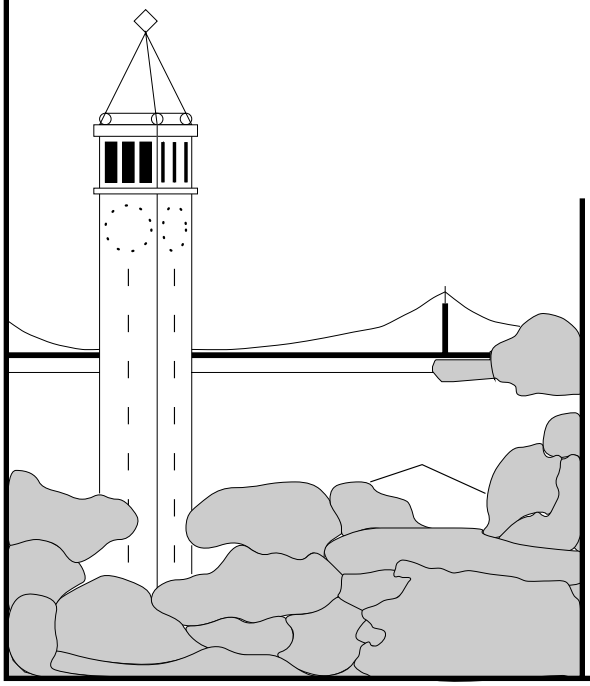
Using Electronic Tools in the Iterative Design of a Context-Aware Tour Guide: A Case Study

Li, Yang, Landay, James A. and Hong, Jason I.



Report No. UCB/CSD-5-1389

May 2005

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Using Electronic Tools in the Iterative Design of a Context-Aware Tour Guide: A Case Study

Yang Li[1] **James A. Landay**[2] **Jason I. Hong**[3]

[1]Computer Science Division
EECS Department
University of California, Berkeley
Berkeley, CA 94720-1776
yangli@cs.berkeley.edu

[2]Intel Research Seattle
1100 NE 45th Street
Suite 600
Seattle, WA 98105
james.a.landay@intel.com

[3]School of Computer Science
HCI Institute
Carnegie Mellon University
Pittsburgh PA 15213-3891
jasonh@cs.cmu.edu

ABSTRACT
We report on the iterative design of a context-aware tour guide. We employed an informal prototyping tool, Topiary, to help us in the early stage design of the system. By using this electronic support to rapidly prototype ideas and test the prototypes with end users in the field, we efficiently explored the usability issues of map-based navigation techniques. Based on the tests, which involved five different designs and two design iterations, and an analysis of implementation issues, we built a high fidelity prototype of the tour guide system. We describe our experience with this iterative design process as well as suggestions for the early stage design of context-aware applications.

Author Keywords
Informal prototyping, field study, location-enhanced, context-aware, Wizard of Oz, ubiquitous computing.

ACM Classification Keywords
H.5.2 [**User Interfaces**]: Prototyping, Evaluation / methodology; D.2.2 [**Design Tools and Techniques**]: User interfaces

INTRODUCTION
Context-aware applications can provide useful services by leveraging the contextual information of people, places and things. Although several studies [1, 2, 5] have been conducted on prototyping and evaluating context-aware systems, little work has been reported on the early stage design of context-aware applications that occurs before any system development has started. Our case study of this design process can provide useful information for ubicomp researchers as well as designers of this type of application.

