Improving the Early Phases of Web Site Design via Informal Design Tools and Automated Usability Assessment

Marti Hearst
Information Management & Systems
102 South Hall
University of California, Berkeley
Berkeley, CA 94720-4600
hearst@sims.berkeley.edu
http://www.sims.berkeley.edu/~hearst

James Landay
Computer Science Department
683 Soda Hall
University of California, Berkeley
Berkeley, CA 94720-1776
landay@cs.berkeley.edu
http://www.cs.berkeley.edu/~landay

September 29, 1999

PROJECT SUMMARY:

Improving The Early Phases of Web Site Design via Informal Design Tools and Automated Usability Assessment

Despite the abundance of design recommendations and guidelines, web site usability, especially for information-centric sites, continues to be a pressing human-computer interaction issue. Given a projected growth of 196 million new sites within the next five years, an estimated 90% of web sites providing inadequate usability, and a severe shortage of user interface professionals to ensure usable sites, tools and methodologies are needed to accelerate and improve the web site design process. In this proposal, we describe tools and techniques intended to make web site design more natural, efficient and effective.

One way to improve web site design is to make the iterations in the initial design faster and easier to change. To this end, we propose to improve the early phases of web site design via two interlinked mechanisms: creating new software design tools to support the early informal parts of the design process, and creating new evaluation methods for helping designers make motivated choices between alternative designs before implementation.

Informed by our ethnographic study of web site design, we have designed DENIM, a pen-based web site design system that supports sketching input, allows design at different refinement levels, and unifies these levels through zooming. This builds upon our earlier experience with SILK, a tool for sketching graphical user interfaces. In DENIM we extend that work to support the specific needs of large information-centric web sites, including information design and navigation design.

To fully and effectively support the iterative design process, the initial implementation of DENIM needs to be augmented in two major ways. First, it must be extended support *redesign* of *existing* designs. Second, it must support facilities to perform *evaluation* within the early stages of design. Thus we will develop new algorithms, strategies, and tools to allow designers to evaluate the usability of their designs during early phases of design, allowing for comparison of different design options before undergoing the costly process of implementation.

One of the underexploited opportunities in usability assessment resides in the area of automation of evaluation. We have surveyed the state of the art in automation of user interface usability evaluation. This revealed significant gaps in the automation design space. Of the 53 usability methods surveyed, only 13% can be conducted without formal or informal user or evaluator testing. We propose to investigate two types of simulation that have not been applied to user interface evaluation: trace-driven discrete event simulation

and Monte Carlo simulation. We will integrate these facilities within DENIM so that it is easy to alternate between (re)design and evaluation of design. In summary, this project will address the following research questions:

- Can computerized informal design tools be developed that are nearly as natural to use as the informal paper tools currently used in information design and navigation design?
- Can the web site design process be improved by the application of automated evaluation techniques such as simulations?

The project will result in the following deliverables:

- Tools for informal design of web site information and navigation structure.
- Evaluation of these tools.
- Simulation algorithms and tools for aiding in the comparison of information-centric web site designs.
- Evaluation of these algorithms and tools.
- Tutorials on how to perform new evaluation methods.
- Tight integration of these two sets of tools.

The design tools will be made available to professional designers, and their feasibility will be assessed in relation to paper-based tools and formal computerized design tools. The output of the usability evaluation tools will be validated by comparing the simulator behavior to the results of user studies. We will use a large health care intranet as our evaluation testbed.

1 SIGNIFICANCE OF PROPOSED RESEARCH

Despite the abundance of design recommendations and recipes for building a usable web site [18, 19, 65, 67, 68, 76, 78, 82, 83, 84], web site usability, especially for information-centric sites, continues to be a pressing human-computer interaction issue. With an estimated 90% of web sites providing inadequate usability [75], a projected growth of 196 million new sites within the next 5 years [68] and a severe shortage of user interface professionals to ensure usable sites, tools and methodologies are needed to accelerate and improve the web site design process. Nielsen estimates [68] that there are only about 20,000 user interface professionals, and notes that if all sites were to be professionally designed, then every usability professional in the world would need to design one web site every working hour to meet demand.

Another recent report [26] describes the results of IBM's recent efforts to revise its external web site:

> On IBM's web site, the most popular feature was the search function, "because people couldn't figure out how to navigate the site," said Carol Moore, IBM's vice president for Internet operations. The second most popular feature was the 'help' button, because the search technology was so ineffective.
>
> IBM's solution was a 10-week effort to redesign the site, which involved more than 100 employees at a cost Ms. Moore estimated "in the millions." As the redesign neared completion in February, IBM consciously held off on promoting the web site, so it could gauge the effectiveness of the new design. The result: In the first week after the redesign, use of the 'help' button decreased 84 percent, while sales increased 400 percent.

(It is important to note that the improvement is attributed to the redesign of the site's information structure, as opposed to the pages devoted to the purchasing process.) This anecdote indicates that not only did the organization's profits increase, but the time and effort required by the customers was also probably reduced.

However, not every organization can afford to spend millions to redesign their sites. How can the design and evaluation of the usability of web sites be improved? A key idea is to make the iterations in the initial design faster and easier to change and evaluate. This would allow detailed what-if analysis to be conducted in the early stages of the design to avoid costly mistakes.

To this end, we propose tools and techniques to make web site design more natural, efficient and effective. Informed by an ethnographic study of web site design, we have created a preliminary implementation of a pen-based system, DENIM, that supports sketching input, allows design at different refinement levels, and unifies the levels through zooming. In a preliminary evaluation, web site designers gave high marks to DENIM's improvement in their ability to express their ideas, iterate quickly through versions of a design, and overall efficiency. This system, its preliminary evaluation, and the improvements that are required to make it a fully usable tool are described in Section 2.

Section 3 describes the second goal of this proposal: the creation of a methodology and tools for comparative evaluation of alternative web site designs. The results of our extensive survey of automated usability methods [39] revealed that a new approach to usability evaluation – simulation – should be quite effective for comparative analysis of alternative web site designs. Our proposed approach is a combination of a trace-driven discrete-event simulator and a Monte Carlo simulator.

One reason a designer should be able to use such a simulator to realize cost savings is that one of the more time-consuming parts of the web site design task is the graphic design. Our study, as well as other accounts [78] find the web site design process tends to be grouped into three areas: information design, navigation design, and graphic design. These areas overlap, as does the role of the people who carry them out (see Figure 1). The graphic design portion has been shown to be relatively unimportant for the findability on the site [65, 84] but is relatively more expensive to redesign and reimplement than the information structure or the navigation structure [19, 76, 78]. The graphical layout of the page is highly dependent on the navigation structure and information organization, but the converse does not hold.

Significant savings should be realized by simulating the relative effectiveness of the information structure and the navigation structure before the site is implemented. Simulation should help assess systems globally, across a wide range of tasks. They should be helpful in reducing the number of designs to prototype,

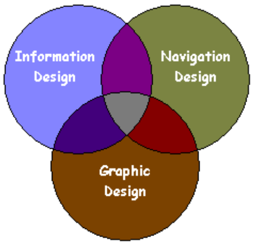


Figure 1: Three components of web site design.

improving initial prototypes, and helping designers choose among design alternatives before committing to expensive development costs.

This is not to preclude the use of low-fidelity prototypes and discount usability engineering; these should be used as well. Although automated UE has many benefits, it does not capture important qualitative and subjective information (such as user preferences and misconceptions) that can only be unveiled via user testing, heuristic evaluation, and other standard inquiry methods.

This project will address the following questions:

- Can computerized informal design tools be developed that are nearly as natural to use as the informal paper tools currently used in information design and navigation design?
- Can the web site design process be improved by the application of automated evaluation techniques such as simulations?

This investigation will result in a design tool that reflects the current practices of real web designers, allowing for tightly coupled iterative design and evaluation in the early stages, when iteration is most cost-effective [69].

