Handbook of Research on User Interface Design and Evaluation for Mobile Technology

Volume I

Joanna Lumsden
National Research Council of Canada
Institute for Information Technology – e-Business, Canada

Information Science REFERENCE
INFORMATION SCIENCE REFERENCE
Hershey · New York

Acquisitions Editor: Kristin Klinger
Development Editor: Kristin Roth
Senior Managing Editor: Jennifer Neidig
Managing Editor: Sara Reed
Copy Editor: Joy Langel, Katie Smalley, and Angela Thor
Typesetter: Jeff Ash
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

and in the United Kingdom by
Information Science Reference (an imprint of IGI Global)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Handbook of research on user interface design and evaluation for mobile technology / Joanna Lumsden, editor.

p. cm.

Summary: "This book provides students, researchers, educators, and practitioners with a compendium of research on the key issues surrounding the design and evaluation of mobile user interfaces, such as the physical environment and social context in which a device is being used and the impact of multitasking behavior typically exhibited by mobile-device users"--Provided by publisher.

Includes bibliographical references and index.

ISBN 978-1-59904-871-0 (hardcover) -- ISBN 978-1-59904-872-7 (ebook)

1. Mobile computing--Handbooks, manuals, etc. 2. Human-computer interaction--Handbooks, manuals, etc. 3. User interfaces (Computer systems)--Handbooks, manuals, etc. I. Lumsden, Joanna.

QA76.59.H36 2008

004.165--dc22

2007024493

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter XXI
Tools for Rapidly Prototyping Mobile Interactions

Yang Li
University of Washington, USA

Scott Klemmer
Stanford University, USA

James A. Landay
University of Washington & Intel Research Seattle, USA

ABSTRACT

We introduce informal prototyping tools as an important way to speed up the early-stage design of mobile interactions, by lowering the barrier to entry for designers and by reducing the cost of testing. We use two tools, SUEDE and Topiary, as proofs of concept for informal prototyping tools of mobile interactions. These tools address the early stage design of two important forms of mobile interactions: speech-based and location-enhanced interactions. In particular, we highlight storyboarding and Wizard of Oz (WOz) testing, two commonly used techniques, and discuss how they can be applied to address different domains. We also illustrate using a case study: the iterative design of a location-enhanced application called Place Finder using Topiary. In this chapter we hope to give the reader a sense of what should be considered as well as possible solutions for informal prototyping tools for mobile interactions.

INTRODUCTION

The iterative process of prototyping and testing has become an efficient way for successful user interface design. It is especially crucial to explore a design space in the early design stages before implementing an application (Gould et al., 1985). Informal prototyping tools can speed up an early-stage, iterative design process (Bailey et al., 2001; Klemmer et al., 2000; Landay et al., 2001; Li et al., 2004; Lin et al., 2000). These tools are aimed at lowering the barrier to entry for interaction designers who do not have technical backgrounds, and automatically generating early-stage prototypes that can be tested with end users. The informal look and feel of these tools and their fluid input techniques, for example using pen sketching (Landay et al., 2001), encourage both designers and end users to focus on high level interaction ideas rather than on design or implementation details (e.g., visual layouts or colors). These details are often better addressed at a later stage. In this chapter, we focus on informal tool support for the early stage design of interactive mobile technologies. In particular, we describe informal prototyping tools that we developed for two increasingly important forms of mobile interaction: speech-based interactions (Klemmer et al., 2000) and location-enhanced interactions (Li et al., 2004).

The first of these two types of interactions, speech-based, works well on mobile phones, the major platform of mobile computing. These devices often have tiny screens and buttons to increase mobility, which makes speech interaction an important alternative. Although the accuracy of speech recognition is an important concern for a successful speech-based UI, the real bottleneck in speech interface design is the lack of basic knowledge about user "performance during computer-based spoken interaction" (Cohen et al., 1995). Many interaction designers who could contribute to this body of knowledge are excluded from speech design by the complexities of the core technologies, the formal representations used for specifying these technologies, and the lack of appropriate design tools to support iterative design (Klemmer et al., 2000). SUEDE (Klemmer et al., 2000) demonstrates how tool support can be used in the early stage design of speech-based user interfaces.

The second of these two types of interactions, location-enhanced, is important because of its implicit nature. While the explicit input channels (e.g., keyboarding or mouse pointing) available on mobile technology are more limited than on the desktop, the bandwidth of implicit input (using contextual information) is greatly expanded on mobile platforms. Mobile technology is more available in our context-rich, everyday lives than traditional desktop computing. One especially promising form of context-aware computing that has begun to see commercialization is location-enhanced computing, applications that leverage one's current location as well as the location of other people, places, and things (Li et al., 2004). For example, mobile phone services allow users to locate friends and family (LOC-AID), provide real-time navigation (InfoGation) and monitor and motivate users toward their fitness goals by using phone-based GPS to measure the user's speed, distance and elevation (BonesInMotion). E911 transmits a mobile phone user's current location when making emergency calls. However, location-enhanced applications are hard to prototype and evaluate. They employ sophisticated technologies such as location tracking and their target environment is mobile and in the field. Topiary (Li et al., 2004) demonstrates how high-level tool support can be provided for lowering the threshold and cost for designers to design and test location-enhanced applications.

Using SUEDE and Topiary as proofs of concept, we highlight two techniques commonly used in informal prototyping tools: storyboarding and Wizard of Oz (WOz) testing. To overcome the technical barrier for design, both SUEDE and Topiary employ a storyboarding-based approach for specifying interaction logic. To allow easy testing of prototypes, both tools employ WOz approaches where a human wizard simulates a sophisticated, nonexistent part of the prototype such as location tracking or speech recognition. To demonstrate how these types of tool can actually help prototype and test mobile technology, we introduce a case study using Topiary to design the Place Finder application.