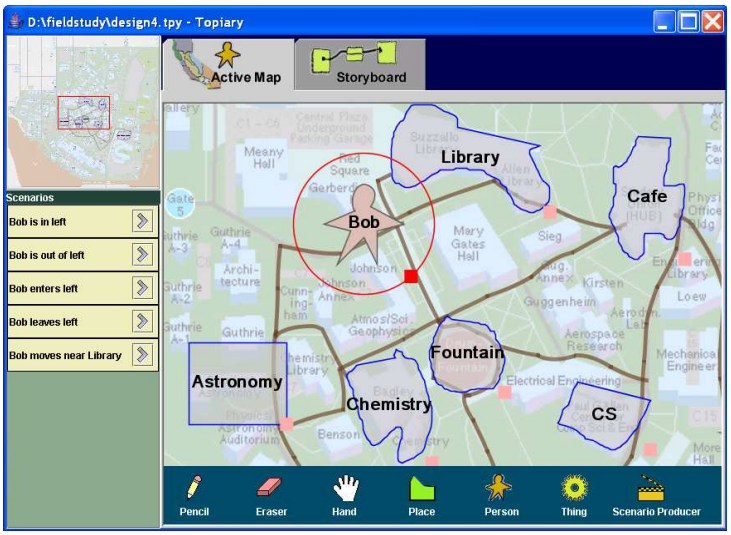


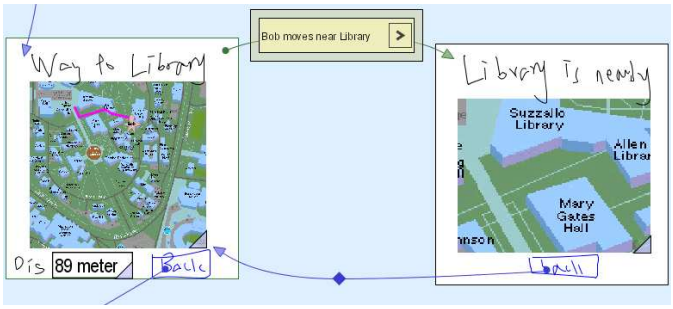
Figure 1. The Active Map workspace of Topiary is used to model location contexts of people, places and things and to demonstrate scenarios describing location contexts.

The early stage design of traditional user interfaces, e.g., GUIs, uses an iterative design methodology that is often conducted with lo-fidelity paper prototyping. This technique has shown great success for quickly prototyping ideas and acquiring early feedback from users. In this paper, we report on a case study in which we iterated on the design of a context-aware tour guide using lo-fidelity prototyping with the support of an electronic prototyping tool, namely Topiary [4].

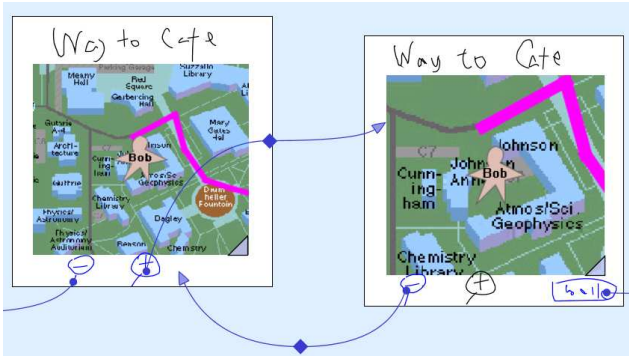
As a first step in examining the early stage design of context-aware applications, we chose a context-aware tour guide as our target application. This type of application can embody many features of context-aware applications. Our iterative design was mainly concerned with map-based navigation techniques on a PDA held by a user walking in the field.

We employed an electronic prototyping tool because it is hard to capture the dynamic factors in context-aware interactions and to conduct realistic tests in the field with traditional paper prototyping. Electronic tools are also helpful in uncovering more usability issues by introducing more interactivity in prototypes.

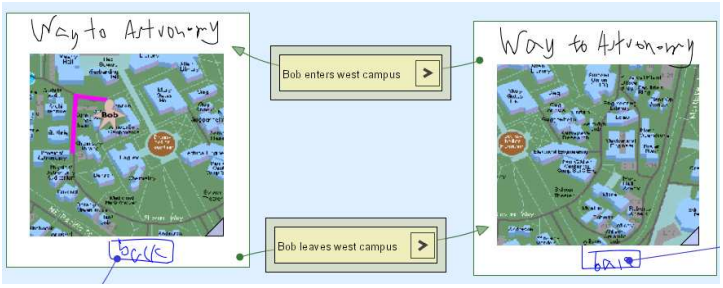
This paper describes our experience with this iterative design process as well as how the results of early stage design benefited building a high fidelity prototype in the following design stage. This experience will give ubicomp researchers and designers first-hand information that will be useful for the early stage design of their own context-aware applications.



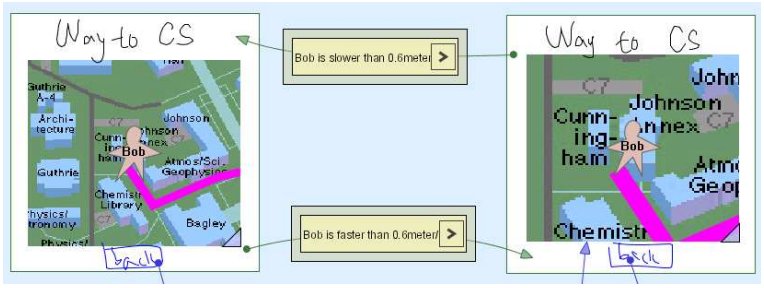
(a) Design #1 shows the entire campus and a detailed map is automatically shown when a user gets close to a target. Here the scenario "Bob moves near library" triggers showing a detailed map around the library.