2 AN INFORMAL WEB SITE DESIGN TOOL

We have conducted an ethnographic study of web site design practice [64] during which we observed that web site designers design sites at different levels of refinement – site map, storyboard, and individual page – and that designers sketch at all levels during the early stages of design. However, existing web design tools do not support these tasks very well. Informed by these observations, we have designed and made a preliminary implementation of DENIM, a system that helps web site designers in the early stages of design. The current version of the system supports sketching input, allows design at different refinement levels, and unifies the levels through zooming.

We have performed an informal evaluation of the system with seven professional designers and found that they reacted positively to the concept and were interested in using such a system in their work. 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. They also gave similarly high marks to DENIM's improvement in their ability to express their ideas, iterate quickly through versions of a design, and overall efficiency.

This study provides strong evidence that this informal design tool would be much preferred by web site designers over the currently available alternatives. However, much work remains to be done in order to make this a fully usable design tool. First, it must be extended support redesign of existing sites. This will require facilities to importing and conversion of the representation of a site into a useful internal format. Second, it should support evaluation of alternative designs. This is described in more detail in Section 3. Additionally, it should be extended in several ways to support the needs expressed by the participants of the informal study.

The next subsection places this work in context of related work. This is followed by a description of the current state of the system, a synopsis of our informal evaluation, and a description of the work we propose to do to complete the system.

2.1 RELATED WORK

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 [34]. 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 [1, 54, 14, 80, 86, 87], or by processing the ink internally while displaying the unprocessed ink [27, 53, 63, 79].

DENIM is most closely related to our earlier work on SILK [53], a sketch-based user interface prototyping tool. Using SILK, individual screens can be drawn, with certain sketches recognized as interface widgets. These screens can be linked to form storyboards [55], 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. DENIM does very little recognition, and what little it does is done in the background, without getting in the user's way. DENIM's use of storyboarding for behaviors is similar to SILK. Other systems influenced by SILK that use storyboarding include Anecdote [28] and PatchWork [86].

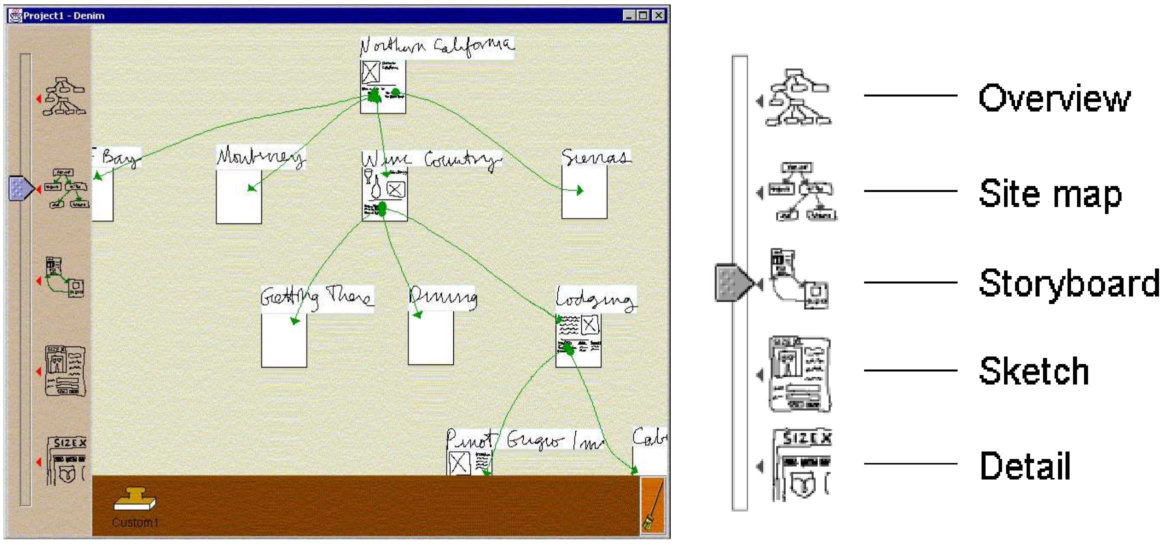
WebStyler [34] is another sketch-based tool for prototyping individual web pages. However, DENIM addresses more aspects of web site design, including designing the site structure and being able to interact with the sketches. There are many problems with using high-fidelity prototypes too early in the design process. By overspecifying the interface, designers are distracted from more fundamental issues such as deciding how the user interface should be structured [8].

The importance of thinking of web site design as a process of refinement and representing the design at multiple levels of detail is discussed in [76, 78]. These accounts of the web site design process are prescriptive in nature, whereas our study into web design was intended to be descriptive of current practices in the field. Our study was also focused on providing information directly related to building tools to support web site designers. There is a lack of early-stage prototyping tools for the web. Our ethnographic study showed that web designers use other tools to fill this gap. Macromedia Director is often used to assemble storyboards, while Visio is used for prototyping the high-level information architecture of a web site. However, Director is designed primarily as 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, since they are not designed for those tasks.

Currently, the most popular tools for creating web sites include Microsoft FrontPage, Adobe GoLive, Macromedia Dreamweaver, and NetObjects Fusion. However, these tools focus on designing page layout rather than the information architecture. Admittedly, each of them has a "navigation view" or a "site structure view" of a web site, which represents the site as a tree. 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 usually distinct and not unified. Lastly, 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.

2.2 THE DENIM SYSTEM

Informed by our ethnographic study [64], we designed and implemented a prototyping tool to assist designers in the early stages of web site design. Intended to be more informal than SILK, we named our system DENIM, which also conveniently stands for Design Environment for Navigation and Information Models. Since we wanted to make our system available to as many designers as possible, we built DENIM in Java 2 using JDK 1.2.2. The DENIM interface is shown in Figure 2a. The center area is a canvas where the user can create panels representing web pages, sketch the contents of those pages, and draw arrows between pages to represent their relationship to one another. The bottom area is a toolbox that will hold tools for inserting



(a) (b)

Figure 2: The DENIM interface showing the *site map* view (a), and detail of the zoombar (b).

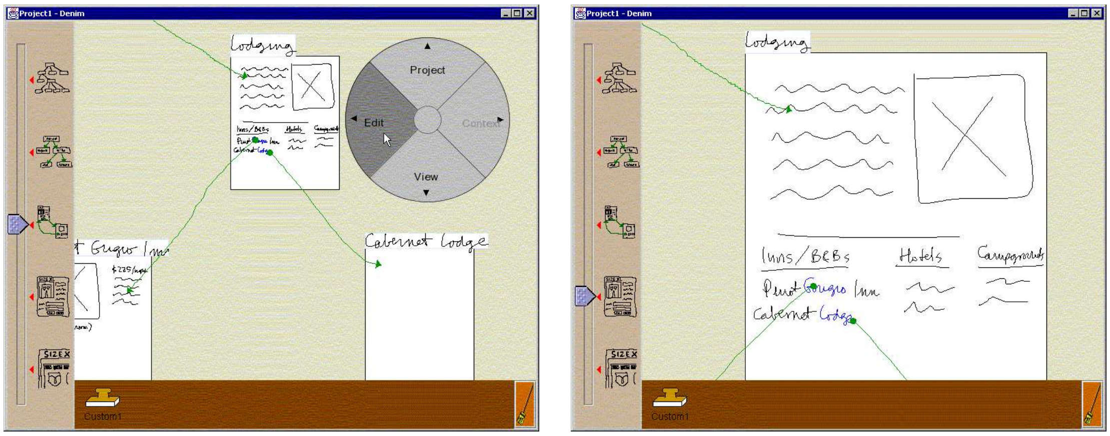
reusable components, such as templates. However, this part is not currently implemented; we plan to add this functionality as part of this proposal.

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 2b). There is also an intermediate zoom level in between each main level. Three zoom levels – the site map, storyboard, and sketch levels – are equivalent to the most common representations of web site designs that we observed during our ethnographic study. The site map level (Figure 2a) gives a view of the site as connected labels with attached thumbnails of individual pages. The storyboard level (Figure 3a) allows the user to view several pages simultaneously and more clearly see the navigational relationships between the pages. The sketch level (Figure 3b) 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, and the detail level providing a more fine-grained view of individual pages, for more precise sketching. To change the zoom level, the user either drags the slider's elevator or clicks directly on one of the icons.

