooshabeh, John Brian Kirby, Jenny Lee, Robert Lee, William Lee, Boxin Li, Yang Li, Benson Limketkai, Nahush Mahajan, Michael Pow, Marc Ringuette, Lifeng Shelley Shen, Quinn Solomon, Orna Tarshish, Michael Thomsen, Eric Tse, and Juan Valencia all made valuable contributions to various versions of DENIM. Francis Li and Kalpana Joshi contributed to an early version of DENIM called SILKWeb. Chris Long's gdt tool (now called Quill), which includes a Java implementation of Rubine's Recognizer, and his expertise in gesture set design were also very helpful. Marti Hearst, Scott Klemmer, and Anoop Sinha provided useful feedback on earlier versions of the material presented in this article. Several images of design artifacts are re-

printed courtesy of Aaron Marcus and Associates (AM+A), Emeryville, California. We wish to thank AM+A for its time and these images, and we want to thank all of the designers who took time out of their busy schedules to participate in either the Web design investigation or the DENIM evaluation or, in one case, both.

Support. NEC USA and Qualcomm provided some of the funds that supported this work.

Authors' Present Addresses. Mark W. Newman, PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304. E-mail: mnewman@parc.com. James Lin, Group for User Interface Research, Computer Science Division, Electrical Engineering and Computer Sciences Department, UC Berkeley, Berkeley, CA 94720. E-mail: jimlin@cs.berkeley.edu. Jason I. Hong, Group for User Interface Research, Computer Science Division, Electrical Engineering and Computer Sciences Department, UC Berkeley, Berkeley, CA 94720. E-mail: jasonh@cs.berkeley.edu. James A. Landay, Group for User Interface Research, Computer Science Division, Electrical Engineering and Computer Sciences Department, UC Berkeley, Berkeley, CA 94720. E-mail: landay@cs.berkeley.edu.

HCI Editorial Record. First manuscript received May 1, 2002. Final manuscript received March 6, 2003. Accepted by Terry Winograd. — *Editor*

REFERENCES

- Adobe. (1987). *Illustrator*: Author.
- Adobe. (1989). *Photoshop*: Author.
- Adobe. (1997). *GoLive*: Author.
- Bare Bones Software. (1993). *BBEdit*: Author.
- Bederson, B. B., & Hollan, J. D. (1994). Pad++: A zooming graphical interface for exploring alternative interface physics. *Proceedings of UIST '94, The ACM Symposium on User Interface Software and Technology* (pp. 17–26). New York: ACM.
- Bederson, B. B., Hollan, J. D., Druin, A., Stewart, J., Rogers, D., & Proft, D. (1996). Local tools: An alternative to tool palettes. *Proceedings of UIST '96, The ACM Symposium on User Interface Software and Technology* (pp. 169–170). New York: ACM.
- Bellotti, V., & Rogers, Y. (1997). From Web press to Web pressure: Multimedia representations and multimedia publishing. *Proceedings of CHI '97, The ACM Conference on Human Factors in Computing Systems* (pp. 279–286). New York: ACM.
- Beyer, H., & Holtzblatt, K. (1998). *Contextual design: Defining customer-centered systems*. San Francisco: Morgan Kaufmann.
- Black, A. (1990). Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers. *Behaviour & Information Technology*, 9(4), 283–296.
- Blomberg, J., Giacomi, J., Mosher, A., & Swenton-Wall, P. (1993). Ethnographic field methods and their relation to design. In D. Schuter & A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 123–155). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.
- Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988). An empirical comparison of pie vs. linear menus. *Proceedings of CHI '88, The ACM Conference on Human Factors in Computing Systems* (pp. 95–100). New York: ACM.