BACKGROUND

User interface tools have been a central topic in HCI research. An extensive review of user interface tools can be found in (Myers et al., 2001). A large number of research prototypes and commercial products have been developed for rapid prototyping of user interfaces (Apple, 1987; Bailey et al., 2001; Hartmann et al., 2006; Klemmer et al., 2000; Landay et al., 2001; Li et al., 2004; Lin et al., 2000; Macromedia; MacIntyre et al., 2004).

In particular, informal prototyping tools are aimed at the early stages of a design process, and are used to create early-stage prototypes for testing key design ideas rather than building full-fledged final systems (Landay et al., 2001). They often result in example-based interface mockups that are able to demonstrate exploratory interactive behaviors but ignore other non-exploratory aspects of a desired system. Informal tools have shown great potential to facilitate the early stages of a design process and have been developed for various domains. For example, SILK is a tool for designing graphical user interfaces (Landay et al., 2001) that allows designers to create GUI prototypes by sketching and storyboarding. DENIM (Lin et al., 2000), a tool for the early stage design of Web sites, has become one of the most popular informal prototyping tools (downloaded over 100,000 times since 2000). Informal prototyping tools are often grounded in current practices of designers, e.g., paper prototyping (Rettig, 1994; Snyder, 2003), and lower the barrier to entry by maintaining the affordance of an existing practice. At the same time, informal tools provide extra value by allowing the easy editing and maintenance of a design, and by generating testable prototypes.

MAIN FOCUS OF THE CHAPTER

In our research, two features have emerged as being particularly valuable for rapidly prototyping mobile interactions. The first is storyboarding, which is inspired by traditional paper prototyping where designers draw key interaction flows visually on paper. Storyboarding is enhanced by electronic tool support to create the states and transitions. Many systems have been influenced by Harel's Statecharts model (Harel, 1987). Storyboarding is employed by both SUEDE and Topiary to lower the technical barrier for creating early-stage prototypes.

The second valuable feature is Wizard of Oz (WOz) testing, where a designer simulates part or all of the application logic by manipulating the interface in response to user input. This significantly reduces the time and labor required to create a testable prototype. As both speech-based interfaces and location-enhanced computing involve a necessary but sophisticated component, that is speech recognition and location tracking, respectively, both SUEDE and Topiary employed a WOz approach to avoid the complexity of introducing these components. To give an example of how this type of tool can help design and evaluate mobile technology in practice, we describe a case study for the iterative design of a PDA-based mobile Place Finder application using the Topiary.

Prototyping with Storyboards

In the early stages of design, it is important that tools allow designers to focus on the high-level concerns of interaction design, rather than forcing designers to also specify how these interactions are implemented. Storyboarding is an efficient way for designers to describe how a user interface should behave by enumerating concrete interaction sequences including both user input and interface output. These sequences should cover the key interaction paths of a proposed system in a particular design space. The concerns of early-stage prototyping are distinct from those of constructing an actual system, which focus more on completeness than exploration of the design space.

SUEDE allows two kinds of storyboarding: linear (conversation examples) and non-linear (design graphs of an actual interface) storyboarding. Designers start a design by creating simple conversation examples (see the Script Area at the top of Figure 1). These examples then evolve into the more complex, graph structure representing the actual interface design (see the design graph at the bottom of Figure 1) (Klemmer et al., 2000). The process of creating linear examples first and then forming more general design graphs is based on the existing practices of speech UI designers: we have found that often, designers begin the design process by writing linear dialog examples and then use those as a basis for creating a flowchart representation of the dialog flow on paper.

Designers lay out linear conversation examples horizontally as cards in the script area. *Prompts*, colored orange, represent the system's speech prompts. They are recorded by the designer for the phrases that the computer speaks. *Responses*, colored green, represent example responses of the end user. They are the phrases that participants make in response to prompts. System prompts alternate with user responses for accomplishing a task. A designer can record her own voice for the speech on both types of cards, as well as type in a corresponding label for each of the cards. By playing the recordings from left to right, the designer can both see and hear the example interaction. For example, in Figure 1, a designer has recorded a conversation example with the following alternating prompts and responses: "message from James," "erase it," "Are you sure," "Yes." After constructing example scripts, a designer can construct an actual design of a speech-based interface using the design graph (see Figure 1). A design graph represents a dialog flow based on the user's responses to the system's prompts. To create a design graph, designers can drag prompt or response cards from a script onto the design area, or create new cards on the design area, and link them into the dialog flow. SUEDE's storyboard mechanism embodies both the input and output of a speech interface in cards that can be directly manipulated (e.g., via drag & drop), and hides the complexity of using speech recognition and synthesis. This abstraction allows designers to focus on high-level design issues.

Topiary's storyboards also embed the specification of input and output interactions into a storyboard. Before introducing Topiary's storyboards, we first discuss Topiary's *Activity Map* workspace, a component designed for creating scenarios describing location contexts of people, places and things by demonstration (see Figures 2 and 3). The created scenarios can be used as input by Topiary storyboards when prototyping location-enhanced interactions (see Figure 4). Modeling implicit input, location context in this case, is a new challenge posed by mobile computing.