Creating Pages In the current design of DENIM, web pages are represented by a label and a panel. The label represents the name or description of a page, while the panel represents the appearance of the page. The labels remain the same size throughout all the zoom levels, so that they can always be read. There are two ways to create a new web page in DENIM. The first way is to simply write some words directly on the canvas while in site map view. These words are automatically converted into the label of a page, and a blank panel is created. The other way is to draw a rectangle, which is converted to a page of approximately the same size as the rectangle. We plan to enhance the system to allow importing of information organizations that have been created using other tools, as well as of existing web sites.

Links Arrows are drawn between two pages to represent a link between those pages. We provide two kinds of links: navigational and organizational. Navigational links are links in the HTML sense: they represent the reference from an item on one page (e.g., a word or image) to another page. Organizational links are used to represent a conceptual link between two pages; that is, the designer eventually wants to make a navigational link from one page to another, but does not want to fill in the details at this time.

Run Mode After a number of pages have been sketched and navigational links drawn between them, it is possible to preview the interaction by entering Run mode. In Run mode, a separate "web browser" window comes up, and individual sketched web pages are displayed in it. If an element inside a page is the source of a navigational link, it is rendered in blue in the browser. Clicking on these elements causes the browser



(a) (b)

Figure 3: The DENIM interface *storyboard view* (a) and *sketch view* (b).

to display the target of the link, just as in a conventional browser. With Run mode, designers can test the interaction of sites that they are designing without having to create a full-fledged prototype. However, this mode should be augmented to capture usage trace information to better support evaluation.

Gestures and Pie Menus Most commands in DENIM can be activated either through gestures or through pie menus. The current implementation supports a relatively small set of gestures, as we are still experimenting with how to best map the functions of DENIM to a set of gestures. To activate a gesture, the user presses the button on the barrel of the pen and makes a stroke. Using a modified version of GDT [59] and Rubine's recognizer [77], we implemented gestures for panning, undo, redo, group select (select everything enclosed by a circular gesture), cut, copy, and paste.

We use a form of semantic zooming [6] in which the interaction with objects changes with zoom. Several of the editing gestures work differently depending on the current zoom level. In the two broadest views, the overview and site map views, gestures work shallowly. That is, you can only select, move, or edit web pages, but not anything inside of a web page. In the two narrowest views, the sketch and detail views, gestures work deeply. That is, you can only select, move, or edit individual ink objects inside a web page, but not web pages themselves. The middle zoom view, the storyboard view, (see Figure 3a) is a mixture of the two, depending on context. For example, the user taps the page's label or an empty area of the page to select the entire page but can tap any object inside a page to select that object. Pie menus [10] are used to provide access to functions not easily mapped to gestures, as well as providing redundant access to certain commands, such as cut, copy, and paste.

2.3 AN INFORMAL EVALUATION

Before completing a full-fledged version of this system, we wanted to be sure we were on the right track in creating a useful tool. To this end, we conducted an informal evaluation of DENIM in order to gain feedback about the usefulness of the basic 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.

2.3.1 Methodology

Seven professional designers participated in the study, five of whom said that web site design projects constituted at least half of their current workload. The remaining two participants were a user interface 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 560Z ThinkPad 300MHz Pentium II laptop running Windows NT 4.0, and an ITI VisionMaker Sketch 14 display tablet.

One evaluation session was conducted per participant, and each evaluation session consisted of three parts. First, 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 gave the user a pre-loaded web site design that we had previously created. and asked them to use DENIM 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 in which the designers were to create a web site to help renters find places to rent and to help landlords find tenants. The participant had 45 to 60 minutes 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 recorded what types of actions they did (e.g., panning, drawing, and creating new panels) and at what zoom levels they performed those actions. This was to give us a sense as to 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.

2.3.2 Results

Users made substantial use of different zoom levels, with usage concentrated primarily in the middle three levels (site map, storyboard, and sketch). 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.

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 current integration of these views through zooming sometimes proved to be problematic. Several of the users became frustrated navigating around their site designs and found that they often had to zoom out to a higher level in order to find their desired target and then zoom back in on that target.

Likewise, users had trouble creating navigational links 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 the target, yet have enough detail to be able to find the specific object on the source page that they wished to represent the link source. One possible way to ameliorate both of these problems would be to introduce a focus+context view [21] into DENIM. Being able to see 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, being able to compress 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. We plan to implement and test this feature as part of this proposal.

Users appreciated the informal mode of interaction provided by DENIM. 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. Two users experienced difficulty reading 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 part of this proposal, we plan to address these concerns by improving handwriting input, as well as supporting the progressive refinement of text objects by allowing their replacement with typed text.

The responses to the post-test questionnaire, though informal, were instructive in several ways. The five web designers had generally positive opinions of DENIM. First, while the web designers ranked the ease-of-use just above average (6.4 out of 10), they ranked the usefulness quite 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 (6.14 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.

2.4 NECESSARY ADDITIONS AND EXPANSIONS

To support redesign of existing sites we must augment DENIM to be able to map an existing site into a design. We need well-defined methods for representing the navigation structure and repeated navigation elements (i.e., navigation bars) in particular. The import engine should support importing both the structure and metadata associated with the site. DENIM should also be able to update the design to maintain consistency with the existing site.

We would like DENIM to work with existing web design tools to fit more naturally into the entire web design cycle. This includes generating HTML and other artifacts that can be imported by other tools, and providing mechanisms for automatically generating medium-fidelity prototypes for presentation purposes. DENIM should also be able to import files from other tools, so that designers can smoothly move back and forth in the design process. Another requested feature is to have a robust history and versioning mechanism, so that a designer can easily search through the evolution of a particular design and compare different designs within a project.

We have devised extensions to the storyboarding mechanism [58] to support the design of more sophisticated web interfaces as well as traditional GUIs, including methods to allow designers to specify their own reusable components. These components can be as simple as a new kind of widget or as complex as a template for a web page. We also need to integrate the designs produced by DENIM with underlying data models so the design can be linked up with databases of content.

We are looking into focus+context techniques to help solve several visualization problems that we described in the previous section. We are also looking into other techniques to help keep a sense of context while zoomed in on individual pages, including radar views, as done in [25]. 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.

Our ethnographic study showed that web site designers use large spaces such as desks and walls to design the information architecture of sites and explore variations of those architectures. One common way this is done is to use Post-It Notes to represent chunks of information and spatially arrange the notes into groups that will represent structural elements of the site [78, 7]. We plan to add support for using spatial arrangement of physical Post-It Notes as in input to DENIM. We will use computer-based object tracking to capture the arrangement of the Post-It Notes and translate those arrangements into DENIM representations of information organization. Having this part of the process captured will aid in the creation and storage of web site metadata, which will be useful for our proposed work in automated usability assessment. Additional changes will be needed to extend DENIM to support usability assessment; these are discussed in Section 3.

2.5 EVALUATION

We plan to make the design tools available in the near term, deploying them in the field, in order to start getting some real usage feedback from designers using them for real projects. We will also perform a longitudinal study in which we give the system to one design firm and our health care testbed (described in the next section), and track their usage closely.

When the system is completed we will evaluate it via formal user studies comparing it against the use of paper artifacts and standard formal design tools. We will use metrics similar to those described above for the informal study as well as more quantitative metrics. We will use designers both from web design firms and from our testbed site.

3 WEB SITE SIMULATION

Usability evaluation (UE) helps ensure that user interfaces enable users to achieve specific goals effectively, efficiently and satisfactorily. The rise of the web has alerted a large segment of society to the importance of good user interface design. Nonetheless, usability evaluation is still a major bottleneck in the iterative design process, and innovations to support this activity lag far behind support for designing and implementing traditional and web user interfaces [38, 69, 81].