- Cataldo, J. W. (1966). *Graphic design & visual communication*. Scranton, PA: International Textbook Co.
- Chiu, P., & Wilcox, L. (1998). A dynamic grouping technique for ink and audio notes. *Proceedings of UIST '98, The ACM Symposium on User Interface Software and Technology* (pp. 195–202). New York: ACM.
- Crampton Smith, G., & Tabor, P. (1996). The role of the artist-designer. In T. Winograd (Ed.), *Bringing design to software* (pp. 37–57). New York: ACM Press.
- Cross Pen Computing Group. (1998). *CrossPad* [Web page]. A. T. Cross Company. Retrieved January 20, 1999, from <http://www.cross-pcg.com/cross/crosspads.html>
- Csikszentmihaly, M. (1990). *Flow: The psychology of optimal experience*. New York: Harper Collins.
- Davis, R. C., Landay, J. A., Chen, V., Huang, J., Lee, R. B., Li, F. C., et al. (1999). NotePals: Lightweight note sharing by the group, for the group. *Proceedings of CHI '99, The ACM Conference on Human Factors in Computing Systems* (pp. 338–345). New York: ACM.
- Edwards, B. (1989). *Drawing on the right side of the brain*. New York: Putnam.
- Erickson, T. (1995). Notes on design practice: Stories and prototypes as catalysts for communication. In J. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 37–59). New York: Wiley.
- 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 2003, The ACM Conference on Human Factors in Computing Systems, CHI Letters* (to appear).
- Evolve Design. (1999). *Evolve process* [Web Page]. Evolve Design. Retrieved January 20, 1999, from <http://www.evolved.com/process.html>
- Fire Engine Red. (1999). *Fire engine red: process: our process* [Web Page]. Fire Engine Red. Retrieved January 20, 1999, from http://www.engined.com/our_process.html
- Fleming, J. (1998). *Web navigation: Designing the user experience*. Sebastopol, CA: O'Reilly.
- Furnas, G. W. (1986). Generalized fish-eye views. *Proceedings of CHI '86, The ACM Conference on Human Factors in Computing Systems* (pp. 16–23). New York: ACM.
- Goel, V. (1995). *Sketches of thought*. Cambridge, MA: MIT.
- Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, 28, 300–311.
- Gross, M. D., & Do, E. Y.-L. (1996). Ambiguous intentions: A paper-like interface for creative design. *Proceedings of UIST '96, The ACM Symposium on User Interface Software and Technology* (pp. 183–192). New York: ACM.
- Harada, K., Tanaka, E., Ogawa, R., & Hara, Y. (1996). Anecdote: A multimedia storyboarding system with seamless authoring support. *Proceedings of Multimedia '96, The ACM International Multimedia Conference* (pp. 341–351). New York: ACM.
- Hearst, M. A., Gross, M. D., Landay, J. A., & Stahovich, T. E. (1998). Sketching intelligent systems. *IEEE Intelligent Systems*, 13(3), 10–19.
- Hong, J., & Landay, J. A. (2000). SATIN: A toolkit for informal in-based applications. *UIST 2000, The ACM Symposium on User Interfaces and Software Technology, CHI Letters*, 2(2), 63–72.

