SKINNI: The Smart Kiosk Information Navigation and Note-posting Interface

by

Tyler Bruce Horton

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2004

Author . . . .
Department of Electrical Engineering and Computer Science
August 17, 2004

Certified by . . . .
Howard E. Shrobe
Principle Research Scientist
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

SKINNI: The Smart Kiosk Information Navigation and Note-posting Interface

by

Tyler Bruce Horton

Submitted to the Department of Electrical Engineering and Computer Science
on August 17, 2004, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

SKINNI is an application that allows users to access, annotate, and share contextually relevant information from smart digital kiosks in public spaces. These platforms will not only enable information exchange from public spaces, but will also support more complex tasks like enterprise integration, informal collaboration, and public need servicing. To test the effectiveness of our system, we deployed SKINNI on four kiosks throughout MIT CSAIL. Usage statistics that we have gathered suggest that SKINNI has been well-received by both lab members and visitors. In short, SKINNI has contributed to the utility of ubiquitous computing in transitional spaces like elevator lobbies, hallways, and lounges.

Thesis Supervisor: Howard E. Shrobe
Title: Principle Research Scientist

Acknowledgments

None of my academic pursuits would have been possible without the loving support of my family. I especially want to thank Kasey, my wife, for her unending patience and unmatched compassion. She truly is my reason for being. Also, my parents deserve many thanks for the countless quantity of time, money, and inspiration they have provided for my education.

SKINNI originated as a group project for the User Interface Design and Implementation course. The group consisted of Ellie Boyle, Max Van Kleek, and myself. I cannot express how much I appreciate the invaluable work they contributed to the development, testing, and installation of SKINNI. Simply put, SKINNI could not exist without the remarkable efforts they both made. Also, I want to thank Rob Miller and Jamie Teevan for teaching such an excellent course.

Clearly, I must thank all the members of the AIRE research group for four years worth of molding me into a better programmer and a better person. My advisor, Howie Shrobe, has been a constant source of encouragement and sound advice. Stephen Peters has aided me daily with his encyclopedic knowledge of obscure functions in Linux, Perl, Java, SQL, and many other computer systems. Meanwhile, Kevin Quigley has provided support and entertainment that have both been vital to my success at MIT. Also, I must thank Krzysztof Gajos for looking past my inexperience to my potential as he hired me at the beginning of my sophomore year. To everyone else in AIRE, I will never forget you or the impact you have had on my life. I hope we can work together again, someday.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

SKINNI, the Smart Kiosk Information Navigation and Note-posting Interface, is an application for touch screen kiosks that promotes communication and informal collaboration within an organization. This application was motivated by the effort to provide a platform that extends ubiquitous computing[1] into transitional spaces. Specifically, SKINNI provides users with a natural and efficient interface for accessing, annotating, and sharing relevant information from smart digital kiosks located in public-gathering spaces.

This thesis provides a complete view of SKINNI's design, implementation, and deployment. This introductory chapter motivates the need for smart digital kiosks, provides background on kiosk development at MIT, and outlines SKINNI's overall contribution. Chapter 2 provides a system overview of SKINNI, including its goals, design process, component structure, and user interface. Following the design overview, chapter 3 presents SKINNI's current implementation and deployment at MIT. Finally, an examination of the conclusions reached during the construction of this system are given in chapter 4. Appendices follow that provide detailed information and discussion about the hardware make-up of smart digital kiosks, touch screen interaction considerations, and selected screenshots of the SKINNI interface.

[1]The field of ubiquitous computing (also called "pervasive computing") is concerned with providing natural human–computer interaction through intelligent systems embedded in the environment. Mark Weiser, founder of the field, noted that computers should "weave themselves into the fabric of everyday life until they are indistinguishable from it." [23]

1.1 Public Information Access Methods

As people move about in the world, they demand information relevant to their current location and situation. Typically, people will attempt to retrieve the information they seek by utilizing a public information access method such as a sign or mobile device. Signs and other passive information interfaces provide geographically relevant content to nearby users, but might not relay necessary information. Personal mobile devices are more flexible than passive interfaces, but require that the user carry the device, are size-constrained, and suffer from limited perceptual bandwidth to the user. Smart digital kiosks situated in appropriate locations provide a platform that improves on the weaknesses of these other information access methods while simultaneously building upon their beneficial features. Additionally, kiosks are equipped with a rich network of sensors and knowledge sources that allow applications to infer the context of their users.

1.1.1 Passive Information Interfaces

Traffic signs, posted flyers, and billboards are some common examples of passive information interfaces in use today. Typically, the information provided on these signs is geographically relevant since their locations are specifically chosen by those who posted them. Users need to be close enough to read a sign in order to access the information it provides. While the ability to access the information from only one area might be construed as a disadvantage, it is worth noting that the user effort required to access the information is minimal and that the scope of the information is likely restricted to the accessible area anyway. However, a significant drawback to passive information interfaces is the inability to interact with the information. Specifically, users cannot search, edit, personalize, or download the posted information. Also, without proper maintenance, the information posted on many passive interfaces quickly becomes outdated.

1.1.2 Personal Mobile Devices

Many people carry personal mobile devices such as laptops, PDAs, and cell phones. Each of these devices offers a distinct range of computational functionality, yet all provide mechanisms for information retrieval. People store information on these devices that will be useful to them as they move about in the world. However, users cannot anticipate all the information they will need access to, especially as their context—location and situation—changes. For example, inaccurate directions may cause someone's context to change from "according to plan" to "lost". In this case, the user must access some network to fetch proper directions or other required information. Unfortunately, users may find that relying on their mobile devices to access contextually relevant information is not always convenient, efficient, or even possible. To summarize, these devices sometimes suffer from one or more of the following drawbacks: limited screen real estate, awkward user interfaces, short battery life, costly subscription fees, and unavailable network service.

1.1.3 Smart Digital Kiosks

Smart digital kiosks, like the ones shown in figure 1-1, are information interfaces that automatically collect and manage knowledge, infer user context, and provide a natural and efficient interface to relevant information. Additionally, smart kiosks are designed to avoid the previously outlined pitfalls associated with personal mobile devices and passive information interfaces. Table 1.1 provides a summary of comparisons between smart kiosks and the other information interfaces. The comparison points that are most significant to smart kiosks are context awareness, availability, augmenting and sharing information, information exchange, and multiple user support.

Context Awareness

Smart digital kiosks are designed to deliver information which is relevant to a user's current context. The kiosk's location—part of this context—is known *a priori*. Signs and other passive interfaces also deliver content based on their situated geography.

Figure 1-1: Smart Digital Kiosk Designs
The center frame shows the most recent kiosk design. This form-factor is currently deployed in three elevator lobbies at CSAIL. The original prototype kiosk (left) features two displays, a camera, a sound deflection shield, and a linear array microphone. The large display kiosk (right) provides a unique user interface device known as IGOR [13]. This robotic mask interprets users' gestures and translates them into mouse events.

However, kiosks are also able to make inferences about a user's current situation—another major part of context—based on dynamic knowledge sources and rich sensory input. For example, a kiosk located inside an electronics store might infer that a user carrying a Bluetooth® device could be interested in learning about new accessories for that device. Kiosks are able to provide a level of awareness beyond what personal devices can. For personal device to use context, the user must explicitly provide it with situational knowledge. This extra user effort is less transparent and thus inconvenient, error prone, and counteractive to the benefits of awareness. Additionally, most modern mobile devices either do not possess or utilize knowledge about where they are physically located, although this characteristic is likely to change in the near future.

Availability

Availability refers both to the presence of an information interface and its ability to access information. For passive interfaces, the information is always available as long

	Passive Information Interfaces	Personal Mobile Devices	Smart Digital Kiosks
Context Aware	Location Only	Explicit	Inferred Situation; Known Location
Presence	Where installed	User-dependent	Where installed
Network Connection	N/A	Typically Wireless	Typically LAN
Searchable	No	Yes	Yes
Editable	No	Yes	Yes
Information Exchange	Transcription	Device–Device or Device–Network	Kiosk–Device or Kiosk–Network
Multiple Users	Yes	No	Yes
Private	No	Yes	No
Personalizable	No	Yes	Yes
Portable	No	Yes	No
Display Size	Medium–Huge	Small–Medium	Medium–Large
Power Requirements	None	Battery or Adapter	Outlet
User Cost	None	Expensive	None
User Effort	Approach and Read	Potentially Complex	Minimal
Memory	Single Image	Unlimited	Unlimited

Table 1.1: Information Access Method Comparison Table

This table outlines the three information access methods (passive information interfaces, personal mobile devices, and smart digital kiosks) and compares their characteristics point-by-point.

as the sign itself is physically present. For digital devices, accessing information is dependent not only on the physical presence of a device, but also on network requirements. Of course, the presence of a personal mobile device is entirely dependent upon the user, while kiosk availability requires separate installations. Smart digital kiosks may someday be present the same way public telephones are now, thus providing sufficient availability.

Augmenting and Sharing Information

Using a personal mobile device to share information with other people is trivial provided that a reasonable network is available. Generally though, the information must be targeted to a specific set of known individuals. However, with information kiosks, users can leave messages for unknown people who are assumed to share context by occupying the same area, even at different times. For example, a user may wish to augment a digital train schedule to let others know that a certain train typically runs late. This type of functionality is available with passive interfaces as well, but is typically perceived as graffiti rather than useful information.

Information Exchange

Given a viable network connection, downloading and uploading information with a personal mobile device is fairly standard. Also, many devices provide direct communication with other devices through IR, Bluetooth®, WiFi, or other protocols. Kiosks provide the same type of exchange mechanisms, allowing information to be posted to the kiosk network or downloaded from the network onto mobile devices. For passive interfaces, transcribing its contents is one of only a few options for retaining its information.

Multiple User Support

Personal mobile devices are not designed to support interactions with multiple people at the same time. Rather, these devices are intended to facilitate the needs of one person for most of its lifetime. On the other hand, kiosks are used by many people at different times and could provide support for multiple interactions at once. This functionality would facilitate tasks like informal collaboration and information sharing. With passive interfaces, nothing prevents multiple users from reading the same sign, however interaction beyond reading is not typically supported.

1.2 Interests behind SKINNI

Ubiquitous computing has been an area of active research at MIT for many years. The recent interest in using kiosks to further this field has captured the attention of some research groups and sparked collaboration between them. This collaboration has lead to the development of smart kiosks and SKINNI—the face of smart kiosks.

1.2.1 Project Oxygen

Project Oxygen is a partnership between the Computer Science and Artificial Intelligence Laboratory (CSAIL) at MIT and several industrial research sponsors[2] to develop technology for ubiquitous computing. Research groups in CSAIL focus on a wide range of ubiquitous computing technologies, such as intelligent environments, perception, knowledge access, automation, communication, distributed processor architectures, etc. [1]

1.2.2 AIRE: Agent-based Intelligent Reactive Environments

The Agent-based Intelligent Reactive Environments group (AIRE)[3] develops intelligent environments within Oxygen. The primary focus of AIRE has been outfitting spaces like conference rooms and offices with ubiquitous computing technology. Recently, AIRE has also explored equipping transitional spaces—high-traffic areas such as lobbies and hallways—with technology similar to other intelligent environments. AIRE has demonstrated that situated computing platforms like smart kiosks provide a strong base for incorporating ubiquitous technology into public spaces. [2]

1.2.3 OK-net: Oxygen Kiosk Network

The Oxygen Kiosk Network project (OK-net)[4] is responsible for all aspects of smart kiosk development within CSAIL, including the physical structure of the kiosks, dynamic knowledge maintenance, communication and networking, and application development. While kiosk development was originally limited to the AIRE research group, OK-net is a multi-group collaboration to develop kiosks and applications that fully integrate the latest advances in ubiquitous technology like perception, knowledge access, communication, and intelligent environments. [3]

[2]Project Oxygen sponsors include Acer, Delta Electronics, Hewlett-Packard, Nippon Telegraph and Telephone, Nokia Research Center, and Philips Research. Oxygen is also sponsored by the Information Processing and Technology Office (IPTO) of the Defense Advanced Research Projects Agency (DARPA) of the U.S. Department of Defense.

[3]Formerly known as the Intelligent Room project.

[4]Formerly known as project Ki/o: kiosk gateway for interaction and observation.

1.3 Benefits of SKINNI

The purpose of SKINNI is to facilitate the exchange of contextually relevant information from public spaces. Realizing this goal allows SKINNI to support valuable functions like enterprise integration, informal collaboration, and public-needs servicing. In the future, SKINNI may be augmented to support additional functions like advertisement, entertainment, and kiosk-to-kiosk synchronous communication.

Enterprise Integration

Enterprise integration is the process of coordinating an organization's efforts through a shared platform. This coordination often takes the form of an internal corporate website or mailing list designed to convey official organization-related news. Another powerful tool for enterprise integration is groupware—software that allows multiple people to work on the same project in a shared virtual environment.

SKINNI is not designed to replace these methods for enterprise integration. Rather, SKINNI augments coordination efforts by providing information that may be related to an employee's projects—a significant component of overall context. For example, on the way to a group lunch, an employee uses SKINNI and learns that a well-respected author is giving a lecture at the local university. At lunch, the employee discusses the upcoming talk and the benefits it may provide for various members of the group. After lunch, on the way back to their workspaces, the group members access more detailed information about the lecture through SKINNI.

Informal Collaboration

Interorganizational collaboration is often prompted by an informal conversation between two or more parties in a transitional space. A kiosk running SKINNI may spark this type of informal collaboration by either attracting people to the space or by providing information that matches the shared context of people in the space. For example, two researchers waiting for an elevator notice that SKINNI contains information about a topic that interests both of them. This might cause the two researchers to discuss the topic and how it relates to their work. Even after the elevator arrives and they leave the kiosk, the researchers will continue to discuss the issue and perhaps decide to collaborate on it in the future. [5]

In addition to prompting conversation directly between people, SKINNI may support asynchronous collaboration by allowing users to post queries and responses that apply to some shared context of people in the same location. For example, a student may begin a discussion about a reading assignment on a kiosk in the hallway outside the classroom. Other students in the same class (shared context) may notice the posting and participate in the discussion. This approach is most beneficial for academic pursuits where discussion provides insight rather than concrete answers.

Public Services

Aside from business and academic functions, SKINNI can also provide useful services to the general public. For example, a kiosk running SKINNI in an airport would be able to provide information about flight statuses, local attractions, shuttle schedules, ticket pricing, etc. Clearly, members of the public benefit from the ability to quickly access this type of information. Travelers with mobile devices would be able to download any pertinent information and retrieve it at a later time. Additionally, the ability to annotate information can allow people to offer advice or additional information. In general, anyone can benefit from having easy access to relevant information while occupying a transitional space.

Chapter 2

User Interface Design

When designing SKINNI, the OK-net development team set out to create a user interface that affords natural and efficient access to relevant information. This chapter considers SKINNI's fulfillment of this design goal in the context of a case study: deployment within CSAIL. Two factors motivated our decision to use CSAIL as a test site for SKINNI. First, CSAIL is the product of a recent merger (between the former Laboratory for Computer Science and the Artificial Intelligence Lab) so would benefit from a platform that aides informal collaboration and enterprise integration. Second, the building occupied by CSAIL (the Ray and Maria Stata Center, designed by Frank Gehry) features a unique architectural style that will attract many visitors, thus providing an audience for public services like intra-building navigation.

We began the design process by outlining the criteria SKINNI must satisfy in order to provide an effective user interface. Next, we validated the need for SKINNI within CSAIL by surveying some of its members. We then designed, tested, and evaluated three graphical user interface prototypes, allowing us to make iterative improvements to our design. We also examined the unique aspects of touch screen and speech interactions through additional user testing. Finally, after many preliminary designs, we are able to articulate the final arrangement of the SKINNI user interface.

2.1 Design Criteria

We identified several design criteria that SKINNI must satisfy in order to provide an effective user interface within CSAIL. First, SKINNI must be designed to meet the needs and expectations of its intended users while still respecting their differences, limitations, and restrictions. Next, SKINNI should place few requirements on its users; it should be quick to learn, easy to use, and accessible with minimal prerequisites. Additionally, to avoid some common user interface design mistakes, we applied some general usability heuristics to SKINNI.

2.1.1 Target Audiences

SKINNI is designed to address the needs of two populations within CSAIL: lab members and visitors. A lab member is any person who works in CSAIL and could potentially benefit from SKINNI's interorganizational communication on a daily basis. A lab visitor is any member of the public who has come to CSAIL either to see the building, one of its occupants, or a lab-sponsored event. Typically, members from both of these groups will be seeking different types of information from SKINNI.

CSAIL Members

Undergraduate and graduate students, faculty, research scientists, and administrative and technical support staff are some of the groups of people that compose the set of CSAIL members. These people have different backgrounds (nationality, age, gender, computer expertise, research interests, etc.) and will have a wide range of expectations for what SKINNI will allow them to do. Generally, however, CSAIL members will be using SKINNI to gather information that will aide them on a day-to-day basis as they work in the lab. For example, students may be primarily interested in learning about class-related information while faculty want to learn more about what research other groups are conducting. System administrators would primarily be interested to find out about computer update notifications and other related topics. Of course, all lab members would be able to access any of these types of information if they choose.

CSAIL Visitors

Lab visitors consist of corporate sponsors, documentation personnel (film crews, journalists, etc.), non-CSAIL MIT students, friends and family of lab members, architecture enthusiasts, and tourists. These people would most likely benefit from information pertaining to the physical and organizational structure of the lab. Specifically, they would likely seek a directory to help locate lab members and a map to navigate to a given location. Of course, this type of information could be useful to CSAIL members, especially for those new to the lab.

2.1.2 User Prerequisites

SKINNI is designed to be usable and useful to the vast majority of its target audiences. That is, first-time and infrequent users can still access relevant information through SKINNI without too much difficulty. Additionally, due to the simplicity of the interface, there is not a long learning period for new users to gain an expert level of experience.

When designing SKINNI, we sought to minimize the requirements our system imposed on its users. This effort involved providing intuitive modes for interacting with SKINNI and the information it provides. The system is arranged so that anybody can approach a kiosk, interact with SKINNI, gather information, and simply leave the kiosk.

Touch Screen Freedoms

Touch screens provide a direct and natural means of interacting with information. With SKINNI running on smart kiosks, users need only approach the kiosks and touch on-screen information summaries to access details. Users can navigate through the SKINNI interface by using their fingers to generate mouse events like clicking and dragging. Text entry through the touch screen is not currently supported.

Certain types of touch screens require a stylus for interaction. These screens would improve screen real estate use (by using smaller graphical widgets) and may provide a method for text entry (via pen-based hand writing recognition). However, we decided not to employ this type of hardware for two reasons. First, using a stylus is not as natural or direct as using your finger. Second, if users require a stylus to interact, then a lost stylus greatly reduces the usability of a kiosk.

Speech Freedoms

Speech is an extremely natural form of human communication. Therefore, SKINNI should accommodate speech interaction and not force users to access information solely through a keyboard and touch screen interface. SKINNI is equipped with a speech recognizer (see section 3.1.5 for details) that enables users to vocally access information.

The SKINNI speech interface removes two requirements present in most existing speech interfaces. First, the speech recognition engine is speaker-independent, meaning that users are not required to train the kiosks to respond to their unique voices. Second, the microphone used by the kiosks is an array microphone, thus freeing users from handling a traditional microphone or wearing a headset.

Approach Freedoms

SKINNI is designed to minimize user requirements associated with approaching and leaving kiosks. Specifically, users are not required to log into or out of SKINNI to access its information.

When users approach a kiosk, they will find that SKINNI is in an operational state where all parameters are set to system defaults. If a user is identified or authenticated (details on identification and authentication are discussed in section 3.2.3), SKINNI personalizes these parameters as well as portions of its view. After interacting with the kiosk and gathering relevant information, logging out of the kiosk involves simply walking away from it. Once SKINNI asserts that the user's session has ended, the screensaver will appear and simultaneously reset all of SKINNI's session parameters to the system defaults. The screensaver remains active until the next user session is initiated by someone approaching the kiosk.