How can usability evaluation be improved? One approach is to expand the use of *automated* usability evaluation (AUE) methods. Automated methods should be useful for aiding in comparisons between alternative designs, and for predicting time and error costs across an entire design. It is important to note that we consider automated techniques to be a useful *addition* to standard evaluation techniques such as heuristic evaluation and user testing – not a substitute – because these techniques uncover different kinds of problems [43, 61, 62]. A simulator can make estimates about time required and reachability between all pairs of nodes within a web site – a task too arduous to ask of real users. It can also simulate a wide range of user types, behaviors, and goals, and help the designer identify which parts of alternative designs are optimized for which user types and tasks. It should help the designer identify bottlenecks in the navigation structure, suggesting where additional links are needed, or where changes in the information organization would be beneficial. A simulation conducted during the design phase should also help reduce costs by reducing the likelihood of implementation of faulty designs. Thus our goal in this proposal is to create a simulation system that allows for comparative evaluation of alternative web site designs, especially within the early stages of design.

The next subsections discuss related work in automated usability assessment of web sites, and provide some background about the technique we will focus on, web site simulation. We then present a description of the requirements necessary to implement such a tool and a scenario illustrating how a designer might use simulations to compare and improve some existings designs. This is followed by a description of the changes that must be made to DENIM to support such an evaluation as part of the design process, as well as a discussion of how we intend to evaluate this methodology.

3.1 RELATED WORK

Despite the potential benefits of AUE, this field is greatly underexplored, as described in our extensive survey on this topic [39]. To gain further insight into new methods for automated usability assessment, we systematically compared methods for usability evaluation to those of performance evaluation (PE) [40]; PE is an established methodology for automatically generating quantitative performance data for computer systems. In [40] we describe three methods for automation of usability evaluation: measurement, analytical modeling, and simulation. We found that the major challenges for automating usability evaluation are: automatically generating high-level usage traces, mapping these high-level traces into user interface operations, and simulating user interface behavior across a range of tasks.

As mentioned above, in this research, we propose to focus on simulation, both trace-driven and Monte Carlo. Simulation complements traditional UE methods and inherently supports automatic analysis. Using models of the user and/or the user interface, these approaches simulate the user interaction and report the results of this interaction. Evaluators can run simulations with different parameters in order to study various design tradeoffs and thus make more informed decisions about implementation. Pew and Mavor [71] provide a detailed discussion of this type of modeling and an overview of many of these approaches, including five that we also discussed in our survey: ACT-R (Adaptive Control of Thought) [2], COGNET (COGnition as a NEtwork of Tasks) [88], EPIC (Executive-Process Interactive Control) [50], HOS (Human Operator Simulator) [23] and Soar [74]. Our survey also considered CCT (Cognitive Complexity Theory) [48], ICS (Interacting Cognitive Subsystems) [5] and GLEAN (GOMS Language Evaluation and ANalysis) [49].

Most user interface simulations rely on some variation of the Model Human Processor [11], making use of GOMS analysis [16, 24, 44, 45, 46]. This means that they attempt to model the steps involved in cognitive processing. To represent the underlying user, simulation methods use either task hierarchies (as in GOMS task structure: HOS, CCT), production rules (CCT, ACT-R, EPIC, Soar, ICS), or declarative/procedural programs (GLEAN, COGNET). CCT uses both a task hierarchy and production rules to represent the user and system models respectively. Some model perceptual and motor processing as well (EPIC, ICS, HOS, Soar, GLEAN, CCT). None of these simulation approaches have been adapted for the web.

Our survey revealed only one simulation tool for the web – WebCriteria's Site Profile tool [13]. Site Profile simulates a user's information-seeking behavior within a model of an implemented web site. Site Profile uses a web user model that follows an explicit navigation path through the site and computes an accessibility metric based on predictions of load time and other performance measures. This approach suffers from three major limitations: it uses a single user model, it requires an implemented system and specification of an explicit navigation path, and it does not model characteristics of the web page itself (meaning it only takes into account easily measureable properties such as number of links and predicted load time). As will be shown below, our proposed approach would not have these limitations.

Also relevant to web site evaluation is prior work in log file capture and analysis. This work is less automated than simulation but is an important component. Our survey revealed three general approaches for analyzing web log files: inferential, metric-based, and task-based.

Inferential analysis of web log files includes traffic-based analysis (e.g., pages-per-visitor or visitors-per-page) and time-based analysis (e.g., click paths and page-view durations) [15, 20, 85]. These techniques attempt to infer something about usability or the user's interest in page contents from log files. This analysis is largely inconclusive, since server logs provide only a partial trace of user behavior. Furthermore, server log files are missing valuable information about what tasks users want to accomplish [9].

Metric-based approaches generate quantitative performance measurements. One tool developed by Service Metrics [60] and others [3] allow evaluators to pinpoint performance bottlenecks, such as slow server response time, that may negatively impact the usability of a web site. This is important information to take into account, but does not make use of user models or task models. Furthermore, as in Site Profile, these metrics have focused on easily measurable attributes but not information content and navigation structure of web sites.

Task-based approaches, such as ÉMA [4] and USINE [57], analyze discrepancies between the designer's task model and actual use. ÉMA uses a data-flow task model along with behavior heuristics to detect specific user behaviors that may indicate usability problems. Immediate task cancellation or a shift in direction during task completion are two behaviors detected by ÉMA. USINE compares a task model expressed in terms of temporal relationships to usage traces and reports substantial analysis data. This is a promising approach, but has only been applied to "functional" user interfaces, and needs to be expanded to work on information-centric web interfaces.

One of our earlier projects was the Quantitative User Interface Profiling (QUIP) [36] tool which supports task-based, log file analysis and visualization of function-oriented web UIs. Unlike other approaches, QUIP aggregates traces of multiple user interactions and compares the task flows of these users to the designer's task flow. QUIP encodes quantitative time- and trace-based information, such as the average time between actions and the fraction of users who performed a particular sequence of actions into directed graphs. The research proposed here differs from the work on QUIP in two major ways. First, QUIP was applied to a function-oriented interface, as opposed to an information-centric web site. Second, QUIP can only be applied to already-implemented interfaces, whereas here we intend to work on evaluating alternative designs before they are implemented. This distinction is critical to the design of an evaluation technique.

3.2 THE SIMULATORS

3.2.1 Background: New Types of Simulations

In simulation, the evaluator constructs a detailed model of a system in order to reproduce its behavior [42]. Typically, a computer program (known as a simulator) exercises this underlying model with various input parameters and reports resulting system performance. The underlying simulation model is one of the major differences among the various simulation approaches. The defining characteristics of these models are: the way the system evolves (time dependent or time independent), how its parameters are generated, and the number of states it can evolve into. Another distinguishing feature of a simulator is its workload format. The workload may be in the form of fixed or probabilistic parameters that dictate the occurrence of various system events or an execution trace captured on a real system. Performance analysts consider trace-driven simulations to be the most credible and accurate [42].

Two frequently-used simulation types in the PE domain are discrete-event and Monte Carlo simulation. Discrete-event simulations model a system as it evolves over time by changing system state variables instantaneously in response to events. Analysts use this approach to simulate many aspects of computer systems, such as the processing subsystem, operating system and various resource scheduling algorithms. Execution traces are often employed in discrete-event simulations, since it is relatively easy to log system events. However, application of execution traces to discrete event simulation of UIs is an area that has not been explored in usability evaluation.

Monte Carlo simulations model probabilistic phenomena that do not change characteristics with time. Analysts use this approach to conduct what-if analysis (i.e., predict resulting performance due to system resource changes) and to tune system parameters. Due to the random nature of Monte Carlo simulations, execution traces are not usually employed with this approach.

Monte Carlo simulation is also untested in UE, and could contribute substantially. Most simulations in this domain rely on a single user model, typically an expert user. One alternative is to use a technique for automatically generating plausible usage traces. (Kasik and George [47] describe a method for generating usage traces for interface design whose characteristics differ depending on the underlying user type.) Such a simulator could simulate uncertain behavior characteristic of novice users. This would enable designers to perform what-if analysis and study design alternatives with different user models.

The major difference between Monte Carlo and discrete-event simulation is that the task sequence is not pre-determined in a Monte Carlo simulation. This type of simulation is appropriate for information-centric web sites. Instead of usage traces, the Monte Carlo simulator would pick a random starting point and have a fixed ending point during each simulation run. During the run, the simulator would choose links to follow based on a probability function that takes into consideration the content of links, user goals and other pertinent information. The simulator would report simulated navigation time along with navigation traces to facilitate analysis and to possibly use with a discrete-event simulator. The designer could also use simulation results for multiple site designs to determine the best information architecture for a site.