Topiary's Activity Map workspace employs an intuitive map metaphor for designers to demonstrate location contexts describing the spatial relationship of people, places and things. Designers can create graphical objects on the map to represent people, places and things (see Figure 2). Designers can move people and things on the map to demonstrate various spatial relationships. For example, in Figure 2, Bob is out of the library, the astronomy building and the café. However, Bob is close to the library because Bob's proximity region, indicated by the red circle around Bob, intersects with the library. The proximity region

Figure 1. SUEDE allows designers to create example scripts of speech-based interactions (top) and speech UI designs (bottom) by storyboarding.

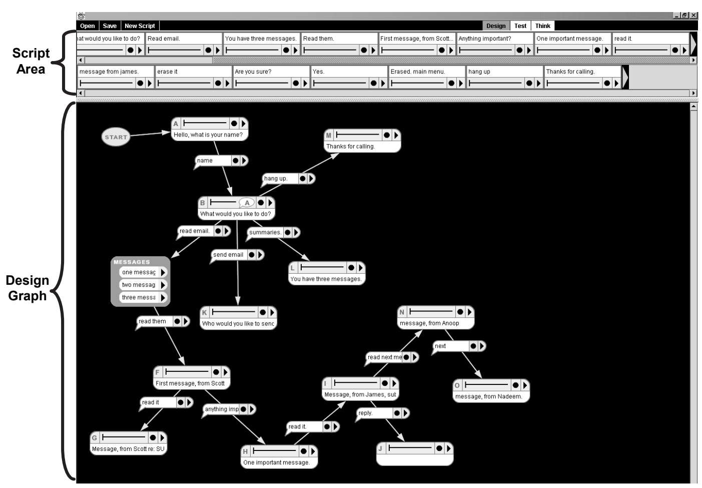


Figure 2. The active map workspace of Topiary is used to model location contexts of people, places and things and to demonstrate scenarios describing location contexts.

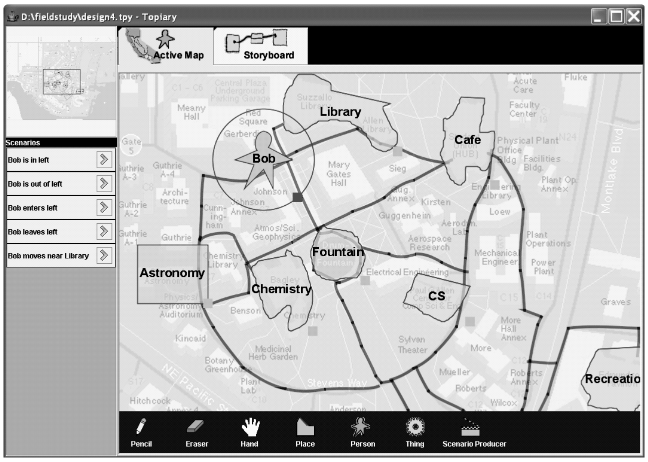
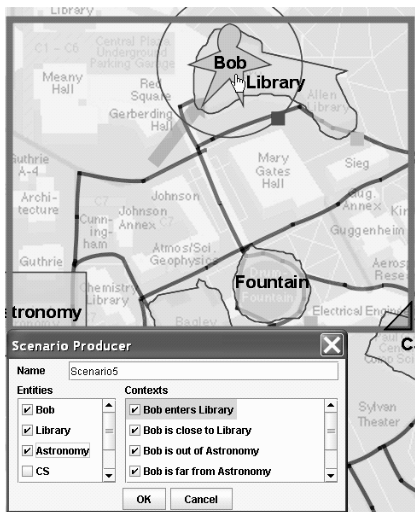


Figure 3. The designer drags Bob into the Library, with the context changing from "Bob is out of Library" to "Bob enters Library." As the entity CS (building) is unchecked, all related contexts to this place are filtered out.



can be resized by dragging the rectangular handle. These spatial relationships can be captured via Topiary's *Scenario Producer*. Like a screen capture tool, a designer can position a *Scenario Producer* window over entities of interest to capture spatial relationships (see Figure 3). A dialog box is then brought up that allows designers to select contexts of interest. Designers can demonstrate dynamic contextual transitions such as "entering" or "leaving" by moving entities within the recording window. For example, dragging Bob into the Library changes the event "Bob is out of Library" into "Bob enters Library" (see Figure 3).

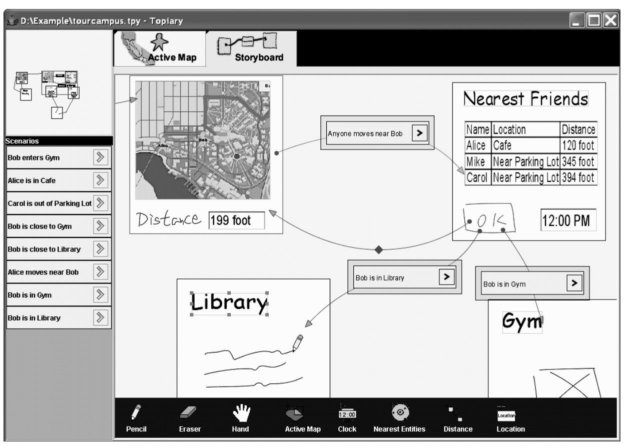
Based on the location scenarios captured in the Active Map workspace, designers can create application prototypes in the Storyboard workspace (see Figure 4). In Topiary, a storyboard page represents a screen of visual output and a link represents transitions between pages. The key innovation in Topiary's storyboards is that scenarios created in the *Active Map* workspace can be used as conditions or triggers on links (Li et al., 2004). Designers create pages and links by sketching. Topiary has two kinds of links (see Figure 4). *Explicit links*, denoted in blue, start on ink within a page and they represent GUI elements that users have to click on, for example buttons or hyperlinks. *Implicit links*, denoted in green, start on an empty area in a page. They represent transitions that automatically take place when scenarios associated with that link occur. Explicit links model explicit interactions taken by end-users though they can be conditioned by sensed information, whereas implicit links model purely sensed data such as locations. One or more scenarios can be added to a link and multiple scenarios represent the logical AND of the scenarios. Multiple links starting from the same source represent the logical OR of transitions.