General Requirements

Unfortunately, deployment of SKINNI within CSAIL has imposed some indirect user requirements on members of the target audiences. For example, SKINNI is English-based, meaning that users who are illiterate or not fluent in English will have difficulty gathering information. However, nothing in SKINNI's system design prevents an implementation configured for any given language, yet no provisions are currently made for simultaneous multiple language support.

Another set of requirements that are indirectly imposed on users, are those pertaining to physiological characteristics of the users. While these limitations do not effect most most members of the target audiences, some users may find difficulty interacting with touch screens (screen height, unsteady hands, etc.), voice interfaces (lost voice, microphone position, etc.), or color-coded widgets (color blind, sight impaired, etc.). Addressing these and other accessibility concerns is left as future work.

2.1.3 Heuristics

Jakob Nielsen [7] and Bruce Tognazzini [20] are usability experts that have developed heuristics to guide user interface design. We applied several of their heuristic techniques to SKINNI and were pleased to find them improve the overall usability of the interface. Heuristics related to anticipation, user control, navigation, consistency, efficiency, and readability were particularly valuable to the development of SKINNI.

Anticipation

In order to build an effective user interface, designers must anticipate the functions that users are most likely to need. This means that controls for common actions should be highly visible and accessible. Tailoring components to meet the needs of each user is one method for improving usability. SKINNI not only makes common controls visible and accessible, but also customizes the presented information based on the current user's context. SKINNI is more usable because it anticipates what information the user needs to access.

User Control and Freedom

While SKINNI provides context-based recommendations, users ultimately have control over what information to access. While accessing this information, SKINNI allows users to "explore" the interface. This flexibility enables users to suspend one task and switch to another without expending undue effort.

Visible Navigation

SKINNI provides an uncomplicated navigation system that allows users to view where they are in the interface, where related information might be, and how they might move from one portion of the interface to another. The interface also clearly guides users through sections of the interface that must be accessed sequentially, while still allowing users to suspend or abort their current tasks.

Consistency and Standards

The SKINNI interface provides a similar view of information regardless of its type. This consistency allows users to infer the function of various graphical controls even if those exact components have not been previously encountered. Also, user familiarity is increased by the use of familiar interface components like tabbed navigation, date-choosing calendar widgets, email dialogs, etc.

Flexibility and Efficiency of Use

Several accelerators are built into SKINNI that allow users to jump from one piece of information to another related piece of information, without having to navigate through undue channels. Some common keyboard shortcuts are also incorporated (arrow key navigation, field tabbing, etc.), but are not accentuated since touch screen interaction is intended to be a dominant input mode compared to keyboard.

Readability

Most information accessed through SKINNI will take the form of text on the screen. This text is crafted to be readable by using appropriate size, font, emphasis, layout, and color scheme.

2.2 Task Validation

We began the design process by validating the need for SKINNI within CSAIL. That is, we ensured that our interface would be used to access information and thus facilitate tasks like community building, collaboration, and knowledge sharing. We performed this task validation by randomly surveying ten members of CSAIL, including eight students and two faculty members.

The survey first asked participants to indicate how well-acquainted they were with their colleagues in various areas of the lab. Specifically, participants were asked to provide the percentage of people they knew who occupied the same floor, adjoining floors, and the rest of the building. The results of this survey showed that lab members were aware of other members in their general vicinity, but knew far fewer—less than ten percent—of people located elsewhere in the lab.

The second part of the survey asked participants to indicate the type of information they might share with the entire lab, the frequency with which they might communicate, and the media used to disseminate. The type of information shared included event notices, discussion topics, lab-specific announcements, and social items. The frequency with which this type of communication occurred varied widely across participants. Media used to share this information was generally restricted to email (lab-wide mailing list) or posting flyers in elevator lobbies and other common areas. Interestingly, several participants noted that these media were not ideal for lab-wide communication due to the "ignorable clutter" they created.

The data from these surveys indicated that SKINNI would fill a real need within CSAIL. Lab members would be able to share information about lab-related and social events and thereby build community. At the same time, kiosks running SKINNI would provide a new medium through which communication on a lab-wide setting need not clutter inboxes and bulletin boards.

2.3 Prototypes

We realized early in the design process that in order to satisfy SKINNI's goals and create an effective user interface, we would have to employ an iterative development process. This iterative approach mitigates large design risks early during development, as opposed to a cascading design process which incurs costly penalties for undetected design flaws. [4, 6]

We designed, tested, and evaluated three SKINNI prototypes before reaching the final design. Each prototype revealed shortcomings or design flaws that were corrected for the next iteration.

2.3.1 Paper Prototype

The first prototype for SKINNI, shown in figure 2-1, consisted of several layers of paper and Post-It Notes™. This approach had some considerable advantages over using a computer to build the initial prototype. First, paper prototypes are faster to make and easier to change: sketching on paper is faster than programming on a computer. Second, even during prototype development, programmers are likely to spend undue time debugging minor code details. With paper prototypes, designers can focus on larger issues without being distracted by smaller ones. Lastly, developers retain only the design from a paper prototype, not its implementation. Discarding a prototype prevents all of its implementation-specific limitations from manifesting in the final system.

We performed a user evaluation of the SKINNI paper prototype with three randomly chosen members of CSAIL. Users manually interacted with the prototype. That is, they used their index fingers as pointing devices and pens for text entry. One of the OK-net developers performed the back-end tasks and updated the interface by removing or applying Post-It Notes™ to the prototype. Each participant was

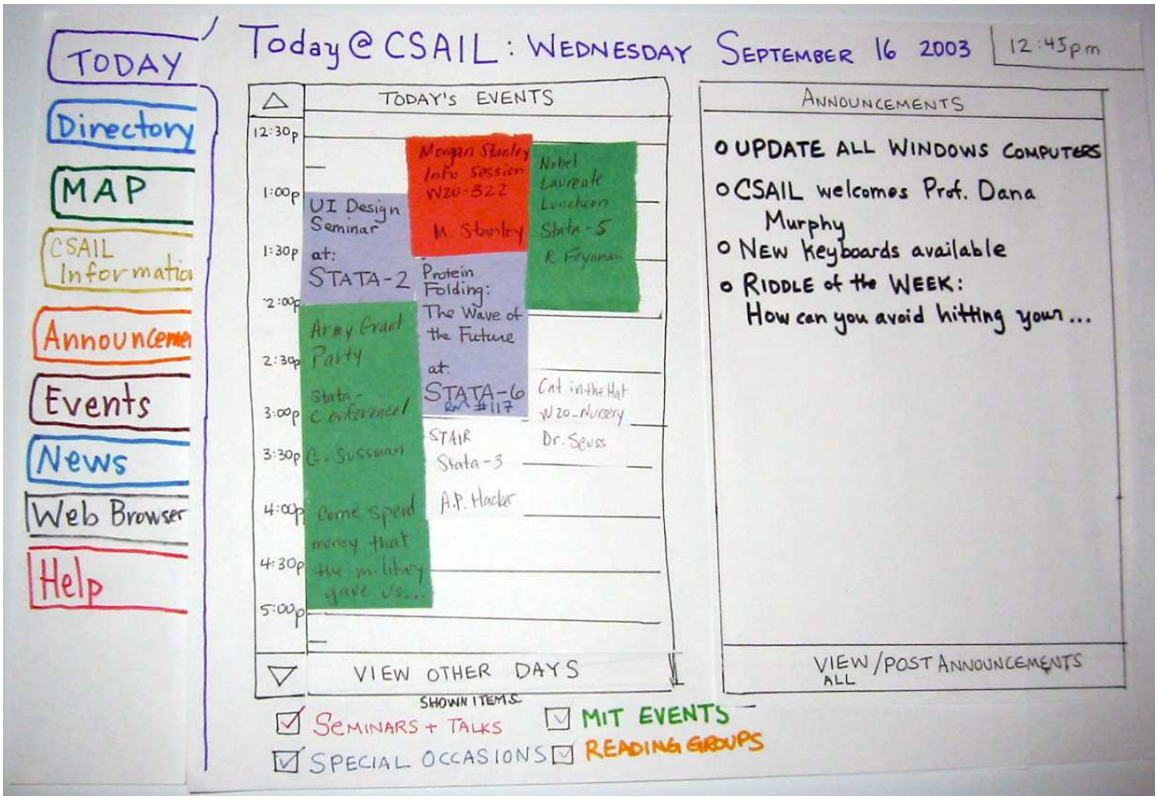

Figure 2-1: Paper Prototype
The first prototype for SKINNI was made with paper for rapid creation, simple modification, and discardable implementation. Users would interact with this prototype manually, using fingers for pointing and clicking input and pens for text entry. During each trial, one of the testers would modify the interface according to user's input.

asked to complete the following tasks devised from scenarios that emulated typical usage:

1. You are a biology student looking for a talk about protein folding that is supposed to start within the hour. Find out where it is going to be and how long it will last.

2. Your colleague has told you that a riddle is posted on SKINNI. You want to see if you can be the first to post the correct answer.

3. There is food left over from your lunch meeting. Let hungry graduate students know that the food is available.

We observed the users performing these tasks and made notes about the prototype's usability shortcomings. These notes are summarized in table 2.1. Specifically, the table provides a list of the major interface design problems uncovered and proposed solutions for incorporating into later designs.

2.3.2 HTML Prototype

The HTML mockup of SKINNI shown in figure 2-2 served as our second prototype. This approach presented a high fidelity view of the interface while omitting the application's underlying functionality. The decision to use HTML (instead of a general-purpose programming language) for a front-end only prototype was motivated by three factors. First, developing a realistic looking interface in HTML requires less time and effort than with a programming language. Second, since HTML is rendered in standard web browsers, we could rely on the browser's built-in navigation functions rather than implement our own. Lastly, the implementation of the HTML prototype would be discarded for the same reason as the paper prototype: to build upon the improved design while removing all implementation-specific limitations.

Due to time constraints, this prototype did not fully incorporate the corrections to design flaws identified with the paper prototype. However, observing users interact with this prototype revealed some other usability concerns. Table 2.2 lists the non-duplicate problems we uncovered as well as proposed solutions for how to fix them. Users completed the same three tasks as with the paper prototype.

2.3.3 Java Prototype

Our final SKINNI prototype was implemented in Java so that we could easily convert the prototype into the final version of the system. A screenshot of this prototype is provided in figure 2-3. We incorporated most of the solutions to problems identified during the previous prototypes and conducted another round of testing. However, this time we had users complete the following six tasks rather than the previous three:

1. You own a Linux machine and heard that a new security vulnerability was discovered in the Linux kernel. Find the security announcement that contains

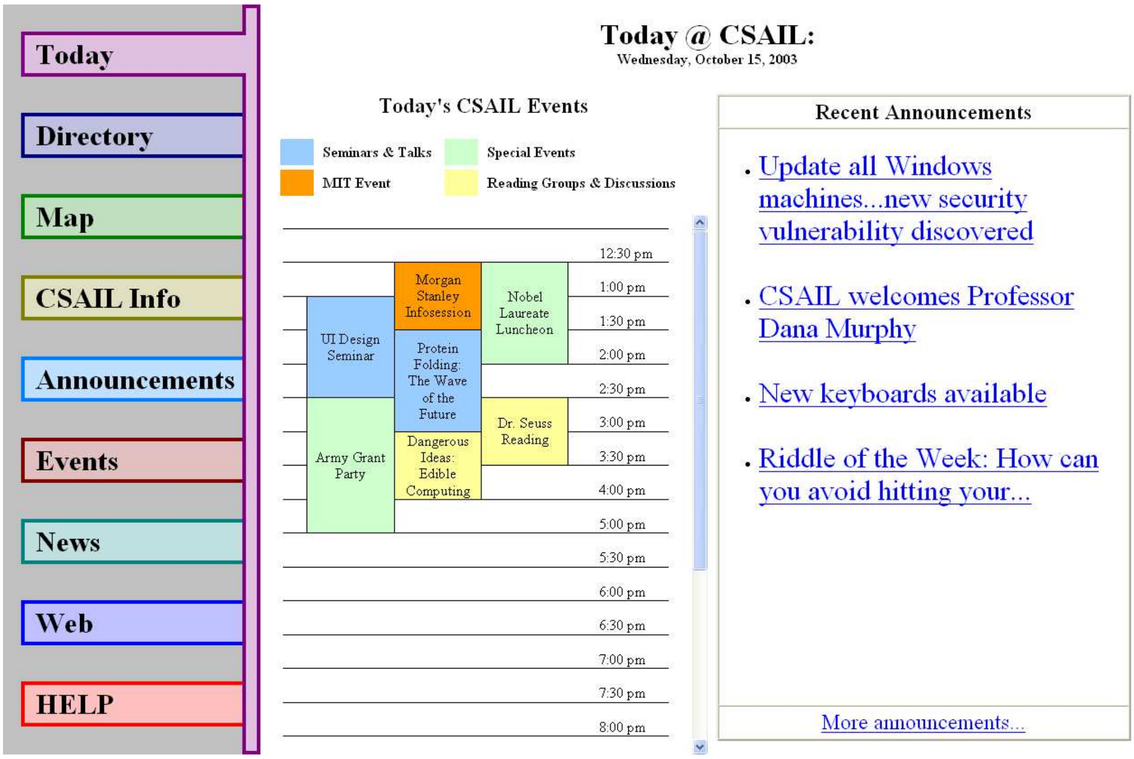

Figure 2-2: HTML Prototype
This HTML prototype of SKINNI provides a higher fidelity view of SKINNI, but still lacked most back-end functionality. The purpose of this prototype was to assess some of the computer-based usability issues present in the system.

the recommended course of action.

2. You are late to office hours with your 6.825 TA, Rodney Brooks. Find his office number and where it is located on the floor.

3. It's around noon on Friday and you're wondering whether GSL (Graduate Student Lunch) is being served this week. Determine whether it is and find out what's for lunch.

4. You hear that there's a synthetic biology seminar some time next week. Find out exactly when and where it will be and send yourself a reminder.

5. Your group meeting just concluded and you have leftover food that you want hungry graduate students to have. Let them know it's available.

6. There is a discussion about changing the name for CSAIL, but you like the way it is. Find out what others' suggestions are and make your own opinions known.

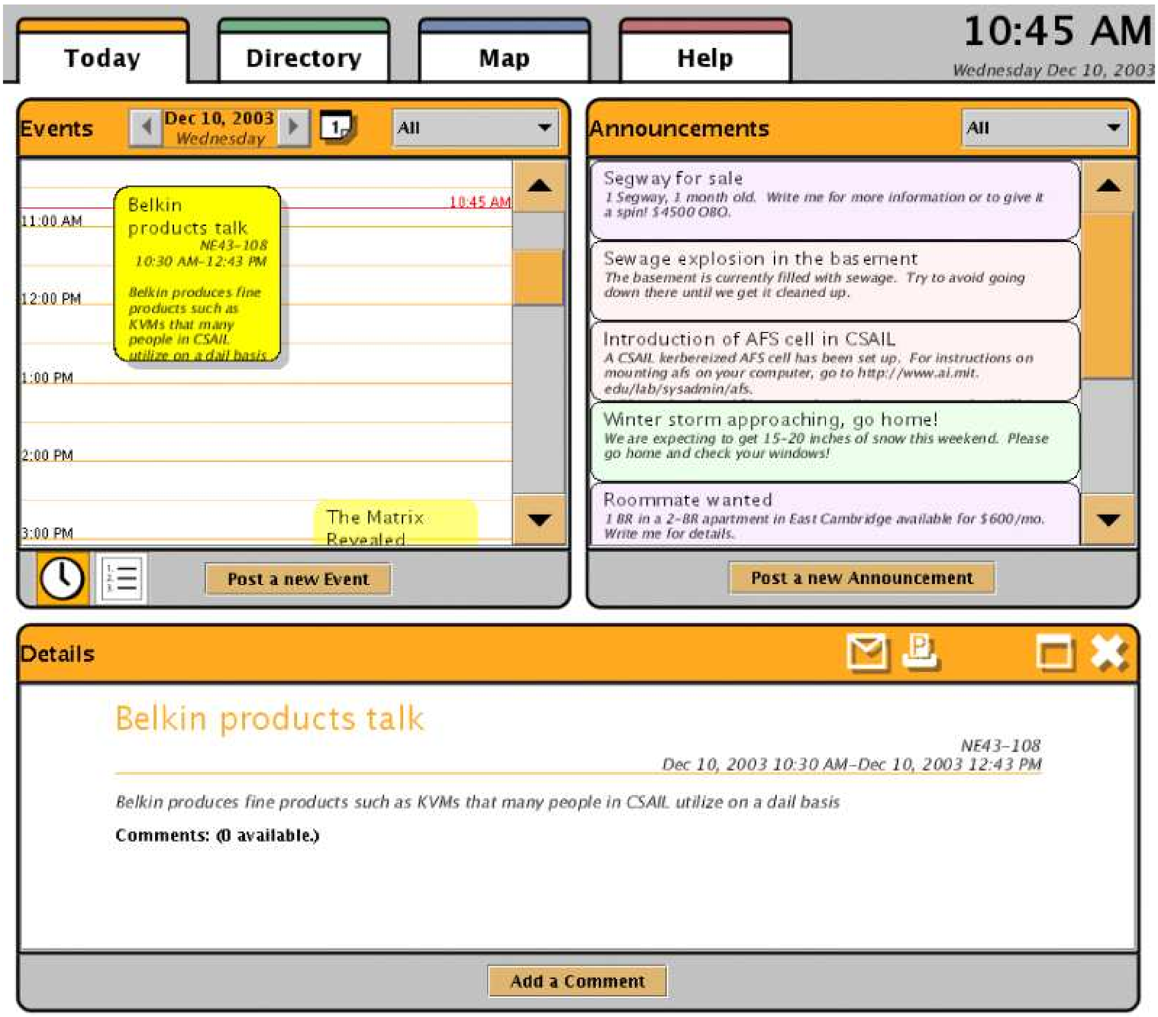

Figure 2-3: Java Prototype

By implementing this SKINNI prototype in Java, we were able to test a high fidelity front-end and back end at the same time. Major portions of this prototype's design (both implementation and interface) are present in the deployed version of SKINNI.

Again, we observed users performing the sample tasks and recorded the difficulties that they experienced while using the interface. These problems and their proposed solutions are provided in table 2.3. After incorporating all of these changes, plus those not yet addressed from previous prototypes, we proceeded to the formulation of SKINNI's final interface design.

2.4 Touch Screen Considerations

Nearly all graphical interfaces must trade off between screen real estate devoted to information display and interaction controls. With touch screen kiosks, where the primary input mode is the user's finger, standard component sizes do not facilitate efficient use. Rather, widget dimensions must be increased so that users can consistently touch on-screen controls. To this end, SKINNI's interface features oversized buttons, scrollbars, lists, and tables. However, enlarging these components too much limits available screen real estate for content delivery.

We conducted a user study to determine how SKINNI's graphical components should be sized so that as much relevant information could be visible at once. The keystone of this study was measuring the smallest target that users could consistently touch. Details about the user study, the hypotheses tested, and the interpreted results are provided in appendix B.

The results of our study show that the smallest button that a user can consistently touch is 0.625×0.625 inches (60×60 pixels on a 96 d.p.i. screen). SKINNI ensures that commonly accessed graphical controls conform to this minimum size requirement where possible.

2.5 Speech Interface Considerations

SKINNI is equipped with a speech interface that allows users to quickly access specific pieces of information with minimal GUI interaction. Before building the SKINNI speech interface, we had to construct a grammar that conformed to how users commonly phrased their information requests. Specifically, we wanted to know how people would ask for office locations, telephone numbers, email addresses, and other related information. [19]

We conducted a survey of random CSAIL members to determine how they might phrase their information requests. Combining the responses we received allowed us to create a natural grammar for seeking directory and map information through SKINNI.

An abbreviated version of this grammar is provided in table 2.4.

2.6 Graphical Interface Details

The SKINNI graphical user interface (GUI) provides means for interacting with contextually relevant information in a natural, direct, and efficient manner. Familiar components like tabs, search fields, and dialog windows allow users to quickly grasp how the interface works. At the same time, novel component structures let users quickly navigate between related information items. Figure 2-4 provides a screenshot of SKINNI during typical use in CSAIL. Additional larger screenshots of SKINNI are provided in appendix C.