3.2.2 Requirements

Several mechanisms must be specified in order to build such a simulator. One is a mechanism for path generation for use with the Monte Carlo simulator. Another is a mechanism for predicting and measuring various metrics (such as time, complexity, reachability), and finally an interface is needed that allows application of these mechanisms in a flexible manner. These require a format for representing user tasks and goals. They also require a method of modeling user actions. Currently we plan to create four user model types: expert, novice, basic, and random. We will follow and expand on the ideas of Kasik and George [47] on how to generate usage traces for interface design whose characteristics differ depending on the underlying user type.

Computing the quantitative metrics is the most challenging part of this project. Our objective is to be able to predict key factors that impact usability, such as page complexity, reading time, and “scent” between links. We will experiment with different ways of measuring readability, cohesion of information structure, and complexity of information. We will start with notions of scent in information navigation [22, 72, 73] and will also use empirically validated guidelines [65, 66, 84], and established reading complexity scores [12]. We will also take into account research on depth vs. breadth tradeoffs in web site design [56] as well as menu design [41, 52, 51]. We have extensive experience in information retrieval and computational linguistics [29, 30, 31, 32, 35, 33, 70], and will make use of this background in developing these metrics.

The simulator will combine these metrics to compute a number of quantitative measures – such as navigation time and reachability – and output navigation traces. This simulation algorithm can be used in several ways, including:

Check all navigation paths. After running the simulator a number of times (for one or more tasks) and displaying the traversed paths, the system will highlight all of the paths that were generated, along with their respective timing information, to indicate simulator coverage. An analysis of this coverage may identify areas where links need to be added to the site to increase reachability or links should be removed because they add unnecessary burden to choice making.

Check all pairs of navigation paths. The simulator can be configured to start from every page within the site and traverse to information within all other pages of the site. This may be useful for determining how easy or difficult it is to reach each item of information from every location in the site.

Such an analysis should indicate the relative effort necessary in moving between pairs of information that have some logical relationship to one another. It should also help determine how difficult it is for users to move out of a deep location within one subject area and into another subject area.

3.2.3 Simulation Runs

Each run of the simulator requires the following steps. First, pick a starting page. There are three different models for how to do this. (a) User specified. (b) Randomly chosen independent of task (this may be used for the all pairs condition) (c) Chosen based on the task (this is equivalent to following a link returned by a search engine or a link from an external page). The last choice would require some metadata representation.

Next, repeat the steps below until either (a) the target page is reached; or (b) a stopping criteria is reached (i.e., maximum navigation time, all paths from starting point exhausted, or maximum number of links traversed).

The steps for simulation are:

1. Read the page
   - This entails computing a read time based on the following:

     a. page complexity (e.g., content volume and difficulty)
     b. page visitation (if previously viewed, then less read time)
   - Update the system clock after computing the read time.
2. Make a decision
   - The decision time incorporates time for

     a. deciding if target is reached (if so, end simulation run)
     b. if not, deciding what to do next (this should be proportional to the number of options). This entails deciding on a navigation option (e.g., following a link or using the back button) as follows:
       i. survey options (create a list of choices)
       ii. compute a probability for selecting each option based on criteria described below.
       iii. prune options: if the probability for an option falls below a certain threshhold, then eliminate the option.
       iv. choose option: if there is more than one option remaining, then the Monte Carlo algorithm determines the choice; this choice could be based on history (i.e., a user model or record of previous choices) or random.
   - Update the system clock after computing the think time.
3. Navigate
   - This entails making the selected page the current page and marking it as visited. It also entails estimating a link traversal time and updating the system clock. This could be a fixed page download time or a variable that takes into account network variations, caching, and characteristics of the page itself.

Step 2b above requires computation of a probability for selecting a next move based on certain criteria. These include:

(1) Metadata match: do a pairwise comparison of metadata between the current and potential page to compute a score indicating relatedness of content (it should be possible to pre-compute these for all links and store them in a structure); do a pairwise comparison of metadata between the target information and next page to compute another match probability; multiply the probabilities to compute a final match probability.

Another possibility is to maintain a composite metadata analysis that's updated at each link. This composite analysis would take into account metadata on the previous pages traversed and the target information. This composite metadata could then be used in the pairwise comparison to compute a final match probability.

(2) Page visitation: if a page has been visited before, adjust selection probability according to the user model under use.

(3) User model: adjust the match probability based on the simulated user. For example, if we are simulating a user learning by exploration, then the system state would reflect the user's current learning. If a page under consideration is consistent with a user's learning, then we would increase the match probability. Note that this criteria is different from (2), since it affects pages that have not been visited before.

3.2.4 A Scenario

Consider the following example of using simulations to evaluate alternative site designs (see Figure 4). This example is necessarily very simple for expository purposes.

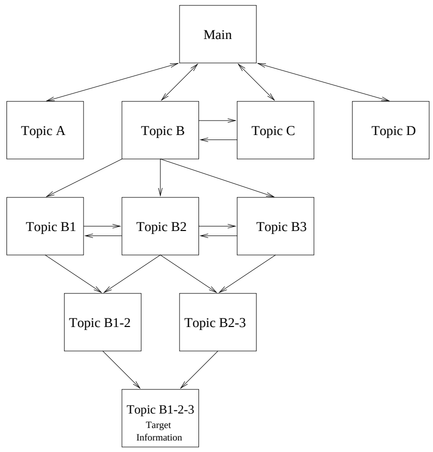


Figure 4: Web site example for evaluation scenario.

Say you are the designer of the XYZ Agency site. You have received feedback that many users find it difficult to locate a particular type of target information. You need to identify the source of difficulty and to determine the best way to mitigate the problem. There are some constraints on how you can address the problem. In particular, you want to change the content and navigation bars as little as possible.

Since this is the first time that you use DENIM with this site, you need to import information from the existing site. You specify the site's URL and options to perform a full import. The tool creates a representation of the site that includes the following for each page: a panel, a label (page title), and metadata that describes the page contents and topic. The import engine also computes a complexity score for each page based on the amount of text, images and links. Finally, each page is marked as an existing page versus a design page and automatically assigned a distinguishing color based on its metadata and annotated with its complexity.

Your next task is to identify the source of difficulty in locating the target information. You mark the page containing the target information as a destination page. DENIM automatically builds a structure to record the key information about this destination page. You then import server log data that depicts recorded clickpaths to this destination page. The clickpaths give you a *rough* picture of how users (i.e., host IP addresses) are navigating the site. DENIM highlights the imported clickpaths along and adjusts the thickness of each link to reflect the percentage of users that followed it. You then highlight an optimal path to the target information based on observed navigation. Your observation of user navigation paths show that the majority of people start from pages A or D, go to the main page and then to page B. From there, the majority of users go to page B3 only to cross link to B2. Most people follow the two additional links from B2 to the destination page. Based on these observations, you annotate pages A, D, and B as start pages for further study.

You then proceed to conduct a discrete event simulation of the predominate navigation paths. The discrete event mode of simulation allows you to explicitly state the navigation path to follow. (This is in contrast with the Monte Carlo simulation mode, in which only starting and ending nodes are indicated, and the path itself is generated by the simulator.) You specify that the discrete event simulator follows the highlighted paths from the three starting pages to the destination page and uses a basic user model. The simulator runs and predicts a match probability and navigation time for each page traversed.

The simulation results show that pages A and D have a high match probability for the target information, yet there is no direct link to the destination page. Both of these pages have a low complexity score (computed during the import operation), yet the navigation time (including reading, thinking and link traversal) is longer than expected. An inspection of the breakdown of navigation time shows that the majority of it was spent reading and thinking. This implies that the user had a difficult time deciding if the target information was on the page or reachable from the page, possibly due to how they entered the site. You inspect the server log data for these clickpaths and see that most users entered these pages from a search engine or external link. Hence, they may have thought that these pages contained the information they were seeking. Simulation results also show an unusually long navigation time for page B and a higher match probability between pages B and B3 than pages B and B2. You attribute the long navigation time to the page complexity.