The Activity Map abstraction allows designers to focus on location contexts of interest rather than how these contexts can be sensed. Topiary's graphical storyboarding allows designers to specify rich interactions by drag & drop or sketching instead of specifying complex rules or Boolean logic expressions. From both SUEDE and Topiary, we conclude that the key to a successful informal tool is to devise an appropriate abstraction that matches designers' conceptual model for design and hides the less important aspects of exploring target interactions. Storyboards, as a meta-design metaphor, should be adapted and developed to fit within a specific domain when being applied.

Testing Using WOz Approaches

Speech-based or location-enhanced interactions resist rapid evaluation because the underlying technologies require high levels of technical expertise to understand and use, and a significant amount of effort to tune and integrate. For example, location-tracking infrastructures are not always available (e.g., GPS does not work well indoors) and they require a great deal of effort to deploy and configure. Incorporating these technologies too early in a design process may distract designers from fully exploring the design space. Consequently, we employed WOz approaches in these tools for testing early-stage prototypes. That is, a wizard (played by a designer or experimenter) simulates what these technologies would do in the final application.

Figure 4. Topiary's Storyboard workspace allows application prototypes to be created. The lower three links (in blue) are explicit links, representing the behavior of the OK button depending where "Bob" is. The top link (in green) is an implicit link, representing an automatic transition from the Map page to the Nearest Friends page when "Anyone moves near Bob"



Wizard of Oz (WOz) testing has been widely employed in user interface design. In a WOz test, a wizard (often played by a designer) fakes an incomplete (or nonexistent) system component to conduct early user evaluations (Dahlbäck et al., 1993). In its most basic form, a WOz test works by the wizard simulating the machine behavior entirely. There is no computation in the loop at all. Examples of this form include testing paper prototypes by having the wizard physically move around the paper-based windows and menus (Rettig, 1994) and testing potential speech interface interaction flows by having a human operator on the other side of the telephone, following a pre-specified interaction graph. When an interactive prototype has been created (at least partially), the wizard can simply use the implemented interface. As a variant of this approach, a programmer can implement a functionally complete but suboptimal interface, and have the wizard control this interface during testing as a means of eliciting users' conceptual models of the task for example (Akers, 2006).

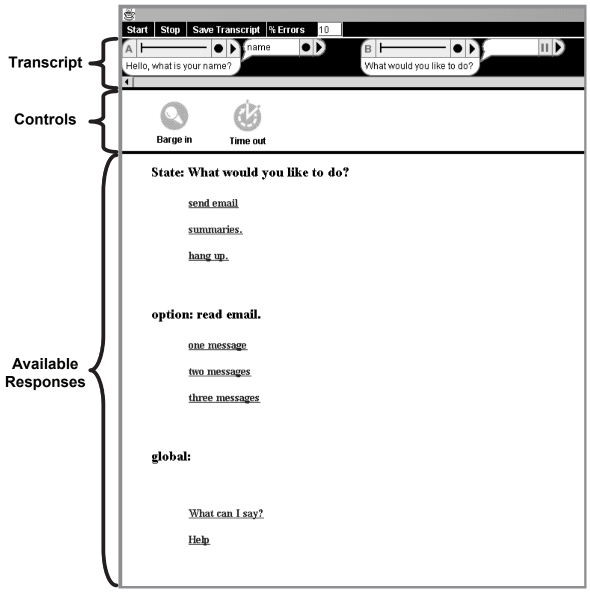
Significant gains beyond these basic approaches can be achieved through tools designed explicitly to support a Wizard of Oz approach. The fundamental insight behind a WOz-enabled tool is that the wizard is provided with a distinct user interface from that of the end user, and that the primary goal for this interface is to enable the wizard to rapidly specify to the system what the user's input was. In SUEDE, the interaction flow and audio prompts are specified by the designer ahead of time, and the user's responses to the speech prompts are interpreted by the wizard and specified to the system using a graphical interface that is runtime-generated based on the user's current state within the interaction flow. During a test, a wizard works in front of a computer screen. The participant performs the test away from the wizard, in a space with speakers to hear the system prompts and a microphone hooked up to the computer to record his responses. During the course of the test session, a transcript sequence is generated containing the original system audio output and a recording of the participant's spoken audio input.

When the wizard starts a test session, SUEDE automatically plays the pre-recorded audio from the current prompt. The wizard interface in SUEDE displays hyperlinks that represent the set of possible options for the current state (see Figure 5); the wizard waits for the test participant to respond, and then clicks on the appropriate hyperlink based on the response. Here, the wizard is acting as the speech recognition engine. Additionally, effective wizard interfaces should provide a display of the interaction history (as well as capture this for subsequent analysis); global controls for options generally available in an interface genre but independent of a particular interface or interface state (these globals can be defined by the tool or specified by the designer); and support for simulated recognition errors. This set of functionality enables the wizard to customize the test as she sees fit, handle user input beyond what was originally designed, and test whether the application is designed in such a way that users can understand and recover from "recognition errors."