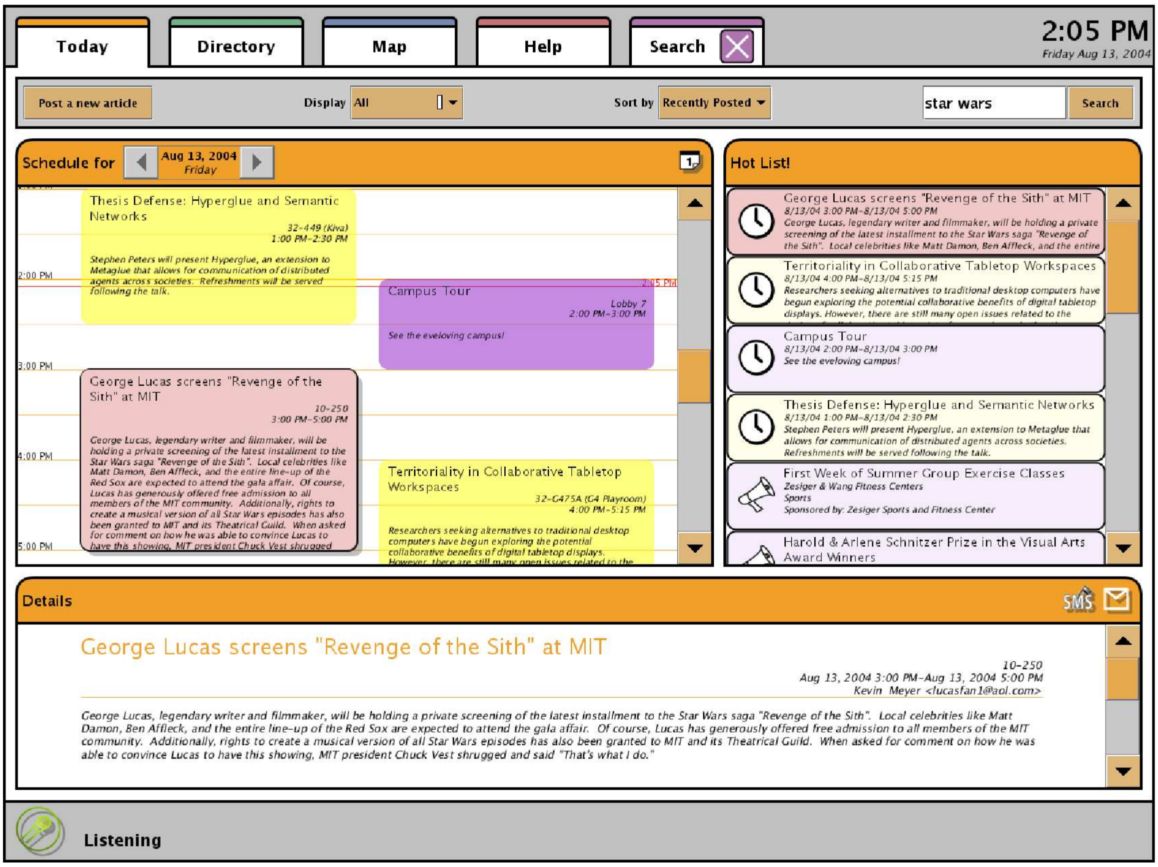

Figure 2-4: SKINNI Screenshot
This overview screen shot of SKINNI shows how the interface might appear after a user selects a relevant information item. Detailed portions of this screenshot appear in some of the remaining figures in this chapter.

2.6.1 Tabs and Panes

The SKINNI GUI adopts a tabbed interface for easy navigation and separation of logically distinct sets of information. Each tab links to an associated pane that presents related information. The case study deployment of SKINNI features tabs labelled “Today”, “Directory”, “Map”, “Help”, and “Search”. The pane associated with the “Today” tab is currently visible in figure 2-4.

All tabs are contained within a horizontal bar that also contains a clock. As with a standard tabbed interface, users switch between tabs by clicking (touching) them. Users cannot reorder tabs. Only tabs with a “X” button like the “Search” tab are able to be closed (removed from the tab bar) by the user. Figure 2-5 provides a detailed view of the tab bar.

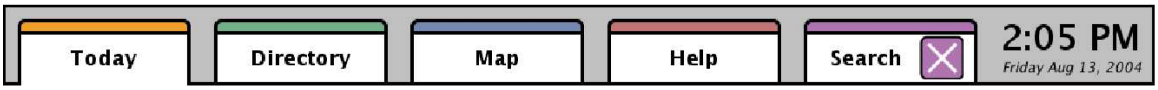

Figure 2-5: The SKINNI Tab Bar
The tab bar provides a means of navigating through logically distinct sets of information within SKINNI. This tab bar shows that the “Today” tab (and associated pane) is selected.

2.6.2 Windows

Panes are occupied by windows—graphical containers comprised of an optional head, required body, and optional foot—like the one show in figure 2-6. A window’s head contains a title and information control components. The title indicates the intended purpose of the window. The controls function on information strictly contained within the window. The window’s body provides the main view of information. Components controlling the view, content, or navigation of information may also be present in the body of a window. All components contained within a window’s body are placed into a scrollpane so that no portion of the view is inaccessible. The foot of a window contains controls that perform operations on the window itself, not the information displayed within the window.

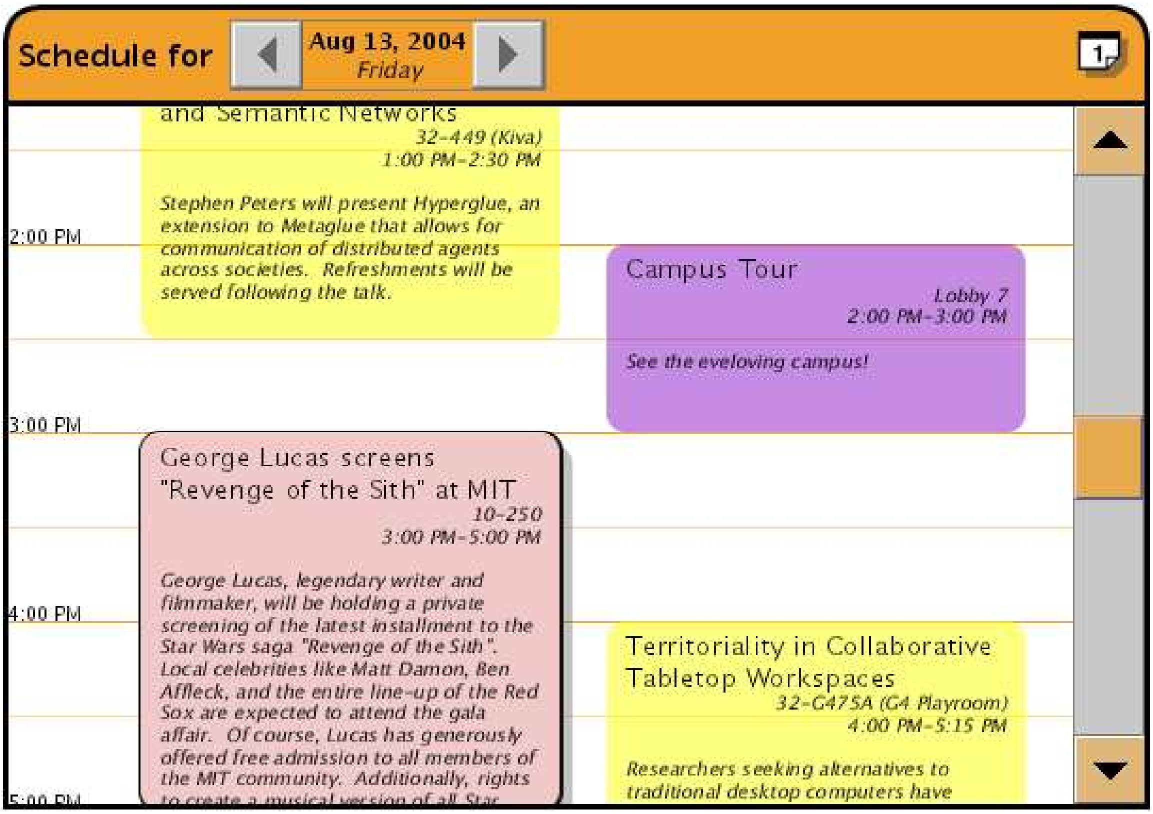

Figure 2-6: SKINNI Window
This SKINNI window is located in the "Today" pane and provides summaries of a given day's events. Controls in the window's head allow users to change the date displayed. Also, each rounded rectangle in the window's body is a control that summons a detailed view of the summary. No controls are present in the window's foot, so it has been removed to conserve screen real estate.

Windows are immobile, non-resizable, and generally non-overlapping. Two factors motivated these decisions. First, user testing indicated that "grabbing" the edge of a window on a touch screen was extremely difficult. Second, we discovered that the interface could quickly become confusing if windows were not arranged in an orderly manner, especially when windows obscured portions of others.

2.6.3 Toolbars

Each pane can contain a type of window called a toolbar, which provides utilities for interacting with available information. For example, users can filter information in the "Today" pane based on its scope: lab event, social posting, MIT-wide news, etc.

Toolbar windows do not have a head or a foot. Figure 2-7 shows the toolbar for the "Today" pane.

Figure 2-7: "Today" Tool Bar

The "Today" tool bar provides widgets that allow users to post, filter, sort, and search. Generally, toolbars provide utilities for interacting with information in that pane.

2.6.4 Dialogs

When SKINNI requires information from the user, it creates a transient window called a dialog. Users complete the form located in the body of the window to respond to SKINNI's query. At any time, the user can dispose the dialog by choosing the "Cancel" button located in the dialog's foot. This action closes the dialog and causes SKINNI to return to its exact state prior to the dialog's launch. If a user attempts to dispose a dialog through a "Submit" or similar button, then the dialog will close and take effect only if the user has provided all required information. Otherwise, an error message appears and the dialog continues to prompt the user until it is answered or canceled.

Like other windows, dialogs are immobile and non-resizable. However, dialogs are positioned in the center of the pane over top of the other windows. When a dialog is launched all other controls within the pane are disabled until the dialog is disposed. This "pane-modal" approach is favorable because it allows users to respond to SKINNI's query while simultaneously being able to access related information in other panes. Particularly, we felt it was important that users could always access the "Help" pane even if a dialog were launched on another pane.

Figure 2-8 shows a section of a pane with a dialog launched. The area of the pane outside the dialog is grayed-out to indicate that its controls are disabled.

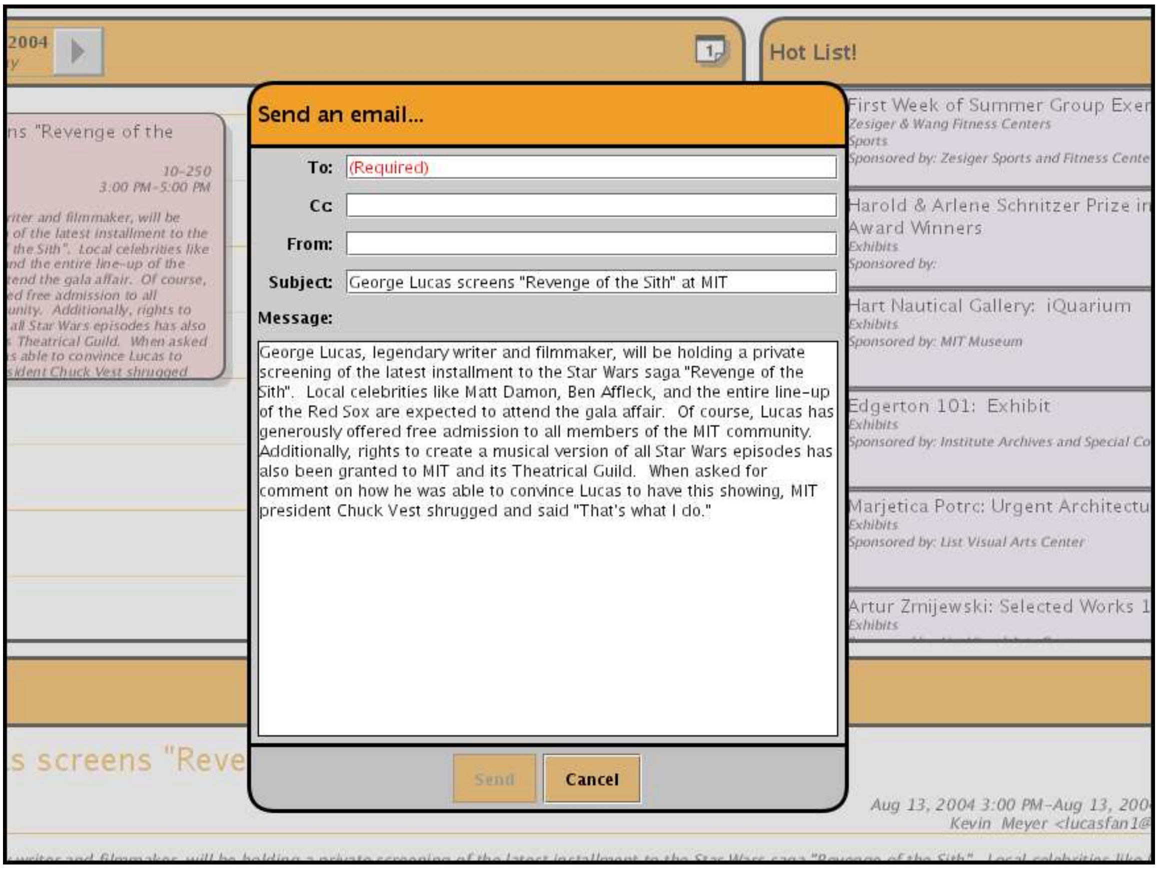

Figure 2-8: SKINNI Dialog

Dialogs like the one shown here allow users to carry out functions where SKINNI requires a considerable amount of information from the user. Here, SKINNI needs to know how it should send an email notification for the selected event. Once the user provides the required "To" field, the "Send" button in the dialog's foot will be enabled. Meanwhile, all controls within the pane that are not part of the dialog are disabled until the dialog is disposed.

2.6.5 Dataviews

Interacting with information is most directly accomplished through the use of effective dataviews. We have adapted four traditional dataviews to meet SKINNI's unique needs: a timeline, details view, SKINNI list, and SKINNI table. All of these reusable components provide improved methods for viewing and navigating through information.

Timeline

The SKINNI timeline dataview shown in figure 2-9 is used to showcase information pertaining to events with well-defined durations. Each of the rounded rectangles represents a single article of information. Clicking on a rectangle selects it and the associated information article. The selection is graphically reflected by the visible "shadow popping" effect. These rectangles are arranged in columns such that concurrent events do not overlap on the screen. The columns resize horizontally to fill the available screen real estate but will not shrink below the minimum button size discussed in section 2.4. The rectangles are arranged such that the most relevant events are located on the left side of the timeline. A horizontal red "now" line indicates the current moment in time and intersects all events that are ongoing.

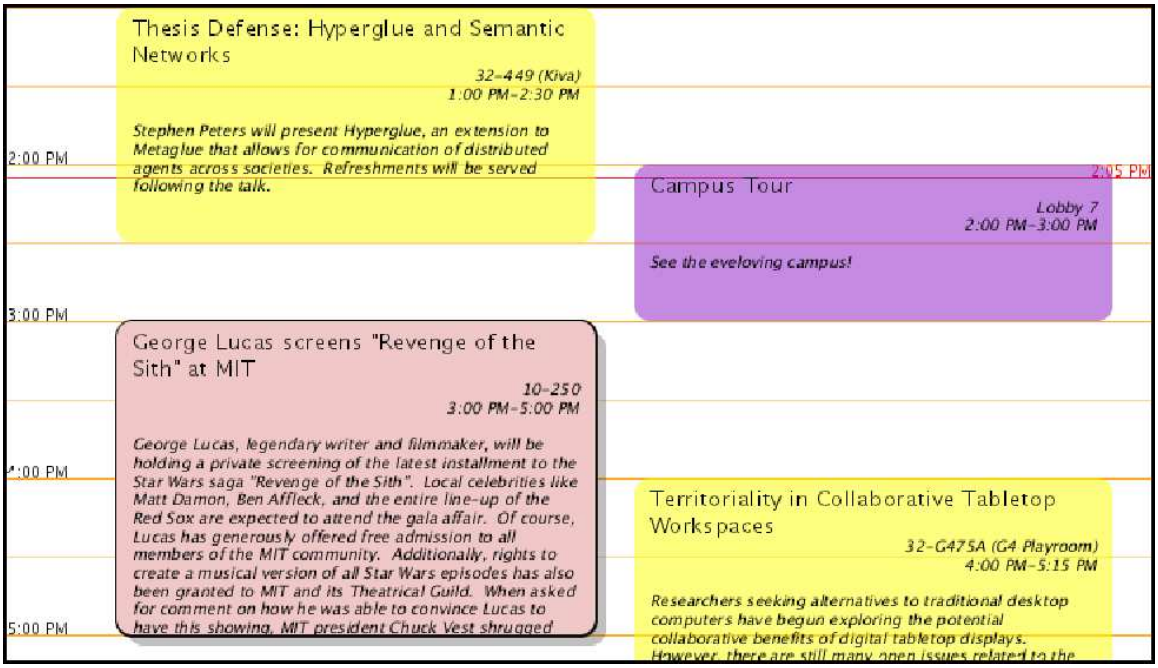

Figure 2-9: Timeline Dataview
The timeline dataview provides users with a day's worth of event summaries at a time. By clicking (touching) a summary in one of the rounded rectangles, they can summon a detailed view of the event. The red "now" line at 2:05 PM intersects all events that are ongoing.

Details Dataview

Most of SKINNI's screen real estate is devoted to providing summaries of relevant information. However, once users find information articles that are relevant, they will want to be able to access their details. The details view component presents all information available on a given item in a well-layed out and readable manner. Figure 2-10 shows a typical details view.

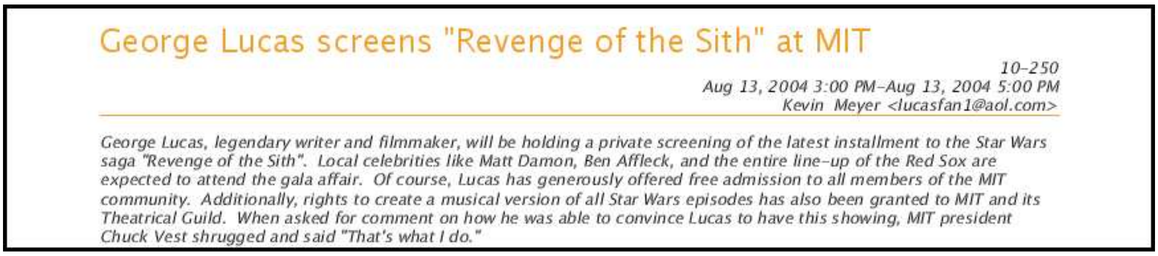

Figure 2-10: Details Dataview
Article details are presented in the form shown here. The title and important metadata are provided above the orange dividing line, while the full description is located below.

SKINNI List

A SKINNI list features two advantages over a traditional list view. First, the SKINNI list supports mixed types of data, so that different kinds of related information can be provided in a single list. Second, selected list items are allowed to have components embedded in them. This provides user's with immediate controls for navigating to related information. Figure 2-11 depicts a SKINNI list with heterogenous data types and an embedded control in the selected item.

SKINNI Table

Like SKINNI lists, SKINNI tables allow embedded components in each cell when the corresponding information item is selected. However, SKINNI tables do still require homogeneous data types represented row-by-row. Figure 2-12 shows a typical SKINNI table with an embedded control in the highlighted row.

Figure 2-11: SKINNI List Dataview
This excerpt of a SKINNI List shows three different types of information presented in the same dataview. The highlighted object is offering the user embedded controls to easily navigate through related information.