Based on the simulation results, you create two designs within DENIM to address these problems. The first design promotes page B2 and its linked pages to the same level as page B and adds links from pages A and D to it. Pages B1 and B3 remain under page B and are linked to pages B1-2 and B2-3 respectively. There are no changes to content, metadata or link annotations in this design. The other design entails similar information restructing as well as changes to link annotations and metadata for pages A and D. These changes are to further distinguish pages A and D from page B2. DENIM automatically changes all modified pages in both designs to design pages. For each design, you highlight an optimal path.

You then conduct a Monte Carlo simulation of navigation from the start pages (A, B and D) to the destination page for each design. You use the same basic user model for these simulations. The simulation results show that the first design decreases the navigation time by 20% and the predominant navigation trace closely follows the optimal path. The results show a 45% decrease in navigation time for the second design.

As a consequence, you decide to conduct further simulation studies with the second design. In particular, you want to determine how accessible the target information is from all other pages in the site, especially after the site restructuring. (The expectation is not that accessibility to a given piece of information is uniform across the site; rather this computation is done to assess if a change to one part has a negative impact elsewhere.) You conduct a Monte Carlo simulation of navigation from all pages to the destination page. You specify the random user model for this study. The simulation results show that it is still difficult to start from Page B and get to the target information. You store the navigation path from this page for future use.

You attribute the navigation difficulty from Page B to its page complexity, which was not addressed in your redesign. You then vary the page complexity score and rerun the simulator with the stored navigation path. You continue this what-if analysis until the simulated navigation proceeds normally and the predicted navigation time is acceptable. You use this estimated page complexity to help you redesign the page. You validate your page redesign with further simulation studies.

Before implementing the site, you want to assess the impact of your redesign on accessing information globally throughout the site. DENIM helps you identify 4 popular destination pages from the server log data to use in addition to the current destination page. DENIM automatically marks these pages as destination pages. You conduct a Monte Carlo simulation study using these 5 pages as destination pages. You specify that the simulator use the random user model and complete at least 15 runs (i.e., navigation paths). You conduct the same simulation study on the original design. A comparison of the simulation results show that your redesign does not negatively impact navigation time for any of the destination pages.

After validating your design, you export the design pages in a basic HTML (or database) format for implementation.

3.2.5 Additional Considerations

Our longer-term goal is to create a system that not only compares design variations, but also suggests navigation structure given an information structure. Such a system, for example, would suggest which items to place on a navigation bar, depending on the distribution of information within the site. This is a more advanced capability; comparing variations of existing designs is an easier starting point. Because the evaluation system will be integrated within a design system, the necessary structural information and metadata will be available both for running simulations and for making suggestions about the site design.

A disadvantage to starting with an existing design is that it may not have appropriate metadata or descriptors of the contents of the web site. In this case, the markup needed by the simulator will require some hand-annotation in addition to automated methods for extracting such information. Fortunately, our testbed (see below) has this kind of information built into the web pages. To allow for flexible generation of designs, the system will eventually support declarative representations of content and structure, similar to that found in web database creation systems such as Strudel [17].

It would be more difficult to apply these ideas to compare two existing web sites that were designed using different underlying content. For example, it would be difficult to apply the simulator to compare two competing web sites. This would require additional work to map the existing web sites into an appropriate uniform representation. Therefore, we do not plan to support this facility in the near future.

3.3 EVALUATING THE SIMULATOR

We have a relationship with the Information Technology/Web Services division of Kaiser Permanente, a large health care institution, and will use a portion of their clinical intranet as the testbed for the evaluation research. This site already has a design, but we will use this to construct alternatives to its existing design. IT managers and usability specialists at Kaiser have already performed needs assessments and tasks analyses for this site; there is also a large and active user base who will act as participants in our user studies.

Before constructing the simulator, we will lay the groundwork for our evaluation within this site and user base. We will analyze a set of information-centric web sites and server logs to identify a useful section for redesign. We will conduct a task analysis to identify at least five typical tasks, and identify 3-5 design alternatives.

After the simulator has been developed, we will evaluate it within this setting. This evaluation will have two stages. We will first conduct a user study to validate predicted versus actual navigation times. We will first measure navigation time while users complete two of the five representative tasks for each of the alternative designs. If these do not match the predicted relative times we will modify the simulator to be more in line with actual usage.

Once the simulator is validated, we will run it on the alternative designs to generate performance predictions. We will choose two sites with nearly comparable performance (approximately 20%-30% performance difference) for a within-groups user study. Participants will complete the previously identified tasks using both interfaces in a random testing order. We will collect quantitative and subjective measures during testing. If the simulation is making incorrect predictions we will adjust the models it uses accordingly.

3.4 AUGMENTING DENIM TO SUPPORT EVALUATION

In the previous section, we described the additions to DENIM that are required to make it a more comprehensive web design tool. In order to support automated evaluation, we must augment it with additional capabilities described here.

We will augment DENIM to support highlighting of navigation paths. The designer should be able to click on paths to highlight them as well as input paths from a file. It may be desirable to also animate a navigation path; this would be particularly useful for simulation studies.

To facilitate analysis, DENIM should allow the designer to view multiple designs at some level of detail. This would provide limited support for design exploration in conjunction with the previously mentioned highlight option.

DENIM should interface with the path generation engine. The interface should enable the designer to specify a number of features for this engine, such as a task template, target information, a starting point, the number of paths to generate and a generation mechanism. The designer should be able to click on nodes to specify target information. Specifying a fixed starting point and target information would enable a designer to contrast generated paths for alternative designs. The engine will use the DENIM design to generate navigation paths and store them in an output file.

DENIM should also interface with a trace processing engine that parses server log files to identify navigation paths. This engine will use the DENIM design to identify valid paths and store them in an output file. Finally, DENIM should be able to process output files from both the path generation and trace processing engines to highlight the paths. Highlighting all paths in a file could help the designer assess site coverage.

4 SCHEDULE AND DELIVERABLES

The proposed research for improving the early stages of web site design is to complete a tool that provides an informal means of sketching interface designs and transforming them into computerized form. This will require the following steps:

- (Year 1) Implement mechanism to import existing sites for redesign.

- (Year 1) Implement mechanism for converting ink to formal representation.
- (Year 1) Experiment with focus+context mechanisms.
- (Year 1) Implement tools for capture of information/metadata design.
- (Year 1) Integrate handwriting recognition mechanisms.
- (Year 1) Begin longitudinal study.
- (Year 2) Implement history and versioning mechanisms.
- (Year 2) Implement mechanisms to support reusable components.
- (Year 2) Evaluate tools for capture of information/metadata design.
- (Year 2) Continue and then conclude longitudinal study.
- (Year 3) Interface to standard web design tools.
- (Year 3) Interface to standard database tools.
- (Year 3) Controlled evaluation of system in comparison with formal techniques.

One of our graduate students already conducted the ethnographic study design described above, and he and a second student implemented the initial version of DENIM over a six month time period. This student is highly qualified to achieve these goals in the indicated time period.

The proposed research for comparative web site evaluation will require the following steps:

- (Year 1) Identify a Web redesign application.
- (Year 1) Augment DENIM to support evaluation.
- (Year 1) Develop a navigation path generator.
- (Year 2) Validate the navigation path generator.
- (Year 2) Develop the simulator.
- (Year 2) Validate the simulator.
- (Year 2) Apply the simulator to web site design comparisons.
- (Year 3) Modify the simulator to generate design suggestions.
- (Year 3) Evaluate these modifications.

One of our graduate students has performed several simulations and created analytical models of large computer systems. She has also conducted the automated usability survey described above and will be the primary implementor of the steps for year 1 and year 2. A second graduate student will be recruited to work on the modifications to the simulator to generate design suggestions, which should be completed in year 3. If time permits, after the initial version of the simulator is complete, we will focus on modifying the simulator to automatically generate suggestions for redesigns.

Besides publishing research articles describing this work and presenting the results of this work at international research conferences, we intend to make the finished tools for design and evaluation available for general use. We will also make documentation available describing how to use the systems, and intend to produce tutorial material to teach people how to conduct automated usability assessments.