Location-enhanced interfaces introduce the additional challenge that, almost by definition, a test must be conducted while moving to be ecologically valid. To address this, Topiary's WOz interface was specifically designed for a wizard to interact with the interface while walking. Topiary automatically generates user interfaces for testing, including the Wizard UI and the End-User UI, based on the Active Map and the Storyboard workspace. The Wizard UI (see Figure 6) is where a wizard simulates location contexts, as well as observes and analyzes a test. The End-User UI is what an end user interacts with during a test and it is also shown in the End User Screen window of the Wizard UI (see Figure 6) so that designers can monitor user interactions. The designer can also interact with the End-User Screen window for debugging purposes. The Wizard UI and End-User UI can be run on the same device (to let a designer try out a design) or on separate devices (one for the Wizard, the other for the user).

During a test, the wizard follows a user; each carries a mobile device, and these devices are connected over a wireless network. The wizard

Figure 5. SUEDE's Test mode is presented in a web browser and allows the wizard to focus on the current state of the UI (top) and the available responses for that state (bottom).



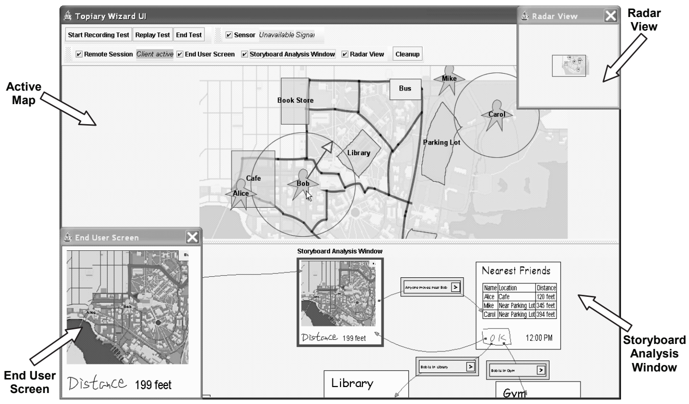
simulates location contexts by moving people and things around on the Active Map to dynamically update their location. The location changes of people and things on a map may trigger implicit transitions in the storyboard that will update the End-User UI. Topiary can also employ real location data if it is available, for more realistic testing at larger scales. A designer can choose to turn on a built-in location-tracking engine, based on Place Lab (LaMarca et al., 2005), which allows a WiFi-enabled or GSM-enabled device to passively listen for nearby access points to determine its location in a privacy-sensitive manner. In addition, a designer can analyze a design by recording a test and replaying it later. Topiary capture users' actions, like mouse movements and clicks, as well as physical paths traveled. The Storyboard Analysis window (see the bottom of Figure 6) highlights the current page and the last transition during a test or a replay session, which can help designers to figure out interaction flows.

Through our experience building SUEDE and Topiary, we have learned that effective tool support for Wizard of Oz testing comprises several key elements: the current state of the user interface (e.g., what is the current page in both tools), the current state of the user (e.g., the user's current location in Topiary) and the set of available actions (e.g., available responses in SUEDE). These elements should be provided to the wizard in an effective manner that allows the wizard to easily grasp and rapidly react. An effective Wizard interface should minimize the wizard's cognitive load by proactively maintaining a visible representation of state and having the displayed (and hence selectable) options for future action tailored to the state at hand.

A Case Study

To demonstrate how an informal prototyping tool can help at an early stage of the design process and how informal prototyping can inform the later design or development process, we report on our experience with the iterative design of a location-enhanced Place Finder using Topiary. A location-enhanced Place Finder embodies many features of location-enhanced, mobile applications. It allows users to find a place of interest more efficiently by leveraging the user's location (e.g., showing a

Figure 6. The Wizard UI has four major parts: The Active Map (a clone of the Active Map workspace) for simulating location contexts, the End User Screen for monitoring a user's interaction or debugging a design, the Storyboard Analysis Window for analyzing interaction logic and the Radar View for easy navigation of the Activity Map.



path to the destination). With the help of Topiary, we were able to efficiently explore the usability issues of map-based navigation techniques on a PDA held by a user walking in the field. Map-based navigation is a key component of a Place Finder application. Based on two design iterations that involved creating five different designs and testing them with four participants in the field as well as an analysis of implementation issues, we built a high fidelity prototype of the Place Finder.

The first iteration included four different user interface designs that shared the same the underlying map of places and paths in the Active Map workspace (see Figure 2). At each iteration, a user test was conducted in the field on a college campus, using a Toshiba Tablet PC and an HP iPAQ™ Pocket PC. During each test, the wireless communication between the two devices was based on a peer-to-peer connection so that the connection was not affected by the availability of access points in the field.

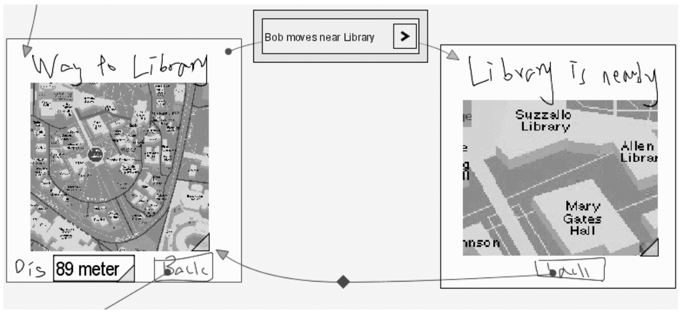
Iteration #1

It took us only three hours in total to create four prototypes, each using a different navigation technique. The first design shows a map of the entire campus (see Figure 7a). The second design shows an area centered on the user and lets the user manually zoom in and out (see Figure 7b). The third design uses the user's current location to show different regions of the campus (see Figure 7c). The last design is similar to the second, except it automatically zooms in or out based on the user's current speed (see Figure 7d). This last design was based on the idea that people are reluctant to interact with a device while walking. All four designs showed the user's current location and shortest path (see the thick pink lines in Figure 7) to the target, both of which are updated dynamically by Topiary.