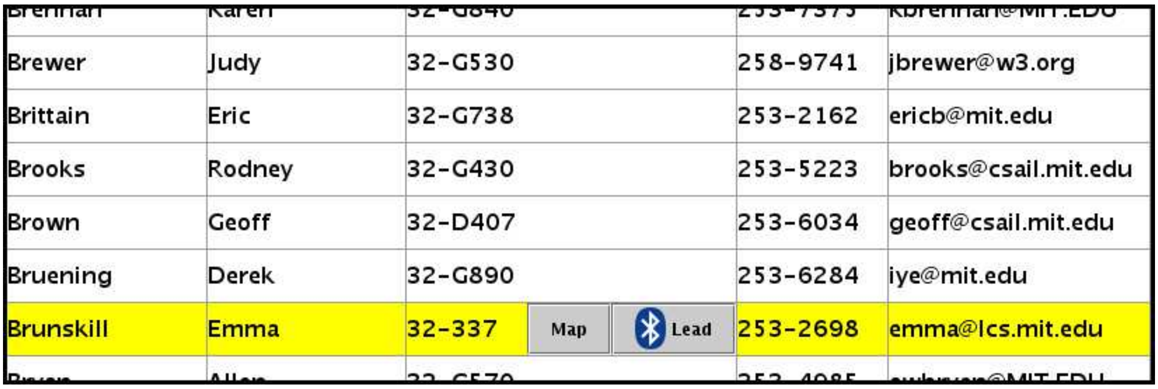

Figure 2-12: SKINNI Table Dataview
This SKINNI table is an excerpt from the "Directory" pane. Like SKINNI lists, the selected members of SKINNI tables offer embedded controls that provide convenient navigation to related information items.

2.6.6 Speech Interface

When a user queries SKINNI through speech, both audio and textual feedback are provided to the user. The region of the screen located below the pane area is devoted to the speech GUI. Figure 2-13 shows how the speech interface provides the user with feedback of the request that SKINNI heard and the response that it generated. This area also provides feedback pertaining to the readiness of the speech system. SKINNI speech can either indicate "Listening" (waiting for speech input), "Not Listening" (ignoring speech input), "Recording" (recognizing speech input), or "Thinking" (processing speech input). The microphone button is used to toggle the listening status of SKINNI.

Figure 2-13: SKINNI Speech GUI
This figure shows an active speech GUI that is responding to the user's request to find Howie Shrobe's telephone number. This GUI provides visual feedback to the user at the same time an audio response is generated. The green microphone button toggles the interfaces listening status (red when deactivated).

Typically, a user will approach a kiosk and find that the microphone button is red and the speech message says "Not Listening". If the user touches the microphone it will turn green and the message will change to "Listening". As the user utters a command like, "Please look up Howie Shrobe's phone number," the message changes to "Recording". After the user completes the verbal request, the speech message switches to "Thinking". After the command is processed, the speech GUI provides two messages: "I think you said: 'What is the phone number for Howard Shrobe?' " and "The telephone number for Howard Shrobe is 555-2341." The second message will also play over the kiosk's speakers as SKINNI automatically selects the "Directory" tab and highlights the entry for Howard Shrobe. [19]

2.6.7 Billboard

When SKINNI determines that a user's session has concluded, the graphical display transitions from the interactive GUI described above to a billboard-like display. This change serves two purposes. First, many elements of the SKINNI GUI are visible at all times, such as the tab bar and speech GUI. So, the billboard functions as a screensaver (preventing screen burn-in) by displaying a slideshow of images. Second, this display attempts to attract people to use the kiosk. It accomplishes this feat by displaying information in a large enough font that people walking by the kiosk can read.

When a user approaches the kiosk, they may choose to read the information as it is displayed, or they can interact with SKINNI. Touching the screen or pressing a key on the keyboard will dispose the billboard display and reveal the primary SKINNI GUI. At this point, SKINNI's session-level parameters will be reset so that the user can interact with a "fresh" instance of SKINNI.

Observed Design Flaw	Proposed Solution
Articles lacked meta-data such as the source of the information.	Meta-data should be included in the detailed view of articles, but remain absent from the summary views.
Users expected shortcuts from one piece of information to related pieces of information.	Hyperlink-like shortcuts could be provided for room numbers that led to maps and for people's names that led to directory entries.
When using the map, users were unclear where they currently were in relation to the entire building.	Provide a button labeled "Show this Kiosk's Location" so that users can learn where they are.
Controls allowing users to customize their session (filters, sorting mechanisms, etc.) lacked sufficient visibility.	Move the controls near the top of the interface so users are more likely to realize they are available.
The settings of customization controls within a tab were not preserved when returning to a tab from another.	Retain tab customization settings until the user either explicitly changes them or until the current user session ends.
Comments and other postings were not visible immediately after they were posted because they were placed at the bottom of lists.	Chronological lists should place newer items at the top. List items should be highlighted on the kiosk immediately after posting to provide feedback that the information item is now present in the kiosk network.
Users wanted to be able to send information items to specific people.	Add the ability to email an information item, attach a note, and specify a list of recipients.
Required form fields were left blank due to a lack of visibility.	Make fields (and their error messages) more visible. Also, where possible, provide sensible default values for required fields.
Users wanted the ability to locate a specific piece of information with a search string or other criteria.	Add a search field to tabs. Allow users to refine search results.

Table 2.1: Paper Prototype Flaws and Solutions

This table highlights the major design flaws observed while evaluating the paper prototype of SKINNI. A solution for each flaw to incorporate into further design iterations.

Observed Design Flaw	Proposed Solution
The distinction between many of the tabs was unclear to the users. Particularly, the "Today", "Events", and "Announcements" tabs were often confused, as were the "CSAIL Info" and "News" tabs.	Simplify the interface by reducing the number of tabs.
User expressed significant confusion over the distinction between an event and an announcement.	Combine events and announcements into a single type called articles.
Colors were too severe and not aesthetically pleasing.	Reduce the saturation of the colors. Choose colors that complement each other better.

Table 2.2: HTML Prototype Flaws and Solutions

This table highlights the major design flaws observed while evaluating the HTML prototype of SKINNI. A solution for each flaw to incorporate into further design iterations.

Observed Design Flaw	Proposed Solution
Finding a specific event in the timeline was difficult if its start time was not previously known. This effect was attributed to only being able to view the first eight hours of a day when using the date spinner.	Add logic to the timeline that automatically scrolls its view to the first event occurring in a given day. This allows more events to be seen when paging through days quickly.
Pressing the "Tab" key in a form did not advance focus to the next field.	Enable expected tabbing behavior within forms.
Users did not know what day of the week the timeline was displaying.	Add day of week to the date spinner.
Room number searching did not allow users to specify the building number or name.	Allow valid prefixes and suffixes when searching for map locations.
Users expected "Subject" field in email composing form. This field was hidden from the user and completed automatically by the prototype.	Make "Subject" field accessible to user.
Users desired additional modes of interaction. Specifically, users felt that they should be able to approach the kiosk and ask it questions the same way one might ask a concierge.	Add a speech interface that is robust enough to handle queries relevant to the application's domain.

Table 2.3: Java Prototype Flaws and Solutions

This table highlights the major design flaws observed while evaluating the Java prototype of SKINNI. A solution for each flaw to incorporate into further design iterations.

Map Lookup Actions	
	where is [room] [thirty two] <room_number>
	[can you] [please] [(show me | tell me)] [a map] [of] [where] [room] [thirty two] <room_number>
	[can you] [please] [(show me | tell me)] [a map] [of] [where] [is] <person> office [is]
	[Do you know] where [is] <person> office [is]
Directory Lookup Actions	
	<person>
	Who is <person>
	Do you know <person>
	[can you] [please] Show me [information] [for] <person>
	[can you] [please] Tell me [information] [for] about <person>
	[can you] [please] lookup <person> [in the directory]
	What is the phone number for <person>
	Do you know the phone number for <person>
	Do you know what <person> phone number is
	[can you] [please] [(lookup | tell me)] <person> phone number
	[can you] [please] [(lookup | tell me)] the phone number for <person>
	What is <person> office
	[can you] [please] lookup <person> office
	[can you] [please] lookup the office for <person>
	What is <person> e mail address
	What is the e mail address for <person>
	Do you know the e mail address for <person>
	Do you know what <person> e mail address is
	[can you] [please] lookup <person> e mail
	[can you] [please] lookup the e mail for <person>

Table 2.4: SKINNI Speech Grammar

This abbreviated speech grammar shows the phrases that SKINNI is able to recognize and act upon. Words enclosed in "[ ]" are optional. Words enclosed in "<>" represent key variables like a person's name or an office number. "(X | Y)" indicates that either "X" or "Y" can be spoken.

Chapter 3

Implementation

When implementing SKINNI, we had to integrate many OK-net systems and build utilities for information handling. Overall, we attempted to maintain our design goals and construct a system that was powerful yet natural and efficient to use. While it is impossible to capture all of the intricacies of the implementation in this type of document, this chapter touches on the most significant aspects necessary to understand prior to examining the large codebase behind SKINNI.

3.1 Integrated OK-net Components

SKINNI integrates smart kiosk systems that allow for context-based information access. Specifically, SKINNI uses the k:info knowledge maintenance system to gather data and reason about the context of a user at the kiosk. This knowledge and its inferences are stored in a semantic network. SKINNI also relies on networking and communication techniques developed by other researchers in the OK-net project. An additional component that SKINNI integrates to provide an effective user interface is a speech recognition engine developed at CSAIL.

3.1.1 Metaglue and Hyperglue

Metaglue is an extension to the Java programming language that serves as a communication infrastructure for distributed agents. The AIRE research group uses these Metaglue agents as "digital shadows" for individual ubiquitous computing components present in an intelligent environment. For example, an intelligent environment would have an agent that managed each device or appliance in that space. Additionally, an agent would overlook every abstract resource in the space, such as a network resource, printing resource, or information-gathering resource. One component that might be overlooked as a necessary part of ubiquitous computing in an intelligent environment is the user that occupies that space. For a person, agents track preferences and other personal characteristics. For example, an agent would be responsible for tracking a user's preferred multimedia configuration or monitoring the user's email accounts for new messages. [11]

The term *society* is used to denote a group of agents that represent the same entity, whether that entity is an intelligent environment, an individual person, a research group, or even a hierarchical collection of societies. While Metaglue is designed to provide robust and persistent communication between agents, it only does so for agents in the same society. Hyperglue was designed as an extension to Metaglue that provides cross-society communication. Hyperglue embeds an additional agent in each society called its *ambassador*. All incoming and outgoing communication between agents in separate societies are coordinated by their ambassadors. The ambassador in each society is responsible for locating requested agents, authorizing the communications request, and establishing communication channels. Once a communication channel has been established, the ambassadors play no further role in the agent-to-agent communication. [8, 10]

Each kiosk in the kiosk network is a separate intelligent environment and therefore has its own society and set of Metaglue agents. The SKINNI application itself is an agent that is primarily responsible for harvesting knowledge and inferring user context. The SKINNI agent communicates either with other agents that directly perform tasks useful to SKINNI or with agents that interface to useful applications.

3.1.2 Communication and Networking

Kiosks are effective information portals because they provide users with flexible means of interacting with contextually relevant data. Beyond the task of simply accessing information, kiosks allow users to exchange information. Specifically, kiosks can communicate with personal mobile devices, other kiosks, and users.

Kiosk Communication with Personal Mobile Devices

A kiosk may form an ad hoc network with a user's personal mobile device. This network would allow the mobile device to download or upload information, control SKINNI, or authenticate the user. A wireless protocol such as WiFi, Bluetooth®, or IR would likely enable this communication, but any protocol common to both the kiosk and mobile device could be used.

Kiosk Communication with other Kiosks

Logically related kiosks that share information are said to be part of the same kiosk network. This means that information posted at any kiosk could be propagated to any other kiosk on its network. However, the way in which the information is utilized at each kiosk will differ based on the configuration of the kiosk and its applications. Specifically, SKINNI provides mechanisms to filter and rank information on an installation-by-installation basis.

Kiosk Communication with Users

Kiosks present a multimodal interface through which they convey information to users. However, this type of direct interaction is not the only kiosk-to-user communication that the kiosk network can provide. Rather, the kiosk network allows users to produce asynchronous and just-in-time (JIT) messages delivered to people who are not necessarily actively using a kiosk. These message delivery services allow users to request notifications at a high level. Specifically, a user can request that some information be sent to a given set of people, at a certain time, or through some media. In the presence of missing or incomplete message specifications, the kiosk network pools all available information about the recipient's current context and delivers the message in the most appropriate and effective manner possible.

3.1.3 k:info—Kiosk Knowledge Maintenance

One of the first applications developed for the kiosk platform was the k:info billboard[1], which collects, selects, and displays context-aware information [17]. The application collects data from online sources (news, weather, email, etc.), perceptual sources (video and audio input), and system status (current time, last interaction, configuration, etc.). This data is then processed by a knowledge-based selector arranged as a blackboard architecture [12]. This blackboard is monitored by Metaglue agents with context-specific expertise that make recommendations about the knowledge. The k:info billboard then displays the most highly recommended piece of information on the kiosk's screen. The mechanisms for automatic knowledge collection and selection used in k:info are also used by SKINNI, but SKINNI manages its own GUI to display relevant information items.

3.1.4 Knowledge Representation with a Semantic Network

The underlying representation for all knowledge retained by OK-net kiosks is a semantic network implemented through a database. A semantic network is a data structure comprised of knowledge nodes and semantic links between the nodes. Combining rich semantic information with a network structure yields benefits over other knowledge representations, including localized operations for updating knowledge and efficient inference reasoning. [9, 10]

To better understand semantic networks and how SKINNI uses them, consider the sample network shown in figure 3-1. This network shows two people, Howie

[1] NOTE: This is the same billboard described in section 2.6.7 that functions as a screensaver and long-range attractor for SKINNI.

Figure 3-1: Sample Semantic Network
This sample semantic network represents some knowledge about Howie and Steve. SKINNI uses networks like this to infer as much context as possible about users.

and Steve, who are both members of the IRoom Project. Howie is responsible for the IRoom Project and therefore supervises Steve. Both are interested in HCI, but Howie is an expert in LISP while Steve is an expert in Metaglue. According to this sample network, Steve is currently in his office, 832, which is part of some other collection of rooms. This is the only knowledge represented in this section of the semantic network. However, this network can be used to infer the state and extended relationships of entities represented in the network. These inferences can be used to support high-level requests such as "Send this message to members of the IRoom project" and "Can Steve introduce me to any LISP experts?" [9]

3.1.5 Galaxy Speech Recognition System

SKINNI uses the Spoken Language Systems group's (SLS) speech recognition engine called Galaxy. Their system has been used in many successful projects including

Jupiter [24], Orion [14], Mercury [15], and others [16]. Galaxy offers some features that make it advantageous to other available speech recognition systems:

1. Galaxy is user independent. This means that users do not have to train the system prior to using it.

2. Galaxy is robust in the presence of mis-articulated phrases. This means that utterances like, "Can you show me the...uh...phone number for Victor Zue?" will still be correctly recognized.

3. Users are not required to add unnatural pauses between words. Some voice recognition systems require split-second silences between words in order to correctly parse an utterance, but Galaxy operates on an entire discourse.

4. Galaxy supports multilingual recognition. While SKINNI does not currently support languages other than English, we plan to support other languages in the future.

5. Galaxy speech domains (grammars) are easy to construct and modify using SpeechBuilder [22]. This means that the grammars can be extended to accommodate new utterances and actions.

SLS is currently conducting experiments that focus on improving speech recognition rates at public space kiosks. Two properties of public space kiosk influence speech recognition adversely compared to Galaxy's initial design. First, kiosks use a different type of microphone (linear array) that may transform audio input in ways that Galaxy is not robust to. Second, the acoustical properties of public spaces often differ from places where voice recognition has typically been used like offices. Specifically, Galaxy needs to adjust for both echoes and ambient noise with higher than normal volume.

3.2 Utilities

We developed some utilities that allow SKINNI to provide users with the most relevant information available. First, we provide an automated system that scores information articles along a wide range of dimensions and then ranks the articles according to their relevance. Second, we provide a search function that carefully examines information and its structure to determine if it matches some criteria. We are also experimenting with another set of utilities that involve identifying and authenticating kiosk users.

3.2.1 Scores and Ranking

The k:info knowledge maintenance system provides SKINNI with a collection of contextually relevant information articles. However, these articles are not sorted according to relevance, this task is left up to SKINNI.

To allow for the greatest flexibility in terms of a ranking mechanism, we constructed a voting system where different agents would score each article according to some criteria. Scores provided by these modules range from -1 to 1 with more positive values indicating more relevant results. For example, an "Upcoming Event" score module may provide a positive score for information pertaining to an event happening later today, a negative score for an event that has already passed, and a 0 (neutral) score for information with no set time duration.

Score modules are registered with rankers that consider the scores of all their modules to determine the final relevance-based ranking. Rankers can introduce another level of complexity, since there is no specified way in which they are required to handle the votes registered by each score module. For example, a "Simple" ranker may just add all of the votes to obtain the final score and rank. A "Weighted" ranker may assign different levels of importance to each score module and its votes. Yet another ranker, the "Veto" ranker allows some information articles to be completely removed from the collection of relevant articles if a given score module awards it a score of -1.

3.2.2 Search

The SKINNI search function examines both the content and structure of nodes within the semantic network to determine if an information item satisfies the search criteria provided by the user. First, SKINNI decomposes the user's search string into tokens (unique words). Next, the search function traverses all nodes in the semantic network and compiles the set of nodes that match the first token. This process is repeated for all tokens in the search string. Lastly, SKINNI returns the intersection of these token-matching node sets in the form of a SKINNI list like the one pictured in figure 3-2.

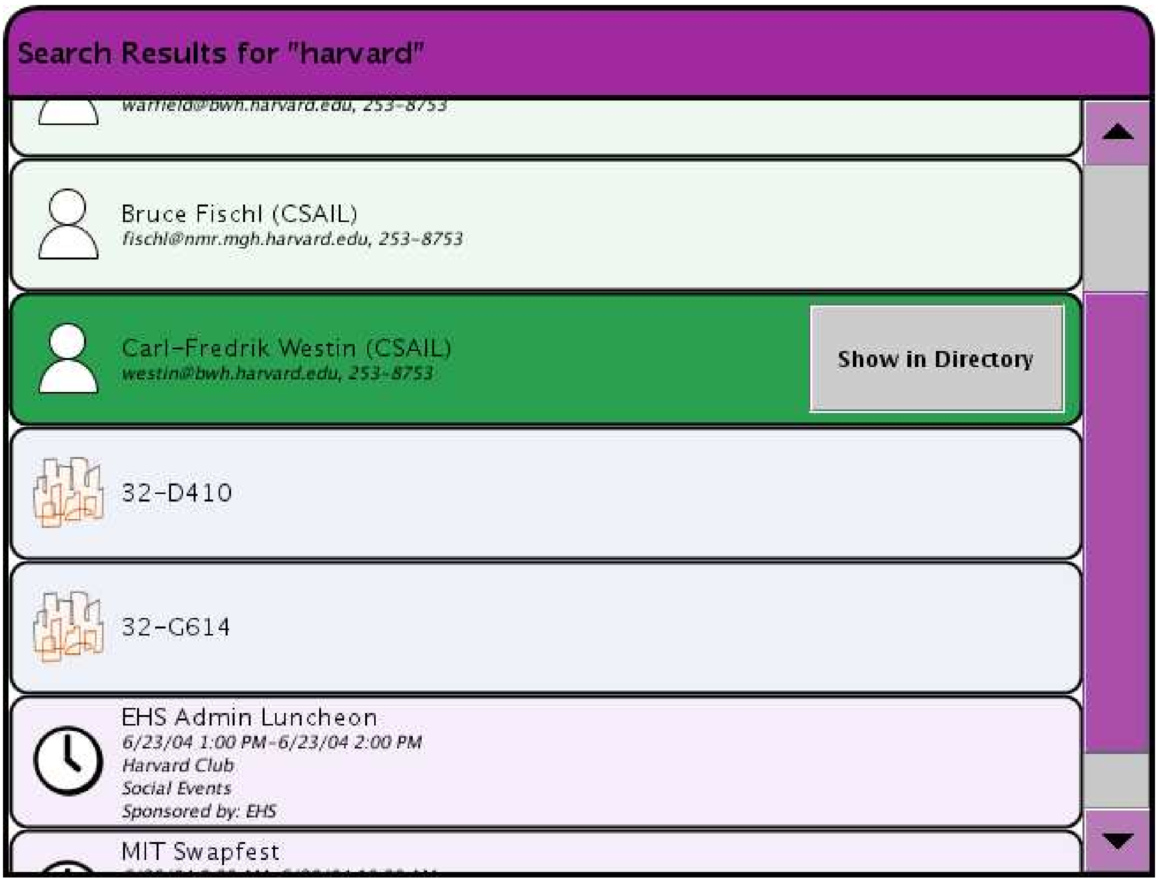

Figure 3-2: Sample Search Results

This SKINNI table of search results for the search string "harvard" is an excerpt from figure C-5. All of these results have a relation to Harvard: The people have "harvard.edu" email addresses, the locations are these member's audiences, and the events mention harvard in their descriptions.