5 RELATION TO OTHER NSF SUPPORT

Profs. Hearst and Landay are among 13 faculty investigators on the UC Berkeley NSF sponsored DLI2 grant, "Reinventing Scholarly Information Dissemination and Use." This grant was only recently awarded. Prof. Hearst's contribution to that project will expand on her prior work in computational linguistics and user interfaces for information retrieval. Her effort for that work will be to develop tools to help improve and semi-automate the process of extracting useful information from large collections of scholarly textual information, and so is not directly connected to this proposal. Prof. Landay's contribution will be the investigation of the use of a research notebook metaphor for accessing and using digital library information. This work will make use of the software architecture underlying DENIM [37], and so is partially connected to this proposal.

References

[1] Gregory D. Abowd, Christopher G. Atkeson, Jason Brotherton, Tommy Enqvist, Paul Gulley, and Johan LeMon. Investigating the capture, integration and access problem of ubiquitous computing in an educational setting. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, volume 1 of *Young Adult Learners*, pages 440–447, 1998.

[2] John R. Anderson. *The Adaptive Character of Thought*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1990.

[3] Beth Bacheldor. Push for performance. *Information Week*, September 20:18–20, 1999.

[4] Sandrine Balbo. ÉMA: Automatic analysis mechanism for the ergonomic evaluation of user interfaces. Technical Report 96/44, DSIRO Division of Informaiton Technology, 1996.

[5] Philip J. Barnard. Cognitive resources and the learning of human-computer dialogs. In John M. Carroll, editor, *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, pages 112–158. The MIT Press, 1987.

[6] Benjamin B. Bederson and James D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *Proceedings UIST'94*, pages 17–26. ACM Press, 1994.

[7] Hugh Beyer and Karen Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, San Francisco, 1998.

[8] Alison Black. Visible planning on paper and on screen: The impact of working medium on decision-making by novice graphic designers. *Behaviour and Information Technology*, 9(4):283–296, 1990.

[9] Michael D. Byrne, Bonnie E. John, Neil S. Wehrle, and David C. Crow. The tangled web we wove: A taxonomy of WWW use. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, volume 1 of *Organizing Information on the Web*, pages 544–551, 1999.

[10] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An empirical comparison of pie vs. linear menus. In *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, Menus, pages 95–100, 1988.

[11] Stuart K. Card, Thomas P. Moran, and Allen Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1983.

[12] L.L. Cherry and W. Vesterman. Writing tools – the STYLE and DICTION programs. In *4.3BSD UNIX System Documentation*. University of California, Berkeley 1991.

[13] Web Criteria. Max, and the objective measurement of web sites. http://www.webcriteria.com, 1999.

[14] Richard C. Davis, James A. Landay, Victor Chen, Jonathan Huang, Rebecca B. Lee, Francis C. Li, James Lin, Charles B. Morrey, III, Ben Schleimer, Morgan N. Price, and Bill N. Schilit. Notepals: Lightweight note sharing by the group, for the group. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, volume 1 of *Profiles, Notes, and Surfaces*, pages 338–345, 1999.

[15] M. Carl Drott. Using web server logs to improve site design. In *ACM 16th International Conference on Systems Documentation*, Getting Feedback on your Web Site, pages 43–50, 1998.

[16] Jay Elkerton, Steven J. Goldstein, and Susan L. Palmiter. Designing a help system using a GOMS model: A preliminary method execution analysis. In *Proceedings of the Human Factors Society 34th Annual Meeting*, volume 1 of *Computer Systems: Modeling*, pages 259–263, 1990.

[17] Mary Fernandez, Daniela Florescu, Jaewoo Kang, and Alon Levy. Catching the boat with strudel: Experiences with a web-site management system. In *Proceedings of ACM SIGMOD*, Seattle, WA, 1998.

[18] Vincent Flanders and Michael Willis. *Web Pages That Suck: Learn Good Design by Looking at Bad Design*. SYBEX, San Francisco, 1998.

[19] Jennifer Fleming. *Web Navigation: Designing the User Experience*. O'Reilly & Associates, Sebastopol, CA, 1998.

[20] Rodney Fuller and Johannes J. de Graaff. Measuring user motivation from server log files. In *Proceedings of the Human Factors and the Web 2 Conference*, Designing for the Web, October 1996. Available from http://www.microsoft.com/usability/webconf.htm.

[21] George W. Furnas. Generalized fisheye views. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 16–23, April 1986.

[22] George W. Furnas. Effective view navigation. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Information Structures*, pages 367–374, 1997.

[23] F. A. Glenn, S. M. Schwartz, and L. V. Ross. Development of a human operator simulator version v (hos-v): Design and implementation. U.S. Army Research Institute for the Behavioral and Social Sciences, PERI-POX, Alexandria, VA, 1992.

[24] Richard Gong and Jay Elkerton. Designing minimal documentation using a GOMS model: A usability evaluation of an engineering approach. In *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, UI Models: Extensions & Applications of GOMS, pages 99–106, 1990.

[25] Saul Greenberg and Mark Roseman. Groupkit – A groupware toolkit. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, Formal Video Program: Prototypes and Enabling Technologies, page 9, 1994.

[26] Susan Greenwood. E-commerce report: Good web site design can lead to healthy sales. In *The New York Times*, August 30 1999.

[27] Mark D. Gross and Ellen Yi-Luen Do. Ambiguous intentions: A paper-like interface for creative design. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Papers: Programming by Demonstration, pages 183–192, 1996.

[28] K. Harada, E. Tanaka, R. Ogawa, and Y. Hara. Anecdote: A multimedia storyboarding system with seamless authoring support. In *Proceedings of the ACM International Conference on Multimedia*, pages 341–351, November 1996.

[29] Marti A. Hearst. Noun homograph disambiguation using local context in large text corpora. In *The Proceedings of the 7th Annual Conference of the UW Centre for the New OED and Text Research: Using Corpora*, Oxford, 1991.

[30] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING)*, pages 539–545, Nantes, France, July 1992.

[31] Marti A. Hearst. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd Meeting of the Association for Computational Linguistics*, pages 9–16, Las Cruces, NM, June 1994.

[32] Marti A. Hearst. Improving full-text precision using simple query constraints. In *Proceedings of the Fifth Annual Symposium on Document Analysis and Information Retrieval (SDAIR)*, Las Vegas, NV, 1996.

[33] Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, March 1997.

[34] Marti A. Hearst, Mark D. Gross, James A. Landay, and Thomas E. Stahovich. Sketching intelligent systems. *IEEE Intelligent Systems*, 13(3):10–19, 1998.

[35] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of the 19th Annual International ACM/SIGIR Conference*, pages 76–84, Zurich, Switzerland, 1996.

[36] Brian Helfrich and James A. Landay. QUIP: quantitative user interface profiling. Unpublished manuscript, 1999.

[37] Jason I. Hong and James A. Landay. A toolkit supporting pen-based interfaces. Technical Report UCB//CSD-99-1062, University of California, Berkeley, 1999.

[38] Scott E. Hudson, Bonnie E. John, Keith Knudsen, and Mickael D. Byrne. A tool for creating predictive performance models from user interface demonstrations. In *Proceedings of the ACM Symposium on User Interface Software and Technology, To appear*, 1999.

[39] Melody Y. Ivory and Marti A. Hearst. Automated usability evaluation of WIMP and Web interfaces. Submitted for publication, 1999. Available at http://www.cs.berkeley.edu/∼ivory/web/research/aue.ps.

[40] Melody Y. Ivory and Marti A. Hearst. Comparing performance and usability evaluation: New methods for automated usability assessment. Submitted for publication, 1999.

[41] Julie A. Jacko and Gavriel Salvendy. Incorporating task complexity into hierarchical menu design. In *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting*, volume 2 of *SPECIAL SESSIONS: Posters*, page 967, 1994.

[42] R. Jain. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling,*. Wiley-Interscience, New York, NY, USA, May 1991.

[43] Robin Jeffries, James R. Miller, Cathleen Wharton, and Kathy M. Uyeda. User interface evaluation in the real world: A comparison of four techniques. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, Practical Design Methods, pages 119–124, 1991.