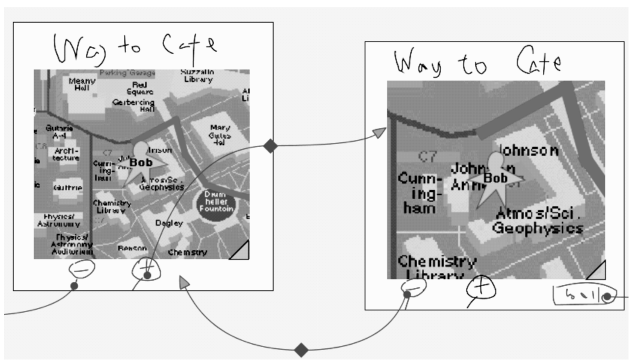
Four navigation segments were included in the test of Iteration #1, one segment for each of the four designs. These four segments were selected based on two principles. First, to smoothly connect the four experimental segments, the target of a segment should be the starting point of the following segment. Second, each segment should cover an area that requires a moderate walk, not too long or too short (e.g., an eight minute walk), and can produce a path with enough complexity to avoid simple paths (e.g., the entire path is a straight line.)

We had three participants try all four designs on a PDA in the field, with a wizard updating their location on a Tablet PC. Each experimental ses-

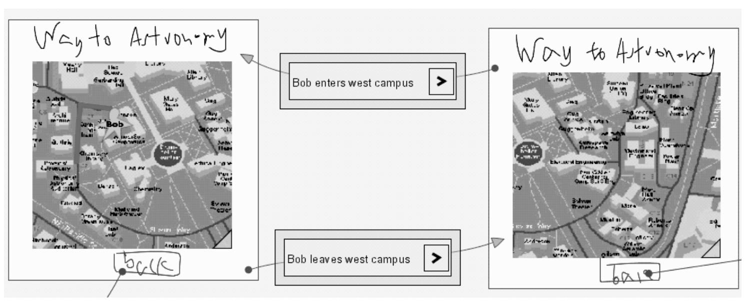
Figure 7. Storyboard fragments of the four designs in Iteration #1. A page, which holds maps and sketches, represents a screen of visual output of the user interface. Arrows (links) between pages represent transitions. The blue links represent GUI elements such as buttons for which scenarios can be used as conditions (not shown here). The green links represent transitions that can automatically take place when the associated scenarios occur.



(a) Design #1 shows the entire campus and a detailed map is automatically shown when a user gets close to a target. Here the scenario "Bob moves near library" triggers showing a detailed map around the library.

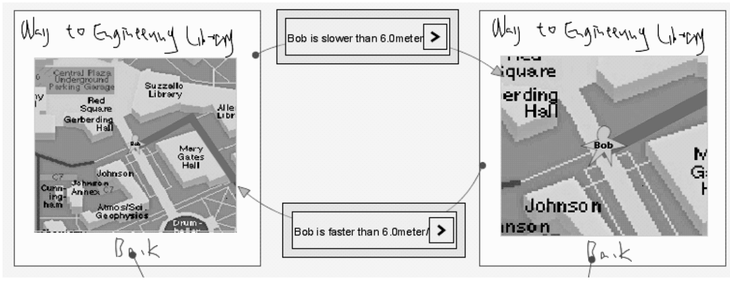


(b) Design #2 shows an area automatically centered on the user and lets the user manually zoom in or out by clicking on the sketched "+" or "–" buttons.



(c) Design #3 uses the user's current location to show different regions of the campus. Here the scenario "Bob enters (or leaves) west campus" triggers showing the west (or east) region of the campus.

continued on following page



(d) Design #4 is similar to Design #2 except it automatically zooms in or out based on the user's current speed. Here the scenario "Bob is slower (or faster) than 0.6 meter/s" triggers showing maps at different zoom levels.

sion lasted about one hour and each segment took about fifteen minutes to complete. During the test, we were able to make some minor changes to the design instantly in response to the participant's suggestions.

All three participants preferred the map centered on the user's current location (#2 and #4). The problem with the first design is that it shows the entire campus on a small PDA screen, which turned out to be hard to read. The third design does show more detail but it does not give a global view of the campus and the participants complained that they could not see the target until they were physically in that region, although they were still able to see the path.

Two participants preferred manual zooming to automatic zooming as they thought manual zooming gave them more control over the zoom levels. However, the other participant thought both kinds of zooming were good to use. All our participants thought the distance label from Design #1 was useful and they also suggested that we should flash the target when users get close to it.

One common problem with the four designs was that there was not enough orientation information provided. We originally thought users could figure out their orientation by referring to nearby buildings and the continuous change of their location on the map.

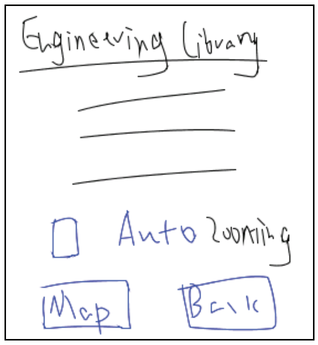
Iteration #2

Based on participant feedback and our observations during Iteration #1, we spent *one hour* creating a new design combining the best features of the four designs (see Figure 8). We added a page for users to choose automatic or manual zooming (see Figure 8a). We explored different ways of showing orientation on a map, including rotating the map, showing an orientation arrow, and showing trajectory arrows (see Figure 8b). These orientation representations are provided by Topiary. In addition, in response to the participants' request, we added the feature of flashing a target when it is nearby. We tested this new design again with three people[1]. Each test session lasted about half an hour in total. In the middle of the test, we turned on the sensor input that is built into Topiary to see how sensor accuracy affected our participants.