The search function uses a level-based approach to determine if a node in the semantic network matches a given search token. This method examines specified fields of that node and its sub-nodes for token occurrences. Table 3.1 lists the data types, fields, and levels that the current deployment of SKINNI uses to match tokens within nodes. When beginning a token search on a node, the search function examines all Level 1 fields. A field that is a `String`[2] object satisfies the search if the token is a case-independent substring of the field's value. For a field that points to another node (a sub-node), the search is satisfied if the token matches the Level 2 fields of the sub-node. This pattern continues for all remaining levels of search. Only the node for which the Level 1 search was initiated is appended to the set of token-matching nodes. Nodes that do not have a Level 1 search capability cannot be returned as a search result.

The following scenario details how SKINNI uses search levels to process a sample search string, "Marvin Minsky". SKINNI considers each search token separately, so the search function begins by examining all Level 1 fields of `Article` nodes in the semantic network for the "Marvin" token. This step will immediately append to the "Marvin"-matching node set all `Article`s in which "Marvin" is a case-independent substring of the title, description, or location. If none of these `String` fields match the token, then the poster and type fields are examined with a Level 2 search. For the type node, if the name field contains the substring "Marvin" then the `Article` (not the `ArticleType` object) matches the token. Similarly, for the poster node, if the email field contains the substring "Marvin" then the `Article` (not the `PersonInfo` object) matches the token. If the email field does not match, SKINNI will consider poster's two remaining Level 2 fields: office and name. Since both of these fields are sub-nodes their fields will be examined through a Level 3 search. The office field is a `CSAILLocation` object that does not support Level 3 search and will therefore be automatically disqualified. However, the name field has three Level 3 search fields: first, middle, and last. If any of these `String`s contain "Marvin" as a substring, then the `Article` (not the `Name` object or `PersonInfo` object) matches the token. Since,

[2]Since Java is the implementation language of SKINNI, `String` is used as a shorthand for `java.lang.String` in this document.

this was an exhaustive examination of fields in the `Article` and its sub-nodes, if the `Article` has not matched the token yet it will never be appended to the "Marvin"-matching node set and will therefore not be returned in the overall results. Now that SKINNI has finished searching all `Article`s in the semantic network for the "Marvin" token, SKINNI examines the remaining data types. Specifically, the search function will only consider `CSAILLocation`s and `PersonInfo`s because they are the only other data types with Level 1 search fields. Since `CSAILLocation` objects have only one searchable field, which is a `String`, they will only match if "Marvin" is a substring of the name field. With the `PersonInfo` nodes, following the same procedure described for `Articles`, there will be a match if "Marvin" is a substring of the email, affiliation, homepage, telephone, or title field of the `PersonInfo` node, or the name field of the office sub-node, or the first, middle, or last field of name sub-node. This entire process is then repeated for all data types with the "Minsky" token from the original search string. The intersection of the "Marvin"-matching nodes and the "Minsky"-matching nodes is returned as the overall search result.

3.2.3 Identification and Authentication

We are currently exploring many ways of identifying and authenticating kiosk users to SKINNI: username/password log-on, Bluetooth® device detection, vision-based approaches, and user stroke recognition. These methods have varying degrees of reliability, security, and natural interaction. While no identification or authentication methods are currently deployed in SKINNI, it is likely that most of these options will be available concurrently so that each module could vote on the identity of the current user. By poling these voters, SKINNI can assert an overall confidence level regarding the user's identity.

Username with Password

Identifying and authenticating people through username and password (or PIN) log-ons is a traditional approach that is widely used for desktop and web applications.

Class	Field	Type	Level 1	Level 2	Level 3
`Article`	poster	`PersonInfo`	X		
	title	`String`	X		
	description	`String`	X		
	type	`ArticleType`	X		
	location	`String`	X		
`PersonInfo`	name	`Name`	X	X	
	email	`String`	X	X	
	affiliation	`String`	X		
	office	`CSAILLocation`	X	X	
	telephone	`String`	X		
	homepage	`String`	X		
	title	`String`	X		
`Name`	first	`String`		X	X
	middle	`String`		X	X
	last	`String`		X	X
`CSAILLocation`	name	`String`	X	X	
`ArticleType`	name	`String`		X	

Table 3.1: SKINNI Search Specifications

This table shows the search levels SKINNI uses when processing a token from a search string. A node matches the search string when all of its tokens are present either in the node's Level 1 `String` type fields or in the sub-nodes' incrementally deeper level fields.

While this method is reliable and secure, we feel that our multimodal kiosk platform should provide more natural alternatives. However, if SKINNI is used for (as yet unspecified) critical functions, the security of username and password log-on should outweigh the natural interaction benefits of other methods.

Bluetooth® Device Detection

Kiosks deployed in CSAIL are equipped with Bluetooth® dongles that constantly scan for nearby devices (Bluetooth®-enabled cell phones, PDAs, laptops with dongles, etc.). If a user carries one or more of these devices, the kiosk could detect them and assert the identity of the user. Unfortunately, this approach requires users to carry Bluetooth® devices, thereby eliminating their ability to log-on to SKINNI if they are not carrying any. Additionally, this approach could cause SKINNI to incorrectly assert the user's identity if one or more Bluetooth® devices not belonging to the current user are in the vicinity of the kiosk. However, we will continue to investigate this method since it can increase SKINNI's user identity confidence level when used in conjunction with other methods.

Vision-based Approaches

A vision-equipped kiosk can provide many ways to identify a user. For example, the kiosk could employ face recognition techniques to compare the user's facial features to those in a database of known users. Alternatively, color histograms could provide a short-lived way for enabling kiosks to recognize that a user is the same person who was recently using another kiosk. Vision can also be used to detect skin and ascertain the presence of a user, even if that user's identity is not known. Since vision is the primary method by which people identify each other, these methods clearly provide natural interaction. However, none of these methods are yet reliable enough for deployment. Additionally, CSAIL members have both directly and indirectly expressed privacy concerns regarding the placement of cameras on public-space kiosks. [18]

Passdoodles

The newest technology we have been experimenting with as an authentication method is Passdoodles [21]. A Passdoodle is a unique gesture a user scribbles on the kiosk's touch screen; typically a variation on the user's signature. By considering both the spatial and temporal make-up of the gesture, the kiosk can assert who is touching the screen. This method requires users to train the kiosk to recognize their Passdoodles, but after this initial training period, interaction is just as natural as using the touch screen. This method is also fairly secure since impostors who know what a user's Passdoodle looks like cannot easily reproduce the scribble with the same temporal signature.

Chapter 4

Conclusions

SKINNI provides an interface for smart digital kiosks that allows users to access, annotate, and share contextually relevant information. This application is one of the reasons why kiosks provide an advantageous platform for information retrieval compared to other interfaces like signs and personal mobile devices. While SKINNI will not replace these other information interfaces, it better facilitates tasks of enterprise integration, informal collaboration, and public-needs servicing.

It is difficult to quantitatively measure the success of an application that is designed to provide users with exactly the information they need given their current context in a transitional space. However, we gathered some usage statistics that seem to indicate that SKINNI was repeatedly used within CSAIL to gather information specific to the lab. For example, figure 4-1 shows a large number of daily interaction sessions during a one month period. An interaction session encompasses all of the usage recorded for a single person at a kiosk at one time; these numbers do not represent the number of unique users. Figure 4-2 provides a breakdown of what panes were requested during these interaction sessions. Clearly, most users were seeking directory information. Figures 4-3 and 4-4 provide the same data as the previous two figures, but are accumulated on an hourly basis. The strong presence of usage during working hours could suggest that the information retrieved from SKINNI led to enterprise integration or some type of collaboration. Further tests, observations, and surveys will reveal users' application of the knowledge SKINNI information.

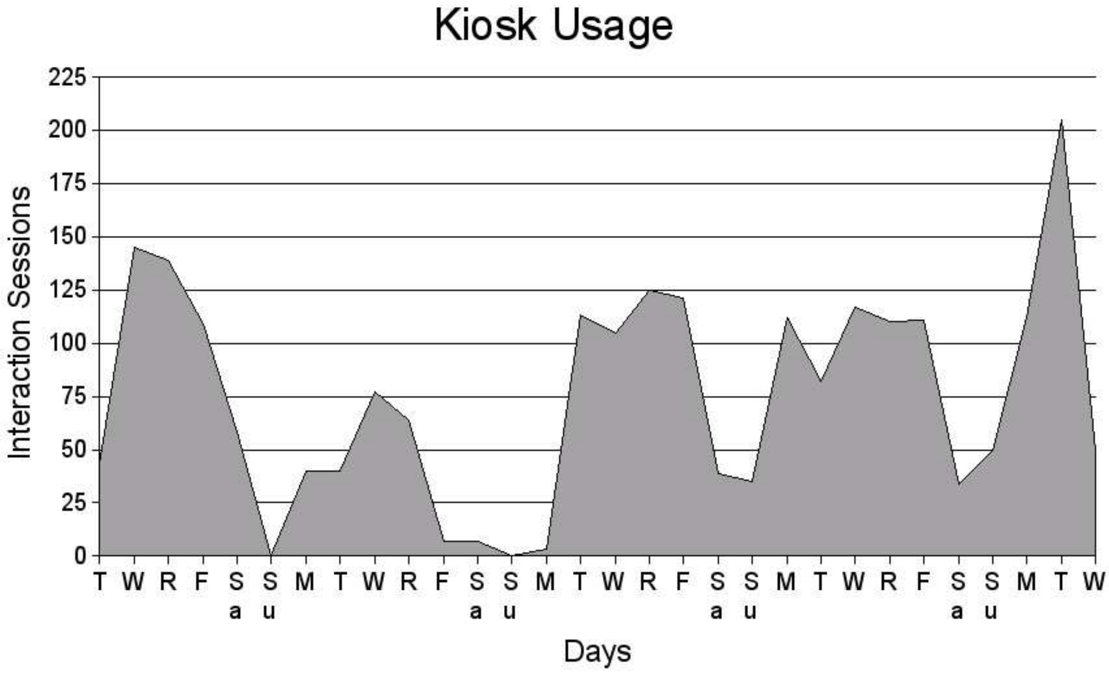

Figure 4-1: Daily Kiosk Usage Statistics
This plot shows the number of user sessions initiated every day during a month. Usage is encouraging and follows a weekly cyclical pattern with lulls on the weekend.

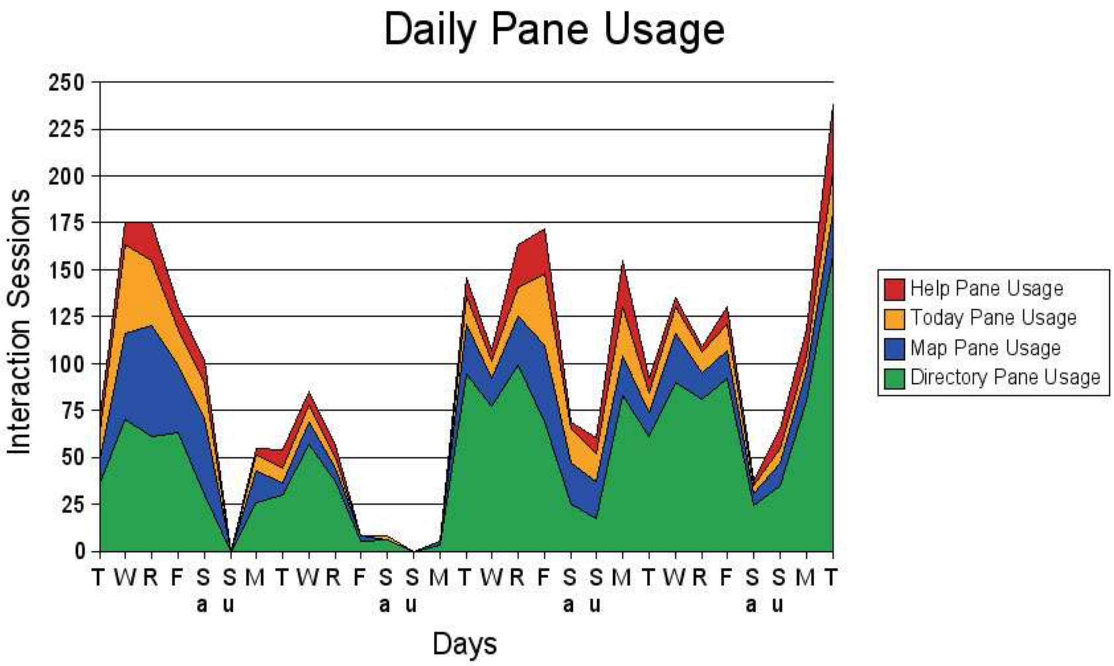

Figure 4-2: Daily Pane Usage Statistics
This figure shows the number of times daily that each pane was selected by a user. The "Directory" pane was the most popular.

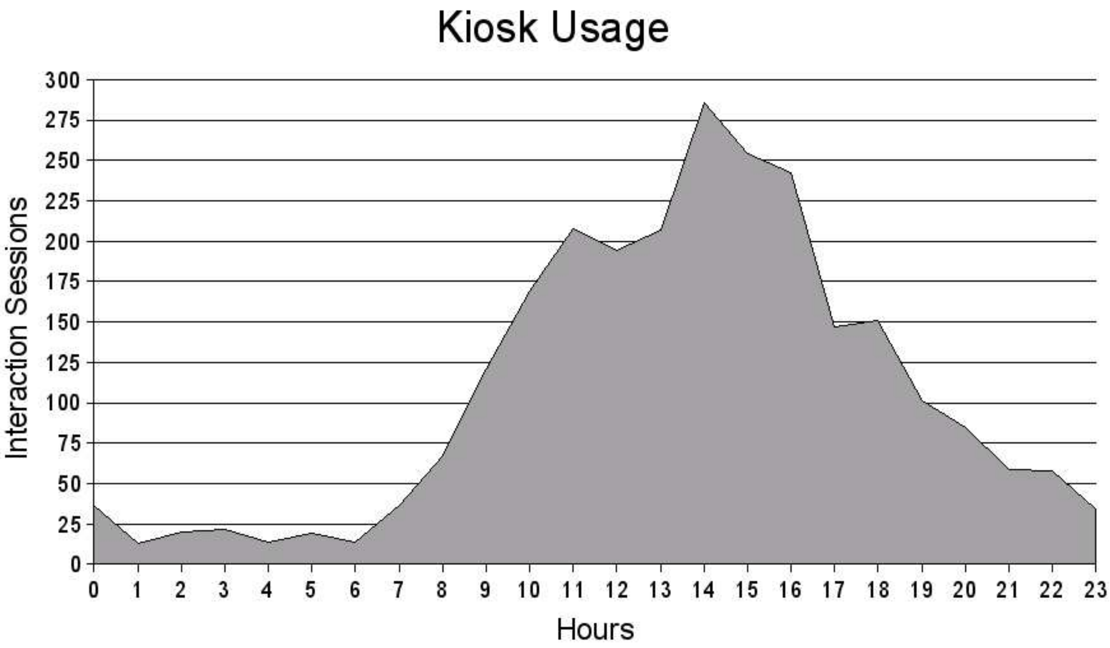

Figure 4-3: Hourly Kiosk Usage Statistics

The accumulated hourly quantity of interaction sessions over the month are presented here. Heavy use during the workday could imply that information was used for interorganizational tasks like enterprise integration or collaboration.

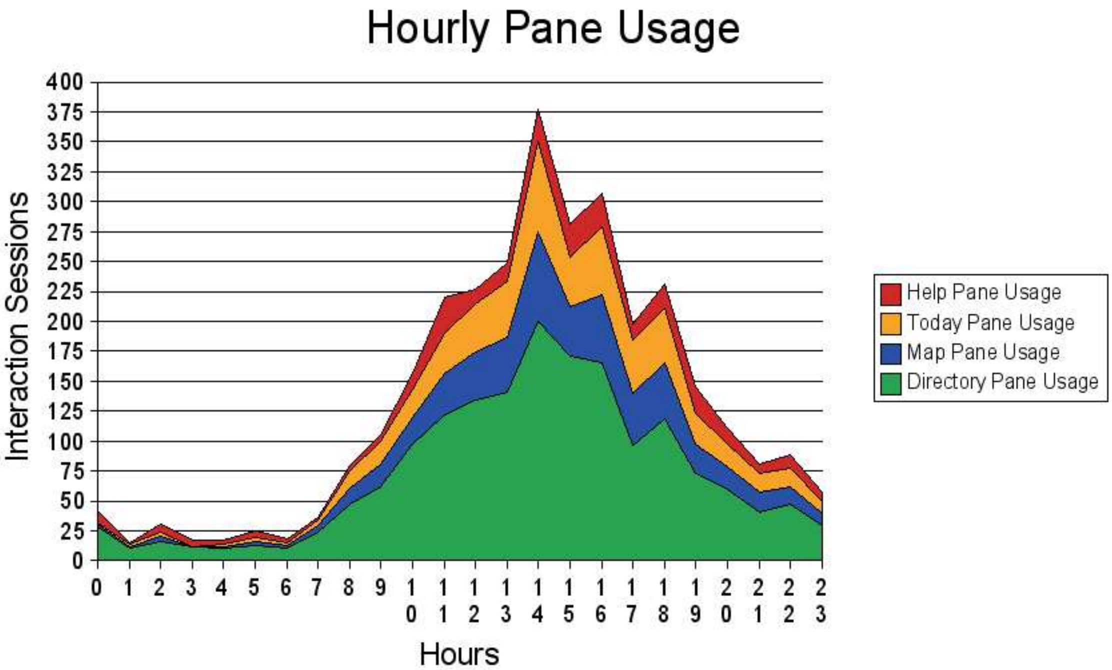

Figure 4-4: Hourly Pane Usage Statistics

This figure shows the accumulated hourly breakdown of pane selection by lab members and visitors.

Appendix A

OK-net Kiosk Hardware

The type, arrangement, specifications, and other parameters of all the hardware comprising the OK-net kiosks were carefully chosen to provide an interface which users could interact with naturally and efficiently. Other design considerations address issues of cost (use of "off the shelf" technology), courtesy (not disturbing non-users sharing the same public space), and flexibility (upgrading components as technology improves). Three kiosk form-factors have been designed and are pictured in figure A-1, figure A-2, and figure A-3. This appendix details the hardware that makes up these platforms designs, the rationale behind the associated design decisions, and the requirements SKINNI places on various groups of hardware.

Pointing In general, pointer input events can be generated by many different devices such as standard desktop mice, trackballs, touchpads, styli, touch screens, etc. For kiosks with medium-sized screens (standard desktop monitors) a touch screen interface is used because it provides users with the most direct method for interacting with the displayed information.

Kiosks with larger displays (many feet in extent) require different means for generating pointer events. Currently, two methods are under investigation. The first approach involves viewing a user's gestures and translating them to mouse events [13]. Specifically, an outstretched arm moving from one region of space to another indicates that the mouse cursor should follow a similar path on the screen. The user generates a mouse click by rotating the outstretched arm 90 degrees. This method requires that the kiosk be equipped with a camera capable of providing a high enough resolution and frame rate to accurately distinguish users' gestures. The second approach to generating mouse events has users "capture" the kiosk's cursor with their own personal mobile devices. Users can then control the cursor by using the default pointing interface available on their devices. This method requires that the kiosk be equipped with various wireless networking hardware and that the user has a device capable of communicating with the kiosk.