[44] Bonnie E. John. Extensions of GOMS analyses to expert performance requiring perception of dynamic visual and auditory information. In *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, UI Models: Extensions & Applications of GOMS, pages 107–115, 1990.

[45] Bonnie E. John and David E. Kieras. The GOMS family of analysis techniques: Tools for design and evaluation. Technical Report CMU-CS-94-181, Carnegie-Mellon University, August 1994.

[46] Bonnie E. John and David E. Kieras. The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3(4):320–351, 1996.

[47] David J. Kasik and Harry G. George. Toward automatic generation of novice user test scripts. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Evaluation*, pages 244–251, 1996.

[48] David Kieras and Peter G. Polson. An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, 22(4):365–394, 1985.

[49] David E. Kieras, Scott D. Wood, Kasem Abotel, and Anthony Hornof. GLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface designs. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Evaluation, pages 91–100, 1995.

[50] David E. Kieras, Scott D. Wood, and David E. Meyer. Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. *ACM Transactions on Computer-Human Interaction*, 4(3):230–275, September 1997.

[51] John I. Kiger. The depth/breadth trade-off in the design of menu-driven user interfaces. *International Journal of Man-Machine Studies*, 20(2):201–213, 1984.

[52] T. K. Landauer and D. W. Nachbar. Selection from alphabetic and numeric menu trees using a touch screen: Breadth, depth, and width. In *Proceedings of ACM CHI'85 Conference on Human Factors in Computing Systems*, Menu Systems, pages 73–78, 1985.

[53] James A. Landay. Interactive sketching for the early stages of user interface design. Technical Report CMU-CS-96-201, Carnegie Mellon University, 1996.

[54] James A. Landay and Brad A. Myers. Interactive sketching for the early stages of user interface design. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1 of *Papers: Programming by Example*, pages 43–50, 1995.

[55] James A. Landay and Brad A. Myers. Sketching storyboards to illustrate interface behaviors. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, volume 2 of *SHORT PAPERS: Scribbling, Sketching, Drawing, and Writing*, pages 193–194, 1996.

[56] Kevin Larson and Mary Czerwinski. Web page design: Implications of memory, structure and scent for information retrieval. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, volume 1 of *Web Page Design*, pages 25–32, 1998.

[57] Andreas Lecerof and Fabio Paternó. Automatic support for usability evaluation. *IEEE Transactions on Software Engineering*, 24(10):863–888, October 1998.

[58] James Lin. A visual language for a sketch-based ui prototyping tool. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, volume 2, pages 298–299, 1999.

[59] Allan Christian Long, Jr., James A. Landay, and Lawrence A. Rowe. Implications for a gesture design tool. In *Proceedings of ACM CHI 99 Conference on Human Factors in Computing Systems*, volume 1 of *Alternatives to QWERTY*, pages 40–47, 1999.

[60] Service Metrics. Service metrics solutions. http://www.servicemetrics.com/solutions/solutionsmain.asp.

[61] R. Molich, N. Bevan, S. Butler, I. Curson, E. Kindlund, J. Kirakowski, and D. Miller. Comparative evaluation of usability tests. In *Proceedings of UPA98*, pages 189–200, June 1998.

[62] R. Molich, A. D. Thomsen, B. Karyukina, L. Schmidt, M. Ede, W. van Oel, and M. Arcuri. Comparative evaluation of usability tests. In *Proceedings of ACM99*, Panels, pages 83–86, May 1999.

[63] Thomas P. Moran, Patrick Chiu, and William van Melle. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Picking and Pointing, pages 45–54, 1997.

[64] Mark Newman and James Landay. An investigation into web design practice. In preparation, 1999.

[65] Jakob Nielsen. The alertbox: Current issues in web usability. http://www.useit.com/alertbox.

[66] Jakob Nielsen. Discount usability for the web. http://www.useit.com/alertbox.

[67] Jakob Nielsen. Web usability: Why and how. *Users First!*, September 14, 1998.

[68] Jakob Nielsen. User interface directions for the Web. *Communications of the ACM*, 42(1):65–72, January 1999.

[69] Jakob Nielsen and Robert L. Mack, editors. *Usability Inspection Methods*. Wiley, New York, 1994.

[70] David Palmer and Marti A. Hearst. Adaptive multilingual sentence boundary disambiguation. *Computational Lingiustics*, 23(2), 1997.

[71] R. W. Pew and A. S. Mavor, editors. *Modeling Human and Organizational Behavior: Application to Military Simulations*. National Academy Press, Washington, 1998. Available at http://books.nap.edu/html/model.

[72] Peter Pirolli. Computational models of information scent-following in a very large browsable text collection. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Finding What You Want I*, pages 3–10, 1997.

[73] Peter Pirolli and Stuart Card. Information foraging in information access environments. In Irvin R. Katz, Robert Mack, Linn Marks, Mary Beth Rosson, and Jakob Nielsen, editors, *Proceedings of the Conference on Human Factors in Computing Systems (CHI'95)*, pages 51–58, New York, NY, USA, May 1995. ACM Press.

[74] T. A. Polk and P. S. Rosenbloom. Task-independent constraints on a unified theory of cognition. In F. Boller and J. Grafman, editors, *Handbook of Neuropsychology, Volume 9*. Elsevier, Amsterdam, Netherlands, 1994.

[75] Forrester Research. Why most web sites fail. http://www.forrester.com/Research/ReportExcerpt/0,1082,1285,00.html, 1999.

[76] Louis Rosenfeld and Peter Morville. *Information Architecture for the World Wide Web*. O'Reilly & Associates, Sebastopol, CA, 1998.

[77] D. Rubine. Specifying gestures by example. In *Computer Graphics: ACM SIGGRAPH*, pages 329–337, 1991.

[78] Darrell Sano. *Designing Large-scale Web Sites: A Visual Design Methodology*. Wiley Computer Publishing, John Wiley & Sons, Inc., New York, 1996.

[79] Eric Saund and Thomas P. Moran. A perceptually-supported sketch editor. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Drawing and Sketching, pages 175–184, 1994.

[80] Bill N. Schilit, Gene Golovchinsky, and Morgan N. Price. Beyond paper: Supporting active reading with free form digital ink annotations. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems*, volume 1 of *Reading and Writing*, pages 249–256, 1998.

[81] Andrew Sears. AIDE: A step toward metric-based interface development tools. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Evaluation, pages 101–110, 1995.

[82] Nathan Shedroff. Recipe for a successful web site. http://www.nathan.com/thoughts/recipe, 1999.

[83] Ben Shneiderman. Designing information-abundant websites. Technical Report CS-TR-3634, University of Maryland, College Park, April 29, 1996.

[84] Jared M. Spool, Tara Scanlon, Will Schroeder, Carolyn Snyder, and Terri DeAngelo. *Web Site Usability: A Designer's Guide*. Morgan Kaufmann Publishers, Inc., San Francisco, 1999.

[85] Terry Sullivan. Reading reader reaction: A proposal for inferential analysis of web server log files. In *Proceedings of the Human Factors and the Web 3 Conference*, Practices & Reflections, June 1997. Available from http://www.uswest.com/web-conference/index.html.

[86] Maarten van de Kant, Stephanie Wilson, Mathilde Bekker, Hilary Johnson, and Peter Johnson. Patchwork: A software tool for early design. In *Proceedings of ACM CHI 98 Conference on Human Factors in Computing Systems (Summary)*, volume 2 of *Late Breaking Results: Support for Design: Experiments and Cybertools*, pages 221–222, 1998.

[87] Lynn D. Wilcox, Bill N. Schilit, and Nitin "Nick" Sawhney. Dynomite: A dynamically organized ink and audio notebook. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Enhancing, Finding, & Integrating Audio*, pages 186–193, 1997.

[88] W. Zachary, J.-C. Le Mentec, and J. Ryder. Interface agents in complex systems. In C. N. Ntuen and E. H. Park, editors, *Human Interaction With Complex Systems: Conceptual Principles and Design Practice*. Kluwer Academic Publishers, 1996.