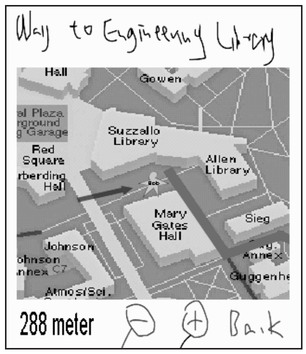
Our participants gave us many useful comments. For example, two of them suggested showing a movement trail to help to indicate orientation. Also, the inaccurate update of the user's location, either by the Wizard or by the sensor input (while it was turned on), did confuse the participants. As a result, one person suggested showing a region for the possible location instead of just a point. They also gave us some other suggestions, such as placing the distance label at the top of the screen rather than at the bottom.

Interestingly, some of our participants did not realize their location was being updated by a wizard rather than by real sensors. It was also observed that the prototype showed an optimal path to a participant who had spent three years on the campus but did not know the existence of this path. We did not know this path either and we simply drew a road network in Topiary by

Figure 8. Two screens (pages) of the new design



(a) A user can select or deselect the checkbox to choose automatic or manual zooming.



(b) A map screen with zooming buttons and a trajectory arrow

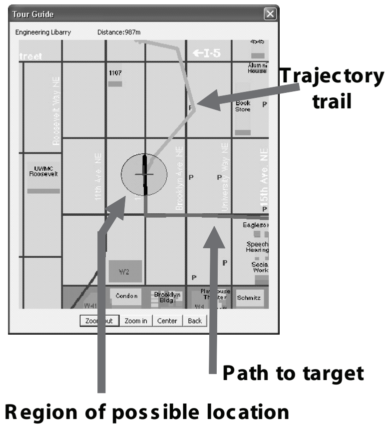
which this path was automatically constructed by the tool.

Building a High Fidelity Prototype

Through these two iterations of informal prototyping and testing, we got a rough view of what the Place Finder should be like. Then it was the time to consider implementation issues and to create a high fidelity prototype. Because we did not want to add an extra device, like GPS, for the Place Finder PDA, we chose to use Place Lab for location sensing, since it requires only WiFi. However, Place Lab, like GPS, cannot provide precise orientation. As a result, we decided to show a movement trail (feedback from the earlier study) instead of showing potentially inaccurate directional arrows or employing map rotation. In addition, because the movement speed cannot be accurately measured, we cut the automatic zooming feature, although one participant showed interest in it. This also helped improve application performance on the PDA.

Based on the earlier tests and an analysis of the implementation issues, we built a high fidelity prototype in Java, using the IBM J9 SWT Java toolkit, in about *two weeks* (see Figure 9). We have used this prototype in the field for hours and it has helped us to find places that we had never been to before. We also got positive feedback from people to whom we demoed this prototype. However, performance on the PDA is still a major issue with this prototype and more profiling is necessary before widely testing it.

Figure 9. The high fidelity prototype was built based on the informal prototypes and an analysis of implementation issues



Lessons Learned

This study offers lessons in two areas. First, it identified usability issues as well as solutions for building map-based navigation techniques in a location-enhanced Place Finder application. Second, the study gives an example of how early stage design iteration can be conducted using informal tools. The study showed obvious advantages over traditional paper prototyping since we were able to test our ideas in the field and leverage those results for later stage development.

Informal prototyping and testing in *hours* was much less expensive than directly building a high fidelity prototype over a period of *weeks* and then testing it with users. The tools allowed us to focus on interaction rather than implementation details. It turned out that little feedback from our participants was related to the informal look of the interface. *Focusing on key interactions* rather than specifying the behaviors of the entire application is important to efficiently conducting early stage design because prototyping tools often employ example-based approaches. In our study, only five places were modeled for testing the five low-fi designs. Once the early usability issues were solved, the design was scaled up to 35 places in the high fidelity prototype.

Carefully testing in the field is important for a successful early stage design because the field is where a mobile application design will be used. Testing in the field requires extra consideration when compared to controlled experiments in a lab setting. The *Wizard of Oz* technique was extremely useful in testing an early stage design since it can reasonably approximate realistic situations. On the other hand, using real sensor input, if not expensive, might help find more usability problems due to the inaccuracy of sensors in a test.

FUTURE TRENDS

Sensors such as accelerometers are becoming available on an increasing number of mobile devices to detect a user's context (e.g., movement, lighting or ambient noise) as well as other peripheral input (e.g., digital compass for the orientation of the device). With these sensors, richer interactions can be constructed. It is important for informal prototyping tools to support interaction design based on the available sensors of the platform. Multimodal interaction that combines multiple interaction modalities has shown promise. Speech-based interaction enhanced by location context is an extremely promising research direction. By leveraging location context, a system can optimize speech recognition by focusing on phrases that have meaning in a particular context. This brings new research opportunities to the rapid prototyping of mobile technology. The two tools that we discussed in this chapter address speech-based interaction and location-enhanced computing separately. It would be interesting to combine the strengths of these types of tools for prototyping location-enhanced speech user interfaces.