Text Entry Some sort of text entry device is required by SKINNI (and potentially other kiosk applications) so that users can conduct searches and annotate information. Keyboards are currently used since they are the most familiar device available. However, as with mouse input, OK-net developers are investigating two other options for text entry at kiosks. The first option is to use an on-screen keyboard. However, this method sacrifices screen real estate, provides no haptic feedback, and forces the user into inefficient or uncomfortable hand positions. A much more encouraging alternative is speech recognition. Clearly, the effectiveness of this approach depends on an adequate audio input system and matching speech recognition software.

Camera A video camera at a kiosk can serve multiple functions including video communication, snapshot acquisition, user identification, and gesture recognition. The usefulness of any of these features depends greatly on the application which is running on the kiosk. Due to expressed privacy concerns, SKINNI does not currently require a camera to function. However, SKINNI kiosks with cameras provide additional modalities of interaction to the user, such as the gesture-based input described above.

Speakers Kiosks are located in public spaces, which means that audio output from a kiosk could disturb people in the surrounding area. A trivial solution to this problem is to decrease the volume of the speakers. Unfortunately, users then might not be able to hear the kiosk's audio feedback due to the ambient noise of the space it is situated in. Though fortunately, for most public spaces, there is an output volume level for which users can still hear the kiosk without the audio output disturbing others nearby. Another solution is to use a directed speaker like the one shown in figure A-2. In this case, a plastic shield directs sound downward so that people standing at the kiosk can hear it clearly while bystanders are only exposed to an attenuated version of the sound.

Microphone One of the most natural modes of human communication is speech, therefore kiosks should be able to listen to their users. There are three forms of microphones that kiosks could use: standard microphones, headsets, or array microphones. A standard microphone mounted on the side of a kiosk lacks the feel of natural interaction since the user must lean in towards the microphone to be heard over any background noise. A headset moves the microphone close to the user's mouth to decrease the noise-to-signal ratio but does not provide a natural feel due to the equipment that must be worn to use them. Array microphones offer a better option for providing users with a natural type of speech interaction. Active noise cancellation and signal amplification across microphones in the array allow users to stand and speak at the kiosk as they might when conversing with another person.

Enclosure There are two reasons to enclose the hardware of a kiosk. The first reason is to protect its components from tampering, theft, or damage. Second, kiosks should be noticeable while still conforming to their environment. Figures A-1, A-2, and A-3 show the three current enclosure designs for smart kiosks. The first figure shows the components of the kiosk contained within a plywood box. This design matches part of the building's architecture and also well protects the kiosk's hardware. In the other two designs, protection is provided by storing most of the equipment controlling the kiosk in a locked server room. These kiosks also blend into their environment by employing a metal frame or embedding flush in the wall, respectively.

Avatar An avatar acts as the personification of a kiosk. If present, the kiosk's avatar may manifest virtually (on-screen assistant) or physically (external robot). Figure A-3 depicts the physical embodiment of a kiosk avatar known as IGOR [13]. This robotic mask is equipped with a video camera in one eye and a microphone in the other. When a user talks to IGOR, visual tracking software moves the robotic mask so it is always facing the user. This tracking improves the reception of the microphone and provides feedback that the kiosk is paying attention to the user. This arrangement allows users to interact with IGOR and the kiosk naturally, almost as if it were a person.

Display The size, quantity, and position of displays for kiosks can vary from one installation to the next. A single display about the same size as a standard desktop monitor, provides a feel of intimate interaction with the kiosk and the data it presents. Multiple screens and large screens facilitate use by multiple users and may serve as an impetus for informal collaboration. In current designs, screens are placed in a plane perpendicular to the ground at an elevation that provides comfortable viewing for people of average height. Options are currently being explored to allow users to adjust the elevation and inclination of the screen to better meet their individual needs.

Network One goal of the OK-net project is to provide a wide range of communication options through the kiosk. Specifically, kiosks should be able to communicate with other kiosks in the network as well as with the personal mobile devices that people might carry with them. Kiosks communicate with other kiosks through a LAN allowing postings or annotations made at one kiosk to propagate to other kiosks. This LAN also enables email and similar messages to travel through the standard network infrastructure to personal devices. Direct communication with personal mobile devices depends on the capabilities of the device that a user carries, but three common wireless modes are identified as most useful: WiFi, Bluetooth®, and IR. Kiosks supporting these protocols would be able to communicate directly with many mobile devices.

Computer Every kiosk needs at least one computer to run applications and coordinate the input and output of all the aforementioned peripherals. Depending on the application, each kiosk's computer or computers will need to meet some criteria for processor speed, memory availability, local storage, graphics acceleration, etc. Additionally, for kiosk form factors like that shown in figure A-1 with the plywood box enclosure, the computer must be physically small enough to fit in the enclosure with all of the other components of the kiosk. The choice of operating system running under the kiosk's applications is not specified. SKINNI is Java-based and will run under all operating systems supporting Java version 1.4.2 and later.

Figure A-1: Deployment Kiosk Design

This kiosk design is currently deployed in elevator lobbies at CSAIL. It features a touch screen display, mounted keyboard, and plywood enclosure. The computer operating this kiosk is a SlimPRO Cappuccino PC (http://www.cappuccinopc.com) equipped with a WiFi card and Bluetooth® dongle. A linear array microphone (not pictured) allows users to interact with SKINNI through speech.

Figure A-2: Prototype Kiosk Design
This kiosk design is currently situated near the AIRE research area. It is used to test new kiosk features before they are deployed on the plywood-enclosure kiosks. This design features a dual display, sound deflection shield, small video camera, and linear array microphone.

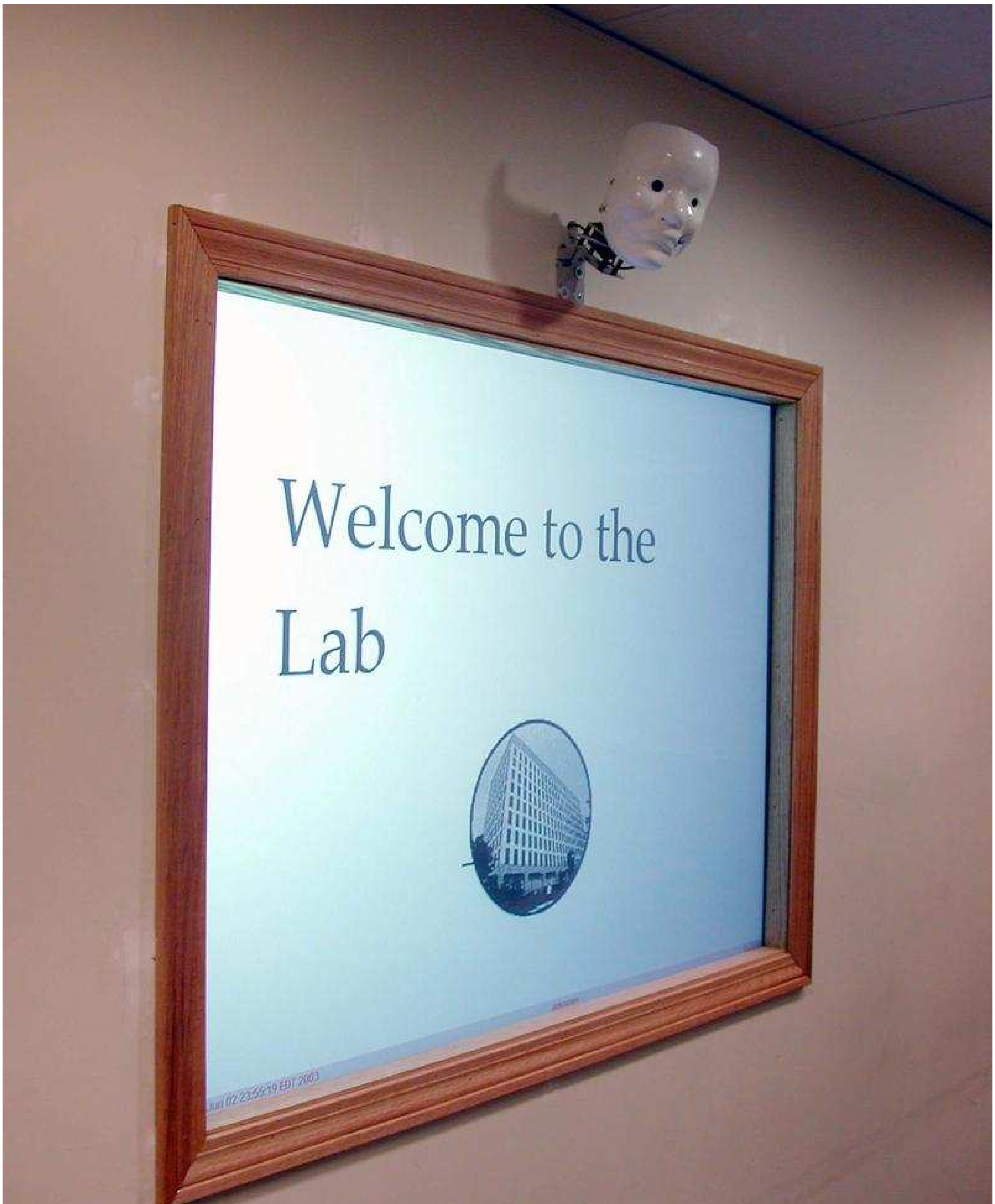

Figure A-3: Experimental Kiosk Design
This kiosk features a large display (3 ft. × 4 ft.) and the IGOR kiosk avatar. The primary purpose of this kiosk is to experiment with natural large display interaction. This installment is not currently deployed.

Appendix B

Touch Screen User Study

User interfaces presented on touch screens must provide graphical components that both display information effectively and enable effortless interaction. Therefore, we conducted a user study to explore the characteristics of interactive widgets that afford excellent usability when presented on a touch screen.

B.1 Setup

We began this user study by formulating hypotheses pertaining to users' typical interactions with widgets presented on touch screen kiosks. These hypotheses consider the effects of an on-screen target's size, shape, and location, as well as a typical user's physical abilities.

Size There is some minimum size for which users can touch an on-screen target during nearly all attempts.

Shape Users will be able to hit targets of different dimensions and densities with varying levels of success.

Position User accuracy will decrease proportionately to the target's distance from the center of the screen.

Fatigue As people use the touch screen for long periods of time, their accuracy will decrease; their target hits will drift downward.

Time Users are more accurate hitting targets the longer they take to do so.

We constructed an interactive computer program to test our hypotheses. The test program ran in full-screen mode and presented a single red target on a white background. Once a user touched the target, it would disappear and a new target with a randomly chosen shape and size would appear elsewhere on the screen. Each target was either a rectangle, ellipse, cross, or X-shape similar to those pictured in figure B-1. Each target ranged along vertical and horizontal dimensions from 10–100 pixels (0.10–1.04 inches on a 96 d.p.i. screen).

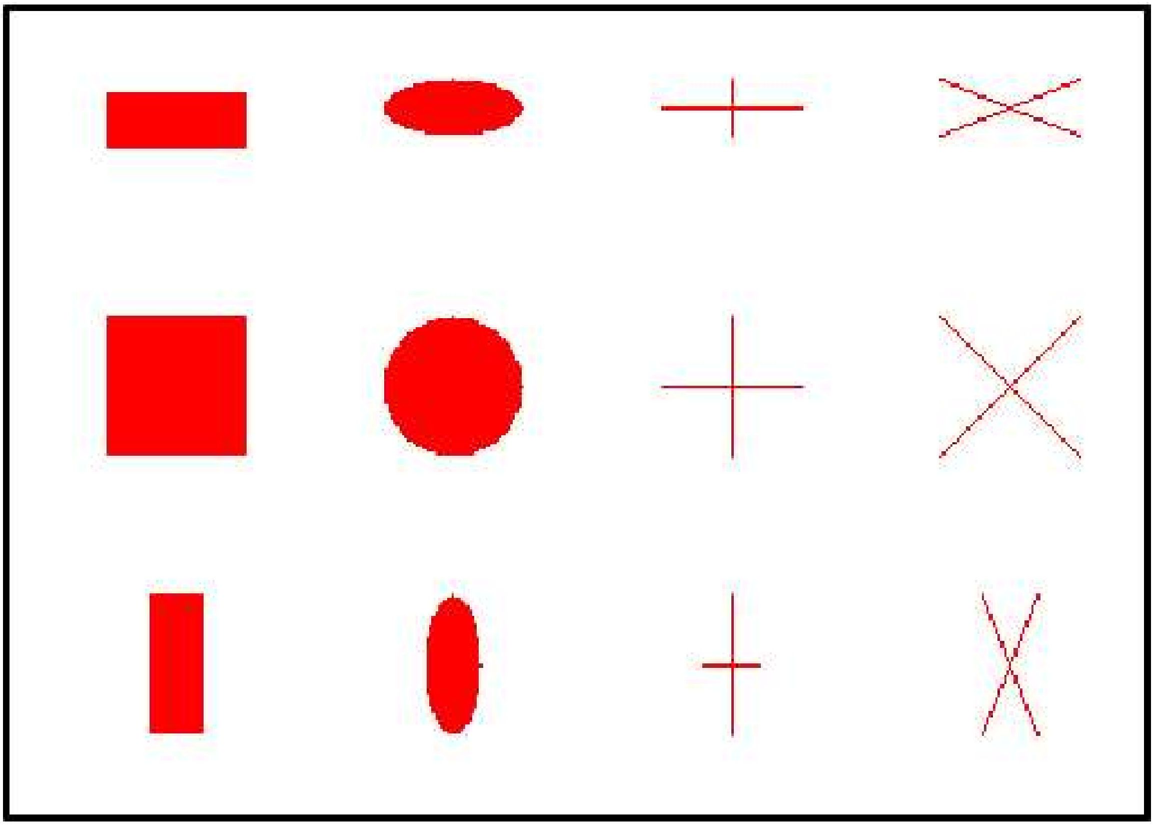

Figure B-1: Sample Targets
This figure provides twelve sample targets from the testing program; three each from the four shape classes. The targets pictured are actual size.

Participants interacted with the test program running on a typical touch screen kiosk for a period of three minutes. They were asked to touch each target as close to its center as they were able. While users were informed that they could take as long as they like in between hitting targets, we encouraged them to proceed at a pace that was "quick yet comfortable, not rushed". Users were also informed that they could terminate their participation at any time if they became fatigued or otherwise uncomfortable with continuing.

B.2 Results

Nineteen randomly selected CSAIL members participated in this user study. Collectively, they hit close to 3,000 targets. The data collected from these trials are presented below and interpreted within the context of each of the proposed hypotheses.

Size Hypothesis Results

We hypothesized that there was some minimum size that a target could be made such that users could touch the targets during nearly all attempts. This means that the absolute distance between the target hit and the target's center cannot exceed the dimensions of the target. Figure B-2 shows an accumulated 2-dimensional histogram of the targets' hit offsets relative to the targets' centers. Figure B-3 shows the same data in a 3-dimensional view. The approximate origin-centered bell shape of this data indicates that a user's ability to hit a target will fall off quickly as its size is reduced. Table B.1 quantifies the size a target must be to be successfully hit a given percentage of the time.

Shape Hypothesis Results

We hypothesized that the shape of a target would effect the user's ability to accurately hit it. To determine this, we measured the absolute distance between each target hit location and that target's center. For each shape, we accumulated these offsets and found their averages. The data we collected (given in table B.2) suggest that there is a negligible effect to accuracy between the shapes we tested.

Hit Rate (%)	X-Offset (pixels)	Y-Offset (pixels)
50	10	9
75	16	16
90	22	24
95	27	29
96	28	31
97	29	32
98	31	35
99	35	39
100	43	43

Table B.1: Minimum Offsets for Desired Hit Rate

These data represent the minimum vertical and horizontal offsets required to observe a given successful hit rate. All pixel measurements correspond to a 96 d.p.i. screen. High accuracy (95% and above) is observed for offsets near 30 pixels.

Shape	Median Offset (pixels)
Rectangle	16.52
Ellipse	17.03
Cross	15.65
X	16.55

Table B.2: Median Shape Offsets

These data suggest that shape does not play a large role in effecting user accuracy. Offsets for all shapes are in the range 16.34 ± 0.69 pixels.

Position Hypothesis Results

We hypothesized that targets located closer to the center of the screen would be more accurately hit. Figure B-4 graphs the instantaneous average of the absolute target hit offset versus absolute distance from the center of the screen. The strong upward trend in this graph supports our hypothesis.

Fatigue Hypothesis Results

We hypothesized that as users underwent a prolonged interaction with a touch screen kiosk, their accuracy would decrease as their arms became tired and drifted downwards. Figure B-5 plots the vertical offsets of the target hits versus the time elapsed in each trial. The curves fit to these data points represent the instantaneous level at

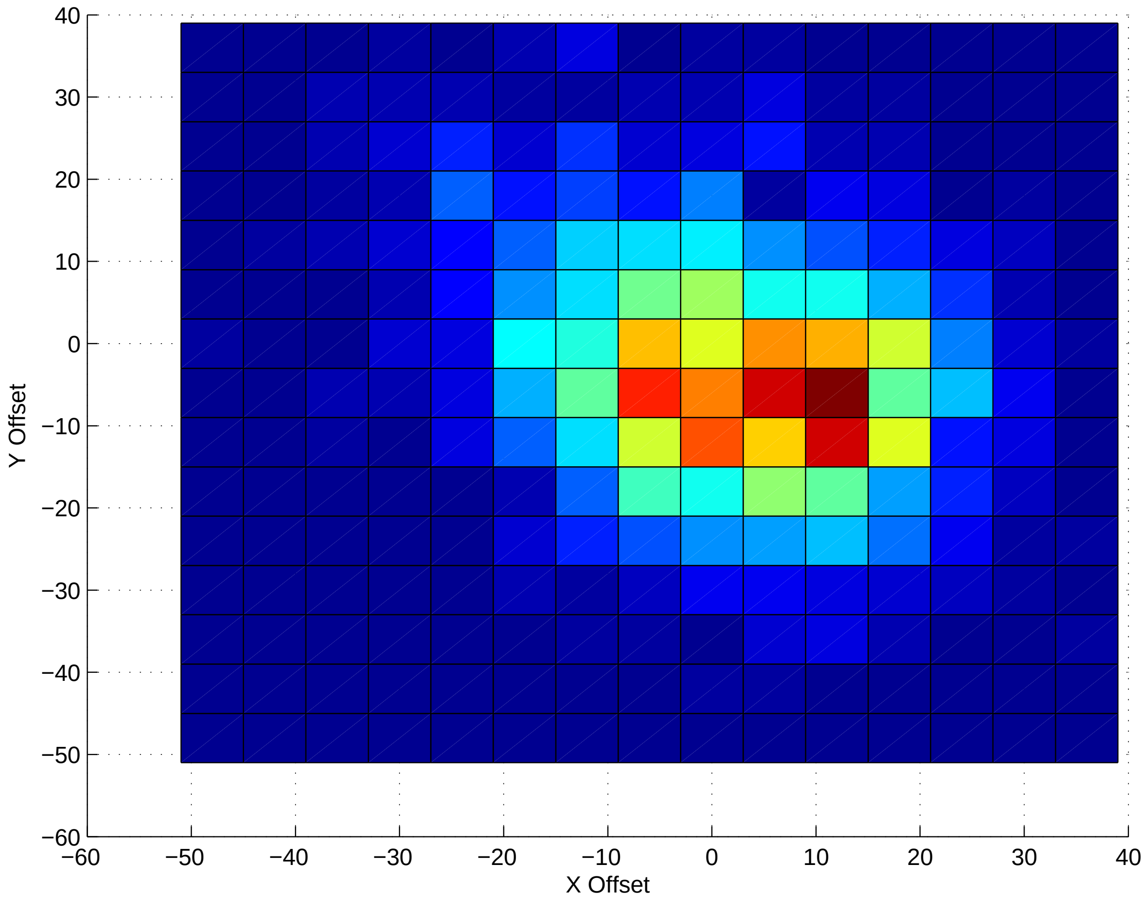

Figure B-2: Offsets Relative to Target Center (2-D)
This figure shows the horizontal (X) and vertical (Y) offsets of users' clicks relative to the center of each target. Nearly all hits are located within a thirty pixel radius of the origin.

which 95% (top), 50% (middle), and 5% (bottom) of data points fall below. None of these lines have an overall trend which varies far from horizontal, therefore showing that fatigue does not significantly effect a user's accuracy during the first three minutes of touch screen interaction.