- Hong, J., & Landay, J. A. (2001). WebQuilt: A framework for capturing and visualizing the web experience. *Proceedings of The Tenth International World Wide Web Conference*.
- Hong, J. I., Heer, J., Waterson, S., & Landay, J. A. (2001). WebQuilt: A proxy-based approach to remote Web usability testing. *ACM Transactions on Information Systems*, 19(3), 263–285.
- Hong, J. I., Li, F. C., Lin, J., & Landay, J. A. (2001). End-user perceptions of formal and informal representations of Web sites. *Proceedings of CHI 2001, The ACM Conference on Human Factors in Computing Systems, Extended Abstracts* (pp. 385–386). New York: ACM.
- Hughes, J., King, V., Rodden, T., & Andersen, H. (1995). The role of ethnography in interactive systems design. *interactions*, 2(2), 56–65.
- Jul, S., & Furnas, G. W. (1998). Critical zones in desert fog: Aids to multiscale navigation. *Proceedings of UIST '98, The ACM Symposium on User Interface Software and Technology* (pp. 97–106). New York: ACM.
- Klemmer, S. R., Newman, M. W., Farrell, R., Bilezikjian, M., & Landay, J. A. (2001). The designers' outpost: A tangible interface for collaborative Web design. *Proceedings of UIST 2001, The ACM Symposium on User Interface Software and Technology, CHI Letters*, 3(2), 1–10.
- 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 2000, The ACM Symposium on User Interfaces and Software Technology, CHI Letters*, 2(2), 1–10.
- 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 2002, ACM Conference on Human Factors in Computing Systems, CHI Letters*, 4(1), 1–8.
- Landay, J. A., & Davis, R. C. (1999). Making sharing pervasive: Ubiquitous computing for shared note taking. *IBM Systems Journal*, 38(4), 531–550.
- Landay, J. A., & Myers, B. A. (1995). Interactive sketching for the early stages of user interface design. *Proceedings of CHI '95, The ACM Conference on Human Factors in Computing Systems* (pp. 43–50). New York: ACM.
- Landay, J. A., & Myers, B. A. (1996). Sketching storyboards to illustrate interface behavior. *Proceedings of CHI '96, The ACM Conference on Human Factors in Computing Systems, Conference Companion* (pp. 193–194). New York: ACM.
- Landay, J. A., & Myers, B. A. (2001). Sketching interfaces: Toward more human interface design. *IEEE Computer*, 34(3), 56–64.
- Lawson, B. (1994). *Design in mind*. Oxford, England: Butterworth Architecture.
- Lewis, C., & Rieman, J. (1993). *Task-centered user interface design: A practical introduction* [Shareware book]. University of Colorado. Retrieved December 1999, from <ftp://ftp.cs.colorado.edu/pub/cs/distrib/clewis/HCI-Design-Book/>
- 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 2000, The ACM Conference on Human Factors in Computing Systems, CHI Letters*, 2(1), 510–517.
- Lin, J., Thomsen, M., & Landay, J. A. (2002). A visual language for sketching large and complex interactive designs. *Proceedings of CHI 2002, The ACM Conference on Human Factors in Computing Systems, CHI Letters*, 4(1), 307–314.