CONCLUSION

Informal prototyping tools play an important role in the early stage design of interactive mobile technology. They lower the threshold for entry and reduce the cost for prototyping and testing. As a proof of concept of informal prototyping tools for mobile interaction, we discussed how SUEDE and Topiary address the design of speech-based interaction and location-enhanced interaction, respectively, the two representative types of interaction for mobile technology. We highlight two common features of these tools: graphical storyboarding and Wizard of Oz testing. To show how these tools can help an iterative design process, we reported on our experience in iteratively prototyping a location-enhanced Place Finder application, and testing its prototypes with real users in the field. The study indicated that this type of tool allowed a designer to effectively explore a design space in the early stages of design. As mobile computing becomes more powerful and prevalent, there will be more opportunities for research on informal prototyping tools for the design and evaluation of interactive mobile technology.

REFERENCES

Akers, D. (2006). CINCH: A cooperatively designed marking interface for 3D pathway selection. In *UIST'06* (pp. 33-42).

Apple (1987). *HyperCard User's Guide*. Apple Computer, Inc.

Bailey, B. P., Konstan, J. A., & Carlis, J. V. (2001). DEMAIS: designing multimedia applications with interactive storyboards. In *ACM Multimedia* (pp. 241-250).

BonesInMotion. BiM Active. Retrieved from http://bonesinmotion.com/corp/

Cohen, P. R., & Oviatt, S. L. (1995). The role of voice input for human-machine communication. In *Proceedings of the National Academy of Sciences*, *92*(22), 9921-9927.

Dahlbäck, N., Jönsson, A. & Ahrenberg, L. (1993). Wizard of Oz studies—Why and how. In *Intelligent User Interfaces '93* (pp. 193-200).

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, *28*(3), 300-311.

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, *8*(3), 231-274.

Hartmann, B., Klemmer, S. R., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A., & Gee, J. (2006). Reflective physical prototyping through integrated design, test, and analysis. In *UIST'06* (pp. 299-308).

InfoGation. Odyssey Mobile. from http://www.infogation.com/

Klemmer, S. R., Sinha, A. K., Chen, J., Landay, J. A., Aboobaker, N., & Wang, A. (2000). SUEDE: A wizard of oz prototyping tool for speech user interfaces. In *CHI Letters: 2*(2)*, UIST'00* (pp. 1-10).

LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I. E., Scott, J., Sohn, T., Howard, J., Hughes, J. Potter, F., Tabert, J., Powledge, P, Borriello, G., & Schilit, B. N. (2005). Place lab: Device positioning using radio beacons in the wild. In *Proceedings of Pervasive'05* (pp. 116-133).

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

Li, Y., Hong, J. I., & Landay, J. A. (2004). Topiary: A tool for prototyping location-enhanced applications. In *CHI Letters: 6*(2), *UIST'04* (pp. 217-226).

Lin, J., Newman, M. W., Hong, J.I., & Landay, J. A. (2000). DENIM: Finding a tighter fit between tools and practice for Web site design. In *CHI Letters: 2*(1), *CHI'00* (pp. 510-517).

LOC-AID. LOC-AID People Service. from http://www.loc-aid.net/people_en.htm

MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2004). DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences. In *CHI Letters: 6*(2), *UIST'04* (pp. 197-206).

Macromedia. Director. from http://www.macromedia.com/software/director/

Myers, B., Hudson, S. E. & Pausch, R. (2001). Past, present and future of user interface software tools. In J. M. Carroll (Ed.), *The new millennium*, (pp. 213-233) New York: ACM Press, Addison-Wesley.

Rettig, M. (1994). Prototyping for tiny fingers. *Communications of the ACM*, *37*(4), 21-27.

Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann.

KEY TERMS

Graphical Storyboarding: A technique that informal prototyping tools often employ for designers to describe how an interface should behave. Like a state transition diagram (STD), it has the concepts of states and transitions. However, in graphical storyboarding these states and transitions represent high level UI components or events rather than the computational elements found in a traditional STD.

Informal Prototyping: A type of user interface prototyping used in the early stages of design in which designers explore a design space by focusing on key interaction ideas rather than visual (e.g., color or alignment) or implementation details. These details are often better considered when creating **hi-fidelity** prototypes at a later stage. Paper prototyping is a representative form of informal prototyping in which designers draw interfaces as well as interaction flows on paper.

Informal UI Prototyping Tools: A type of UI prototyping tool that fluidly supports an informal UI prototyping practice. These tools maintain an "informal" look and feel, use fluid input techniques (e.g., sketching) and can automatically generate testable, interactive prototypes.

Location-Enhanced Applications: Computer applications that leverage the location of people, places and things to provide useful services to users. For example, based on the user's current location, show the nearby restaurants or friends. By using the location context, this type of application reduces explicit input required from a user (such as mouse clicks or typing).

Sketch-Based User Interfaces: A type of user interface in which users interact with a computer system by drawing with a pen. The drawings can be recognized and interpreted as commands, parameters or raw digital ink. This type of interface has shown promise in supporting domains such as UI design, mechanical design, architectural design and note-taking.

Speech-Based Interfaces: A type of user interface in which the user input is submitted mainly via speech. A computer system responds based on either recognized words or vocal variation of the speech. The interface output is typically auditory (e.g., when it is on a phone) or visual.

User Interface Prototyping: A practice of creating user interface mockups to test some aspects of a target interactive system.

UI Prototyping Tools: Electronic tools supporting a user interface prototyping process.

Wizard of Oz Testing: A technique for testing an incomplete interface mockup, named after the movie the *Wizard of Oz*. In this technique, a wizard (often played by a designer) fakes an incomplete (or nonexistent) system component to conduct early user evaluations, (e.g., a wizard can simulate speech recognition when testing a speech-based interface or location tracking when testing a location-enhanced application).