Time Hypothesis Results

We hypothesized that the longer users took to touch a target, the more accurate they would be. Our approach to gauging this effect was to compare inter-click time to the absolute distance between the target hit and the target's center. The data we plotted in figure B-6 indicate that accuracy improved markedly as inter-click time increased.

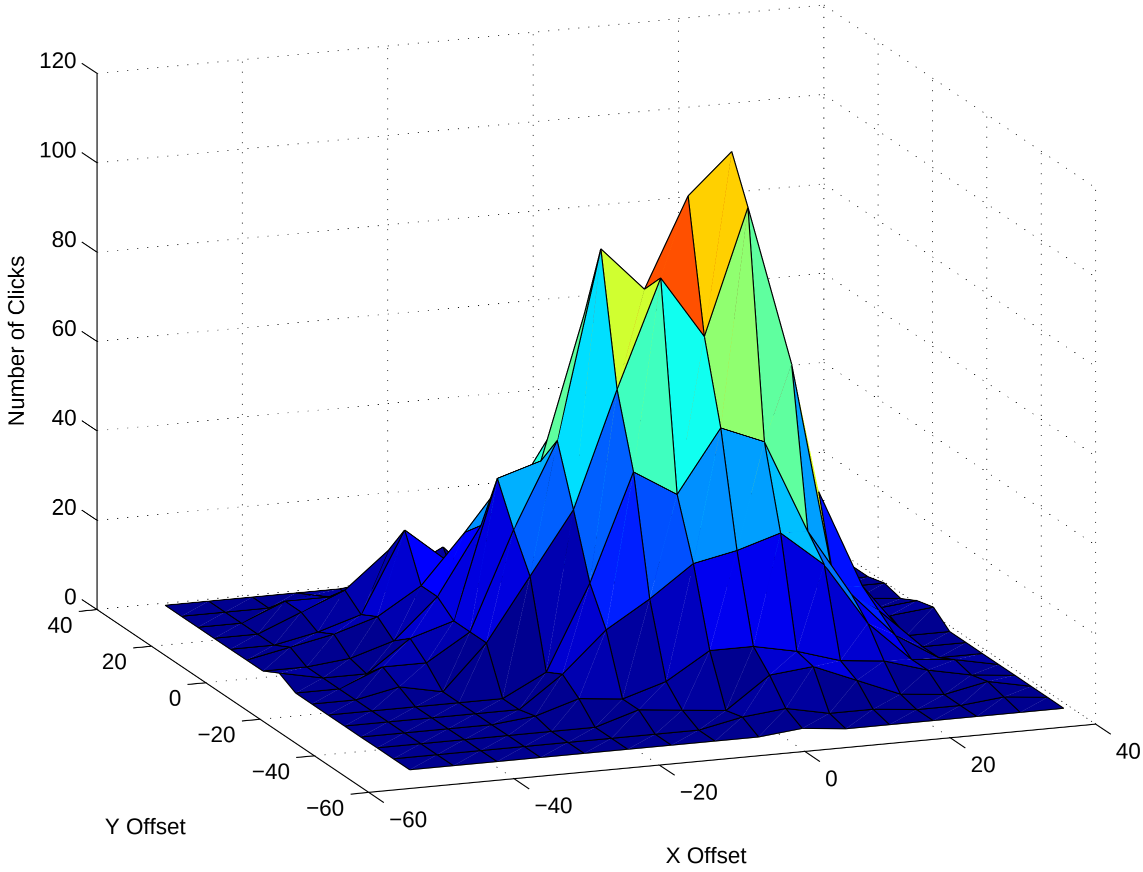

Figure B-3: Offset Relative to Target Center (3-D)
The bell shape of this surface shows that users' accuracy diminishes as the radius of the target decreases.

The fit curves indicate the instantaneous offset at which 95% (top) and 50% (bottom) of data points fall below. The general downward trend of these curves supports this hypothesis.

B.3 Summary

Our study showed that the shape of a target had little impact on a user's ability to hit it. Also, the level of fatigue experienced by users did not significantly effect their accuracy. However, the study results suggest that size, location, and inter-click time are factors that should be considered in interactive touch screen widget design.

When designing the SKINNI GUI, we sought to create widgets that adhered to

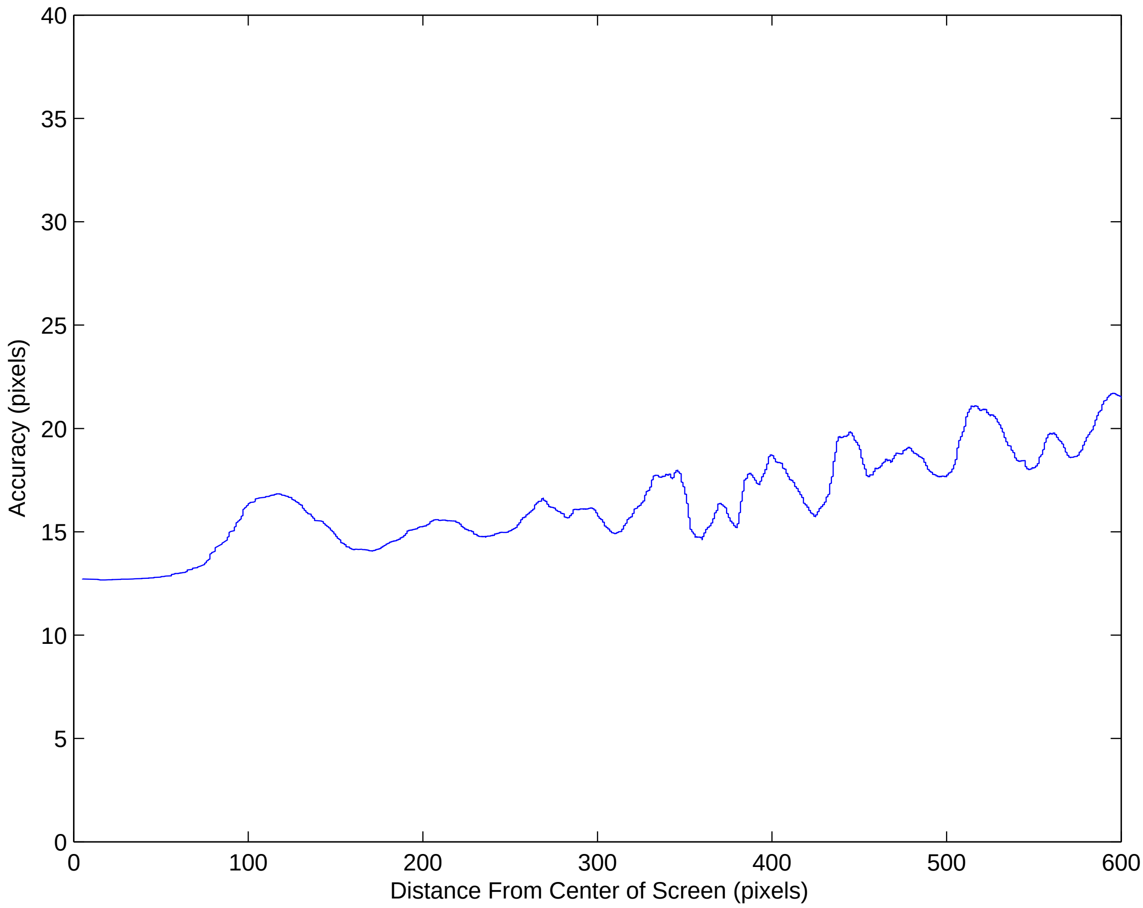

Figure B-4: Offsets Relative to Distance from Screen Center
The strong upward trend of this instantaneous average suggests that a typical user's accuracy decreases as a widget moves away from the center of the screen.

the findings of this user study, especially with regards to size, location, and inter-click time. Unfortunately, we were unable to devise a reliable method for controlling users' inter-click times, so that characteristic was effectively discounted. Additionally, issues of static layout control and aesthetic design limited our ability to centralize widgets or increase their sizes proportionately to their distances from the screen center. This means that size became the driving factor in SKINNI's touch screen widget design.

Our goal was to determine the characteristics of a widget that could be touched by most people nearly all of the time. Since SKINNI's widgets are primarily effected by their size, we will consider our results corresponding to the "Size" hypothesis. Figure B-2 shows that a 30 pixel offset from the origin would encompass all but a few outlying target hits. Table B.1 further supports this observation, suggesting that

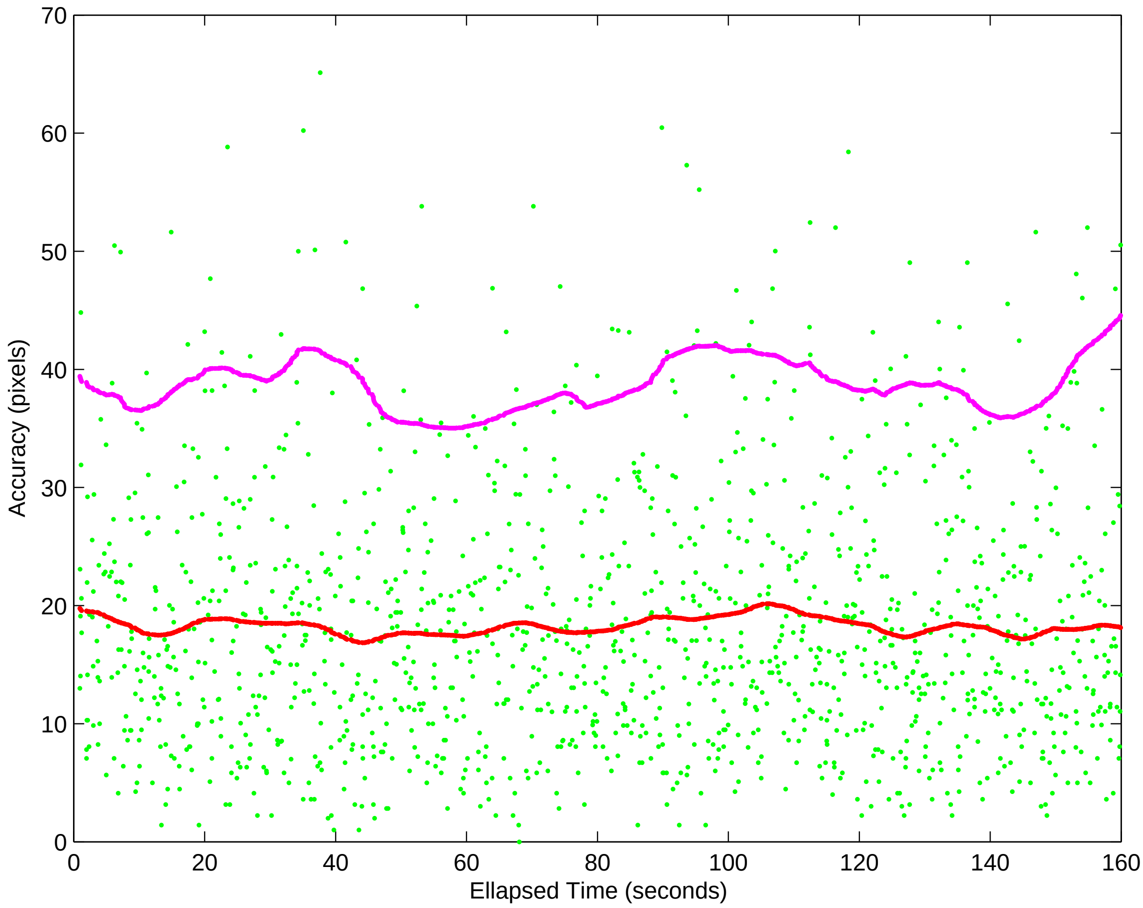

Figure B-5: Fatigue-related Accuracy Loss
The fit curves maintain an almost zero slope, suggesting that fatigue does not effect user accuracy. If our hypothesis held, these curves would trend strongly downwards as the users' arms became tired.

95%–97% of targets that size would be hit successfully in the first attempt. This 30 pixel offset is a radius, so widgets would require a 60 × 60 pixel (0.625 × 0.625 inch on a 96 d.p.i. screen) bounding box to reach the desired level of accuracy.

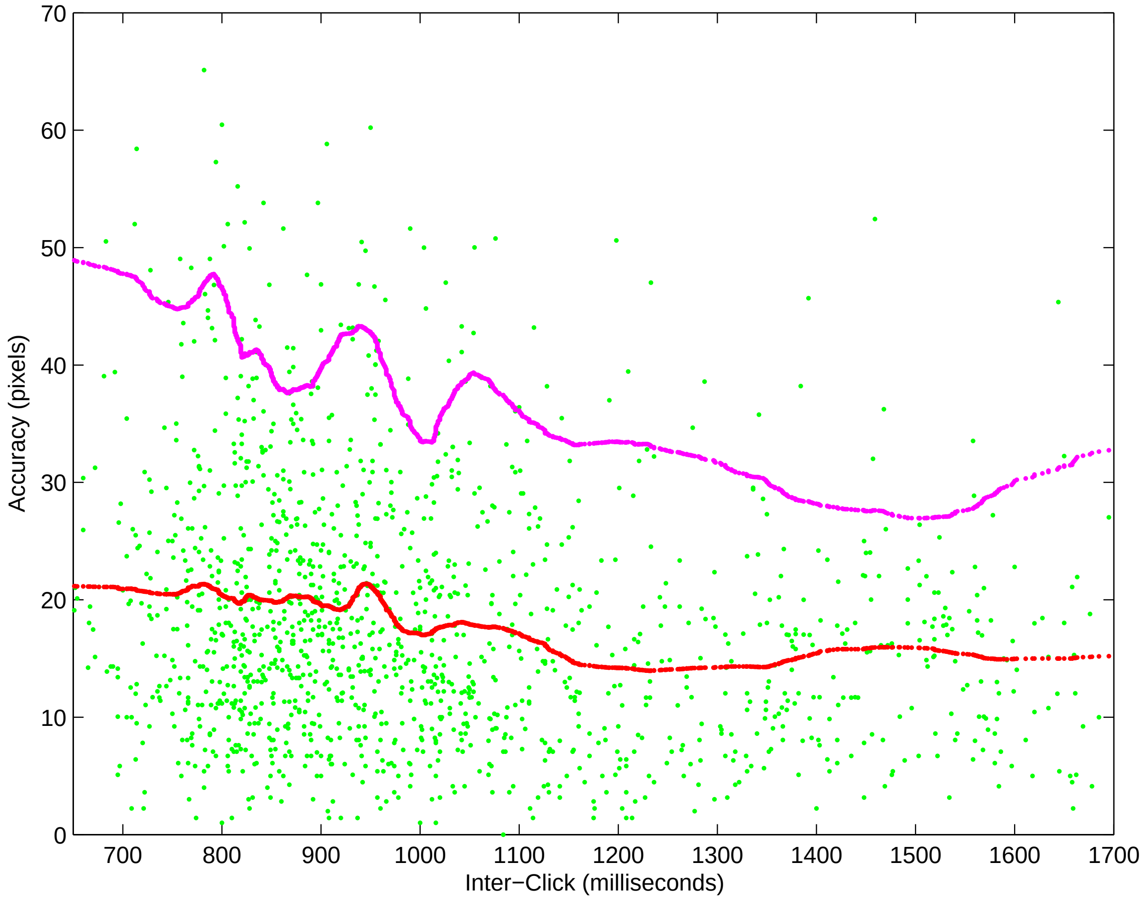

Figure B-6: Relation between Speed and Accuracy
The fit curves begin to trend sharply downwards and then plateau near 1.15 seconds. This suggests that users will be most accurate (able to hit targets closest to their centers) if they take longer than 1.15 seconds to do so.

Appendix C

Full Pane Screenshots

This appendix provides full page screenshots of the SKINNI GUI while each of the five deployed panes is activated:

- Figure C-1 provides a larger view of the "Today" pane as it is shown in figure 2-4. A single event is selected and is displayed in the timeline on the left, the SKINNI list on the right, and the details view on bottom.

- Figure C-2 shows the "Directory" pane with "Boris Katz" selected. The SKINNI table is providing embedded controls that facilitate navigation to related information items.

- Figure C-3 shows the "Map" pane after typical use. Here, the user has found room 32-G708. The user has the option of having this map and written instructions pushed to a Bluetooth® device via the "Lead me..." button near the clock.

- Figure C-4 depicts the "Help" pane after the user has sought information relating to navigating through information on the "Today" pane.

- Figure C-5 provides the entire "Search" pane as it appears after a search for "harvard". All events, people, and locations are provided are related to Harvard in some way.