- Long, A. C., Jr., Landay, J. A., & Rowe, L. A. (1999). Implications for a gesture design tool. *Proceedings of CHI '99, The ACM Conference on Human Factors in Computing Systems* (pp. 40–47). New York: ACM.
- Love, S. F. (1980). *Planning and creating successful engineered designs*. Los Angeles: Advanced Professional Development.
- Macromedia. (1989). Director. San Francisco, CA: Author.
- Macromedia. (1997). Dreamweaver. San Francisco, CA: Author.
- Microsoft. (1983). Word. Redmond, WA: Author.
- Microsoft. (1987). PowerPoint. Redmond, WA: Author.
- Microsoft. (1992). Visio. Redmond, WA: Author.
- Microsoft. (1995). FrontPage. Redmond, WA: Author.
- Microsoft. (2002). Windows XP Tablet PC Edition. Redmond, WA: Author.
- Moran, T. P., Chiu, P., & van Melle, W. (1997). Pen-based interaction techniques for organizing material on an electronic whiteboard. *Proceedings of UIST '97, The ACM Symposium on User Interface Software and Technology* (pp. 45–54). New York: ACM.
- NetObjects. (1996). Fusion. Redwood City, CA: Author.
- Netscape. (1994). Netscape Navigator. Mountain View, CA: Author.
- Newman, M. W., & Landay, J. A. (2000). Sitemaps, storyboards, and specifications: A sketch of Web site design practice. *Proceedings of DIS 2000, The ACM Symposium on Designing Interactive Systems* (pp. 263–274). New York: ACM.
- Nielsen, J. (1999). *Designing Web usability: The practice of simplicity*. Indianapolis, IN: New Riders.
- Rettig, M. (1994). Prototyping for tiny fingers. *Communications of the ACM*, 37(4), 21–27.
- Robbins, E. (1994). *Why architects draw*. Cambridge, MA: MIT.
- Rosenfeld, L., & Morville, P. (1998). *Information architecture for the World Wide Web*. Sebastopol, CA: O'Reilly.
- Rowe, P. G. (1987). *Design thinking*. Cambridge, MA: MIT.
- Rubine, D. (1991). Specifying gestures by example. *Computer Graphics: ACM SIGGRAPH*, 25(4), 329–337.
- Sano, D. (1996). *Designing large-scale Web sites: A visual design methodology*. New York: Wiley.
- Saund, E., & Moran, T. P. (1994). A perceptually-supported sketch editor. *Proceedings of UIST '94, The ACM Symposium on User Interface Software and Technology* (pp. 175–184). New York: ACM.
- Schilit, B. N., Golovchinsky, G., & Price, M. N. (1998). Beyond paper: Supporting active reading with free form digital ink annotations. *Proceedings of CHI '98, The ACM Conference on Human Factors in Computing Systems* (pp. 249–256). New York: ACM.
- Shedroff, N. (1994). *Information interaction design: A unified field theory of design* [Web page]. Retrieved May 1, 1999, from <http://www.nathan.com/thoughts/unified/index.html>
- Shneiderman, B. (1997). *Designing the user interface: Strategies for effective human-computer interaction* (3rd ed.). Boston: Addison-Wesley.
- Strothotte, T., Preim, B., Raab, A., Schumann, J., & Forsey, D. R. (1994). How to render frames and influence people. *Computer Graphics Forum, Eurographics '94 Conference Issue*, 13(3), 455–466.

- Studio Archetype. (1998). *Studio archetype: Process* [Web Page]. Studio Archetype. Retrieved January 20, 1999, from <http://www.studioarchetype.com/process/quickview.html>
- Sumner, T., & Stolze, M. (1997). Evolution, not revolution: Participatory design in the toolbelt era. In M. Kyng & L. Mathiassen (Eds.), *Computers and Design in Context* (pp. 1–26). Cambridge, MA: MIT.
- van de Kant, M., Wilson, S., Bekker, M., Johnson, H., & Johnson, P. (1998). Patch-Work: A software tool for early design. *Proceedings of CHI '98, The ACM Conference on Human Factors in Computing Systems, Conference Summary* (pp. 221–222). New York: ACM.
- Wacom Technology Company. (2000). *Intuos product line* [Web Page]. Retrieved April 30, 2000, from <http://www.wacom.com/productinfo/intuos.html>
- Wagner, A. (1990). Prototyping: A day in the life of an interface designer. In B. Laurel (Ed.), *The art of human-computer interface design* (pp. 79–84). Reading, MA: Addison-Wesley.
- Walker, M., Takayama, L., & Landay, J. (2002). High-fidelity or low-fidelity, paper or computer medium? *Proceedings of HFES 2002, The Human Factors and Ergonomics Society 46th Annual Meeting*, 661–665.
- Waterson, S., Hong, J. I., Sohn, T., Heer, J., Matthews, T., & Landay, J. A. (2002). What did they do? Understanding clickstreams with the WebQuilt Visualization System. *Proceedings of AVI 2002, The ACM International Working Conference on Advanced Visual Interfaces*. New York: ACM.
- Wong, Y. Y. (1992). Rough and ready prototypes: Lessons from graphic design. *Proceedings of CHI '92, The ACM Conference on Human Factors in Computing Systems (Short Talks)*, pp. 83–84). New York: ACM.
- Young Ideas. (1999). *Young ideas—methodology* [Web Page]. Young Ideas. Retrieved January 20, 1999, from <http://www.yideas.com/methodology.html>