(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.

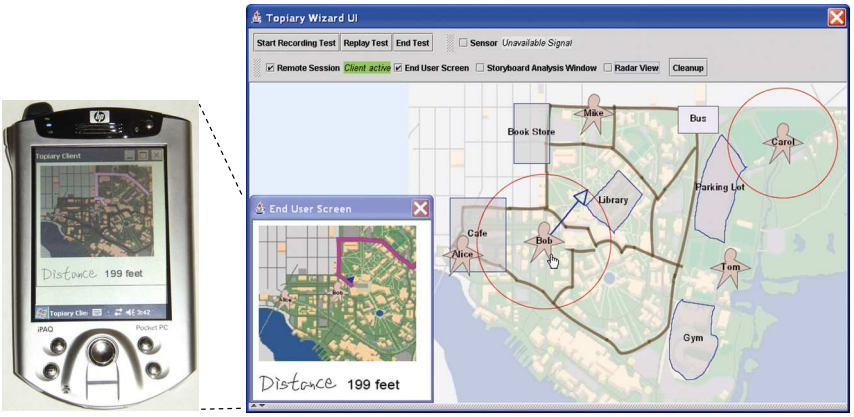


(d) Design #4 is similar to Design #2 except it automatically zooms in or out based on the user's current speed. Here the scenario "Bob is slower (or faster) than 0.6 meter/s" triggers showing maps at different zoom levels.

Figure 2. 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.

THE TOPIARY SYSTEM

Topiary is an electronic tool for rapidly prototyping location-enhanced applications [4]. It allows designers to design based on the high level abstractions of maps, scenarios and storyboards. A designer can model location contexts using an active map (see Figure 1) by creating entities and demonstrating scenarios involving these entities on the map. Then these scenarios can be used as conditions or triggers for page transitions in sketched storyboards (see Figure 2). In addition, a designer can easily test a design using Wizard of Oz techniques, without having to deploy any sensor infrastructure (see Figure 3). Topiary also provides an optional, built-in feature of using real location input from Place Lab [3], requiring only WiFi.



(a) A design can run on a separate device as the End User UI (see the PDA on the left), and a designer can simulate location contexts by moving entities around on the map in the Wizard UI which will trigger page transitions.

(b) An end user can interact with the End User UI (on a PDA) and the wizard (the designer) can follow the user around and update the user's location in the Wizard UI (on a Tablet PC).

Figure 3. Testing with end users using Wizard of Oz techniques.

ITERATIVE DESIGN

We conducted iterative design on the map-based navigation techniques required by the tour guide. The study involved four participants, one female and three males. They were graduate and undergraduate students. Two iterations were performed in this early study and the first iteration included four different designs. 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 (see Figure 3b). In a 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

Design

It took us *three hours* in total to make four prototypes, each using a different navigation technique. The first design shows a map of the entire campus (see Figure 2a). The second design shows an area centered on the user and lets the user manually zoom in and out (see Figure 2b). The third design uses the user's current location to show different regions of the campus (see Figure 2c). The last design is similar to the second, except it automatically zooms in or out based on the user's current speed (see Figure 2d). 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 pink lines in Figure 2) to the target, both of which are updated dynamically by Topiary.

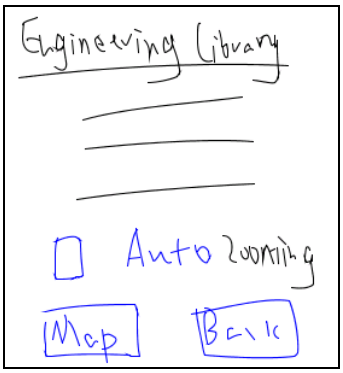
Test

Four segments were included in the test of Iteration #1, each segment for a design. Four different places on a campus were deliberately chosen as targets for the four designs, respectively, based on two principles. First, to smoothly connect the four experimental segments, the target of a segment was the starting point of the next segment. Second, each segment should cover an area that requires appropriate walking distance, not too long or too short, and can produce a path with a certain complexity to avoid simple paths (e.g., the entire path is a straight line.)