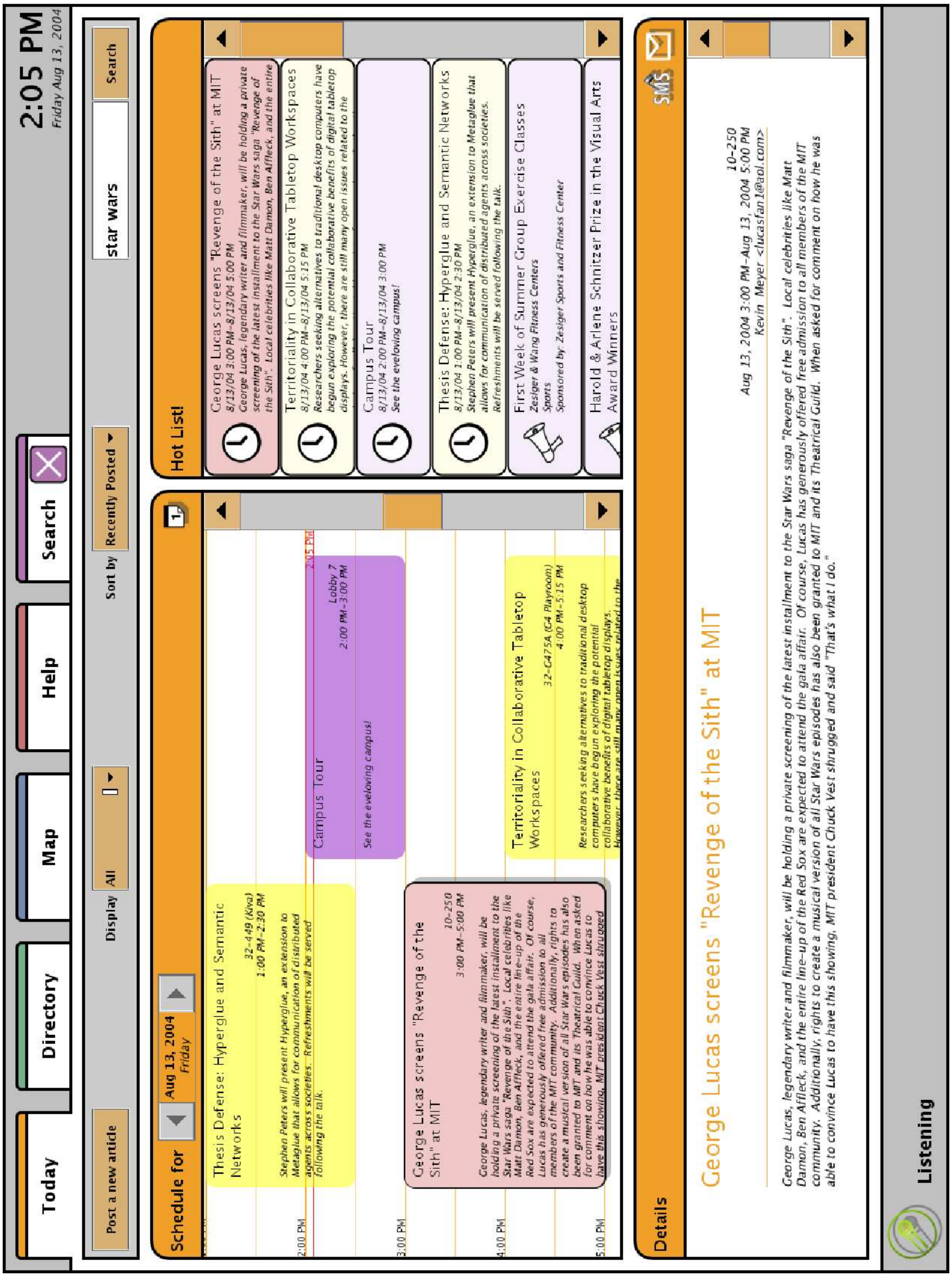

Figure C-1: "Today" Pane

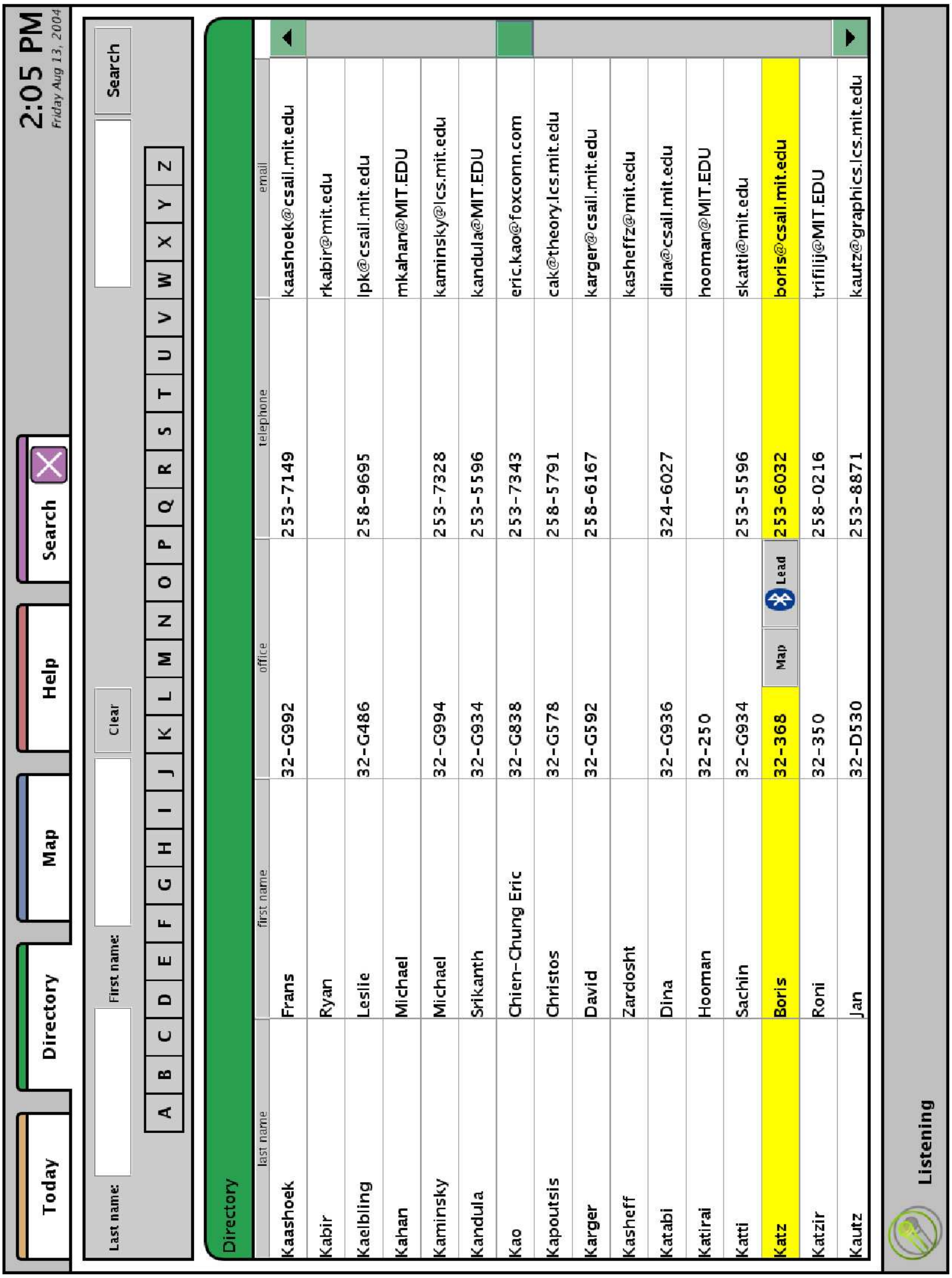

Figure C-2: "Directory" Pane

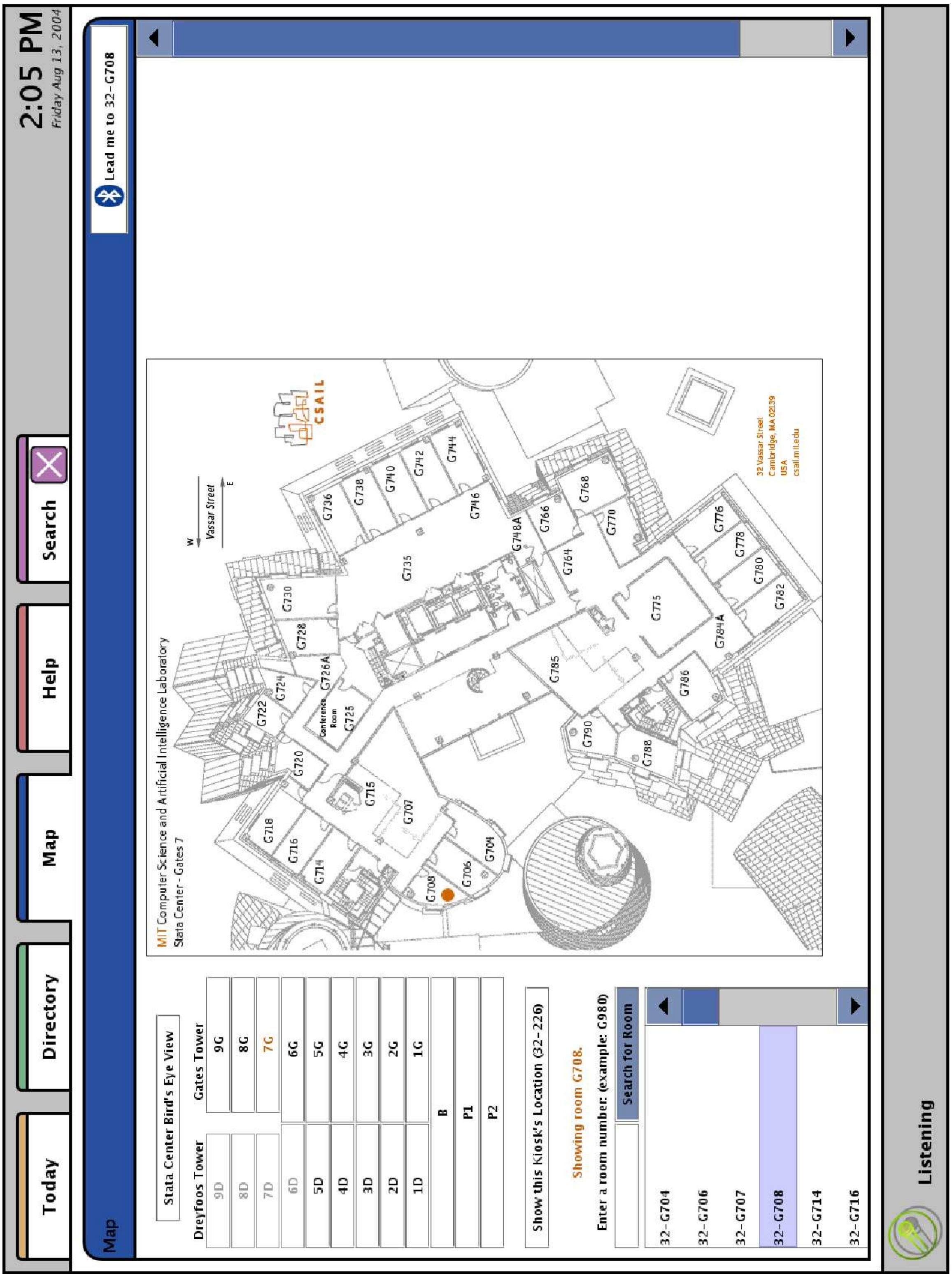

Figure C-3: "Map" Pane

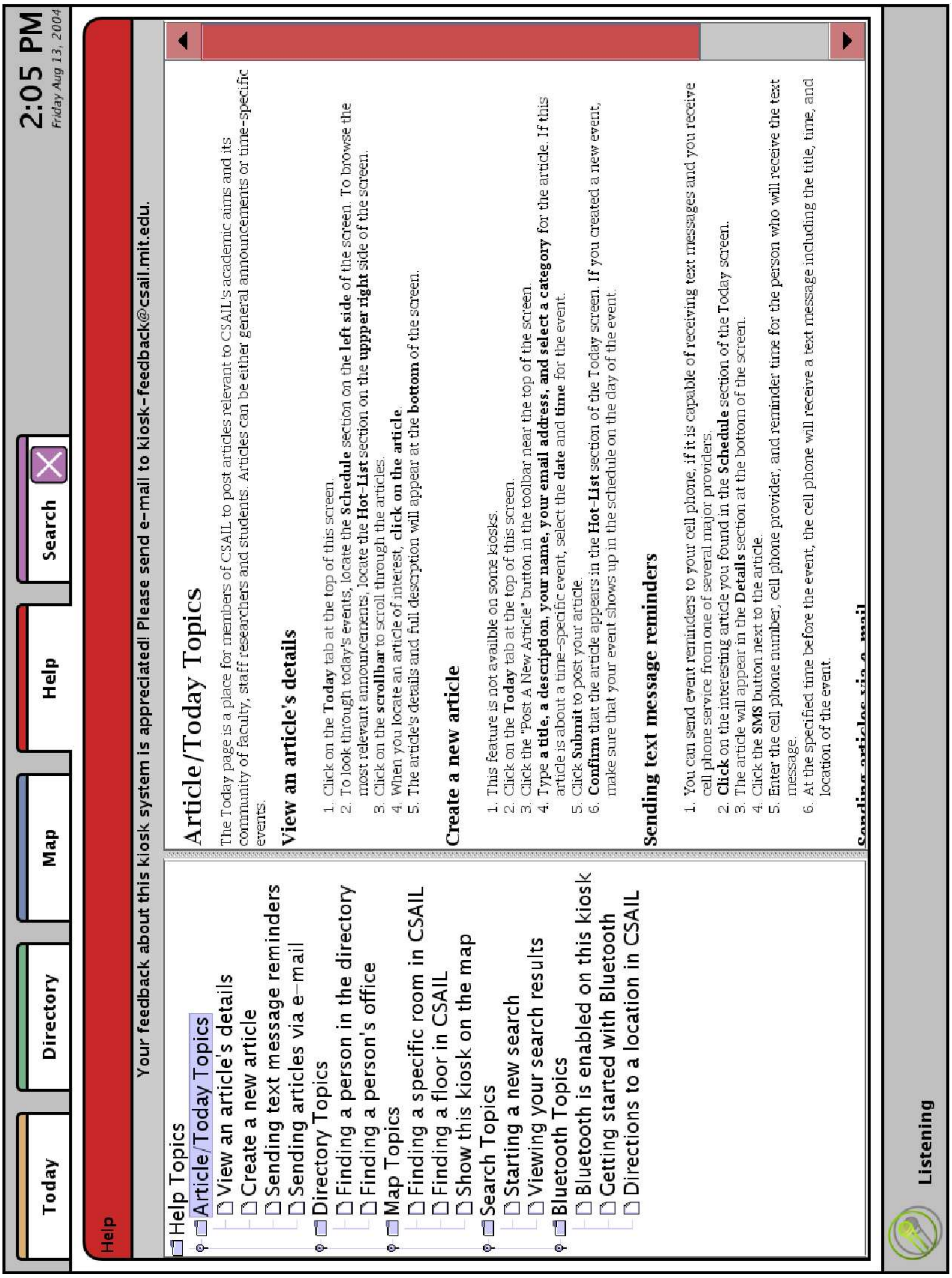

Figure C-4: “Help” Pane

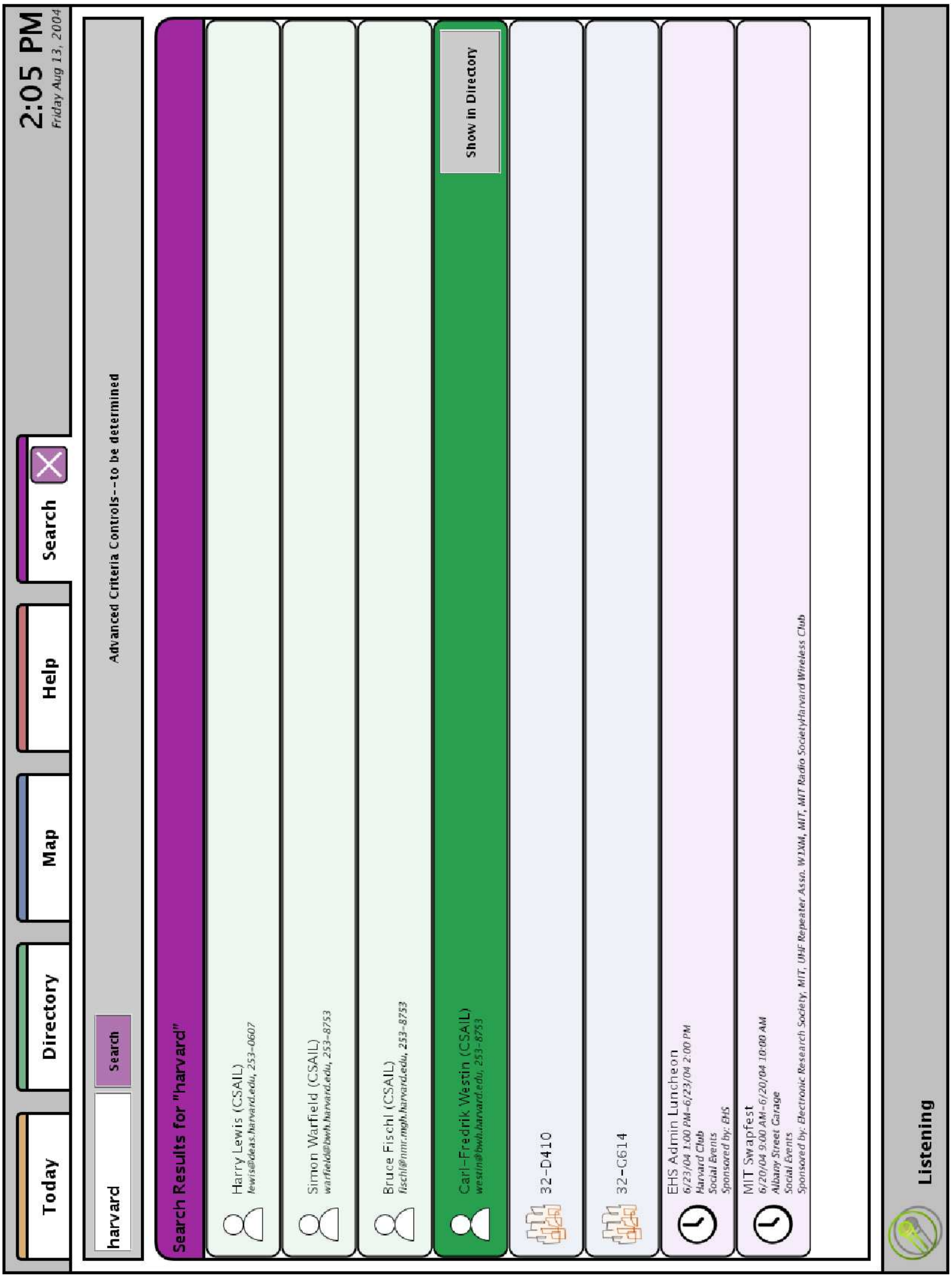

Figure C-5: “Search” Pane

Bibliography

[1] MIT Projecct Oxygen: Overview. http://oxygen.lcs.mit.edu/Overview.html.

[2] MIT Project AIRE. http://www.ai.mit.edu/projects/aire/.

[3] The Oxygen Kiosk Network Project. http://oknet.csail.mit.edu.

[4] B Boehm. A spiral model of software development and enhancement. *SIGSOFT Softw. Eng. Notes*, 11(4):14–24, 1986.

[5] Nathan Eagle, Mav Van Kleek, Alex (Sandy) Pentland, and Howard E. Shrobe. Initiating informal interactions within the enterprise. Originally submitted to UIST 2004. Available at http://www.ai.mit.edu/˜ emax/papers/uist.2004/uist-serendipity.pdf, April 2004.

[6] John D. Gould, Stephen J. Boies, Stephen Levy, John T. Richards, and Jim Schoonard. The 1984 olympic message system: a test of behavioral principles of system design. *Commun. ACM*, 30(9):758–769, 1987.

[7] Jakob Nielsen. Ten usability heuristics. http://www.useit.com/papers/heuristic/heuristic_list.html.

[8] Stephen Peters, Gary Look, Kevin Quigley, Howard Shrobe, and Krzysztof Gajos. Hyperglue: Designing high-level agent communication for distributed applications. Originally submitted to AAMAS '03, 2003.

[9] Stephen Peters and Howie Shrobe. Using semantic networks for knowledge representation in an intelligent environment. In *PerCom '03: 1st Annual IEEE Inter-*

national Conference on Pervasive Computing and Communications, Ft. Worth, TX, USA, March 2003. IEEE.

[10] Stephen L. Peters. *Knowledge Representation and Resource Management in Distributed, Pervasive, Intelligent Environments*. PhD thesis, MIT, 2004.

[11] Brenton Phillips. Metaglue: A programming language for multi-agent systems. M.eng thesis, MIT, 1999.

[12] R. Reddy, L. Erman, R. D. Fennell, and R. B. Neely. The HEARSAY speech understanding system: An example of the recognition process. In *IEEE Transactions on Computers*, pages 427–431, 1976.

[13] Paul Robertson, Robert Laddaga, and Max Van Kleek. Virtual mouse vision based interface. In *9th International Conference on Intelligent User Interfaces*, January 2004.

[14] Stephanie Seneff, Chian Chuu, and D. Scott Cyphers. Orion: From on-line interaction to off-line delegation. In *Proceedings of the 6th International Conference on Spoken Language Processing*, Beijing, China, October 2000.

[15] Stephanie Seneff and Joseph Polifroni. Dialogue management in the mercury flight reservation system. In *Satellite Dialogua Workshop, ANLP-NAACL*, April 2000.

[16] Spoken Language Systems. Sls :: Research initiatives :: Applications. http://www.sls.csail.mit.edu/sls/applications/.

[17] Max Van Kleek. k:info: An architecture for smart billboards for informal public spaces. In *UBICOMP 2003*, Seattle, WA.

[18] Max Van Kleek. Intelligent environments for informal public spaces: the ki/o kiosk platform. M.Eng. Thesis, Massachusetts Institute of Technology, Cambridge, MA, February 2003.

[19] Max Van Kleek, Buddhika Kottahachchi, Tyler Horton, and Paul Cavallaro. Designing speech interfaces for kiosks. In *MIT Student Oxygen Workshop*, Cambridge, MA, September 2004.

[20] Bruce Tognazzini. First principles of interaction design. http://www.asktog.com/basics/firstPrinciples.html.

[21] Christopher Varenhorst. Passdoodles: A lightweight authentication method. http://oknet.csail.mit.edu/papers/varenhorst.pdf, 2004.

[22] Eugene Weinstein. SpeechBuilder: Facilitating spoken dialogue system development. Master's thesis, Massachusetts Institute of Technology, May 2001.

[23] Mark Weiser. The computer for the 21st century. *Scientific American*, (94–100), September 1991.

[24] Victor Zue, Stephanie Seneff, James Glass, Joseph Polifroni, Christine Pao, Timothy J. Hazen, and Lee Hetherington. Jupiter: A telephone-based conversational interface for weather information. In *IEEE Transactions on Speech and Audio Processing*, volume 8, pages 100–112, January 2000.