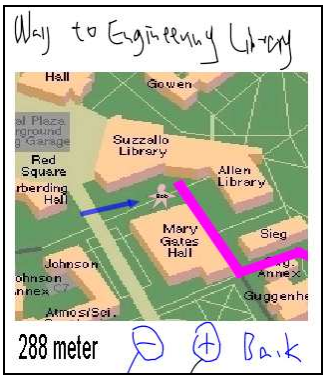
We had three people try all four designs on a PDA in the field, with a wizard updating their location on a Tablet PC. Each experimental session lasted about one hour and each segment took about fifteen minutes. During the test, we were able to make some minor changes to the design instantly in response to the participant's suggestions.

Observations and Feedback

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 by participants. 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.



(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.

Figure 4. Two screens (pages) of the new design.

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 add a flashing effect to the target when users get close to the target.

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

Iteration #2

Design

Based on participant feedback and our observations during Iteration #1, we spent *an hour* creating a new design combining the best features of the four designs (see Figure 4).

We added a page for users to choose automatic or manual zooming (see Figure 4a). We explored different ways for showing orientation on a map, including rotating the map, showing an orientation arrow, and showing trajectory arrows (see Figure 4b). These orientation representations are provided by Topiary. In addition, in response to the participants' request, we added a feature to flash a target when it is nearby.

Test

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.

Observations and Feedback

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 which this path was automatically constructed by the tool.

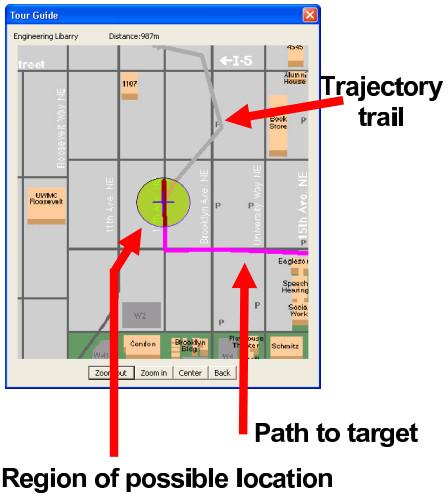


Figure 5: The high fidelity prototype was built based on the informal prototypes and an analysis of implementation issues.

BUILDING HIGH FIDELITY PROTOTYPES

Through these two iterations of informal prototyping and testing, we got a rough view of what the tour guide application should be like. Then it was the time to consider implementation issues and to create high fidelity prototypes.

Because we did not want to add an extra device, like GPS, for the tour guide PDA, we chose to use Place Lab for location sensing, since it requires only WiFi. However, Place Lab cannot provide precise orientation. As a result, we

[1] Note: two of these three participants also participated in the first iteration. Our main purpose here was to uncover usability issues by getting feedback on these lo-fidelity prototypes at the early stages of design rather than to conduct strict experiments.

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 early 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 5).

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.

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 context-aware tour guide. Other ubicomp developers can use our analysis and apply these designs, e.g., manual zooming and movement trails, in other ubicomp applications.

Second, the study gives an example on how early stage design iteration can be conducted using electronic tools. The study showed obvious advantages over traditional paper prototyping, and how the results can be leveraged 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 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 a design in the early stages, 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.

CONCLUSIONS

We reported on the iterative design of a context-aware tour guide. We employed an informal prototyping tool, Topiary, to help us in the early stage design of the system. By rapidly prototyping ideas and testing the prototypes with end users in the field, we efficiently explored the usability issues of map-based navigation techniques. Based on the tests and an analysis of implementation issues, we built a high fidelity prototype of the tour guide system. We contribute to the field our experience with this iterative design process as well as lessons for designing a context-aware tour guide and for conducting efficient early stage design of context-aware applications.

REFERENCES

1. Burrell, J., et al. *Context-Aware Computing: A Test Case*. In *Ubicomp* 2002. Göteborg, Sweden: Springer-Verlag London, UK: pp. 1-15.
2. Cheverst, K., et al., *Developing a context-aware electronic tourist guide: some issues and experiences. CHI Letters*, 2000. 2(1): pp. 17-24.
3. LaMarca, A., et al. *Place Lab: Device Positioning Using Radio Beacons in the Wild.* In *To appear in Proceedings of Pervasive '05*: pp.
4. Li, Y., Hong, J.I., and Landay, J.A., *Topiary: A Tool for Prototyping Location-Enhanced Applications. UIST 2004, CHI Letters*, 2004. 6(2): pp. 217-226.
5. Long, S., et al. *Rapid prototyping of mobile context-aware applications: the Cyberguide case study*. In *Proceedings of the 2nd annual international conference on Mobile computing and networking* 1996. Rye, New York: pp. 97-107.