BrickRoad: A Light-Weight Tool for Spontaneous Design of Location-Enhanced Applications

Alan L. Liu & Yang Li

DUB Group
Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
{aliu, yangli}@cs.washington.edu

ABSTRACT

It is difficult to design and test location-enhanced applications. A large part of this difficulty is due to the added complexity of supporting location. Wizard of Oz (WOz) has become an effective technique for the early stage design of location-enhanced applications because it allows designers to test an application prototype by simulating nonexistent components such as location sensing. However, existing WOz tools 1) require nontrivial effort from designers to specify how a prototype should behave before it can be tested with end users, and 2) support only limited control over application behavior during a test. BrickRoad is a WOz tool for spontaneous design of location-enhanced applications. It lowers the threshold to acquiring user feedback and exploring a design space. With BrickRoad, a designer does not need to specify any interaction logic and can experiment on-the-fly with different designs during testing. BrickRoad is a valuable complement to existing tool support for the early stage design of location-enhanced applications.

Author Keywords

Location-enhanced computing, prototyping, ubiquitous computing, Wizard of Oz

ACM Classification Keywords

H.5.2 **[User Interfaces]**: Prototyping; D.2.2 **[Design Tools and Techniques]**: User interfaces

INTRODUCTION

Location-enhanced applications can provide many useful services based on the location of people, places, and things [3]. However, it is difficult to design and test location-enhanced applications mainly due to the added dimension of location, which increases the complexity of a design. In particular, it is challenging in the early stages of design, where the goal is to quickly get user feedback and explore a design space via experimentation.

Wizard of Oz (WOz) is a valuable approach in the early stages of UI design, as it reduces the cost for acquiring user feedback by allowing a wizard (often played by a designer[1]) to test a design with simulated (nonexistent) application behavior [1], e.g., speech recognition. Recently, WOz approaches have been applied when testing location-enhanced applications [2, 4, 6−8], by allowing a wizard to simulate location tracking that would otherwise be hard to incorporate. In particular, WOz-based prototyping tools have been developed for streamlining the design process of location-enhanced applications [4, 7]. They allow designers to test application prototypes in realistic situations, e.g., in the field, to get ecologically valid user feedback. However, these tools require nontrivial effort from designers to specify how a prototype should behave before it can be tested with end users, and support only limited control over application behavior during a test, which can hamper designers' ability to test design inspirations on-the-fly.

We created BrickRoad (see Figure 1), a WOz tool for spontaneous design of location-enhanced applications. BrickRoad addresses three major design goals:

1. Provide a low threshold to conducting tests;

2. Provide flexibility to support spontaneous experiment-ation with design alternatives;

3. Enable design iteration.

These goals resulted from our experience designing an indoor wayfinding application for individuals with cognitive impairments [6]. In the very early stage of that project, we mainly focused on eliciting use scenarios and conceptual interaction models of users. We wanted a wizard to adapt to each user's pace and level of confusion with different types of directions during a test, as such on-the-fly adaptation can inform the design of application logic.

---

[1] As such, we use the terms of "wizard" and "designer" interchangeably in the paper.
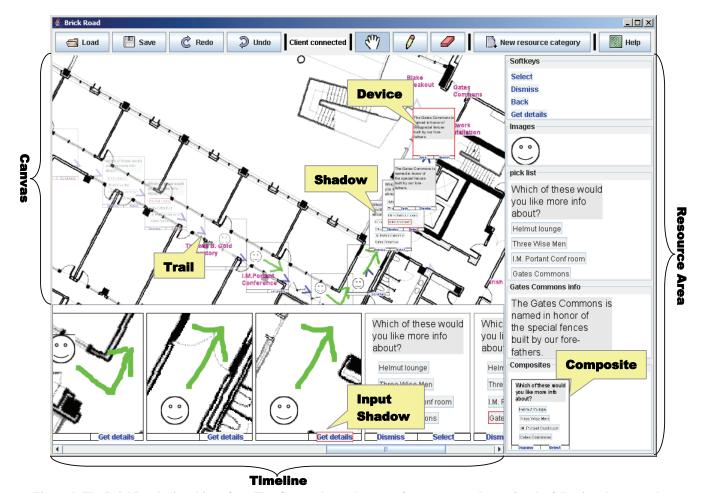
Figure 1. The BrickRoad wizard interface. The *Canvas* shows the map of a test area and contains the following elements: the *Device* shows the current end-user's UI and its position on the map represents the user's current location; the *Trails* record user's previous locations; and the *Shadows* record previous end-user UIs as well as the locations when end-user UIs are updated. The Resource Area stores materials that can be sent to the end-user UI. Multiple materials can be combined into a *Composite*. The *Timeline* shows past end-user UI states in a chronological order.

BrickRoad is aimed at supporting such stages of a design process, which previous tools are not designed for [4, 7]. It allows designers to test various design alternatives on-the-fly, without the need to specify any interaction logic. BrickRoad accomplishes this by allowing a wizard to control the presentation of images, audio, text, and graphical widgets on the user's device, as well as simulate both location sensing and application logic.

AN EXAMPLE BRICKROAD APPLICATION
To describe how a designer can use BrickRoad to design location-enhanced applications, we will use the following example mobile tour guide application running on a WiFi-enabled Windows Mobile Smartphone (e.g., the T-Mobile SDA). The application allows a user to get directions to and information on nearby places of interest using the phone's joystick and softkeys. It displays a map showing the user's current location and updates when the user moves. The user can bring up a list of the nearby places of interest with a softkey. Picking a place brings up the map with a path from the user's current location to the selected place of interest.

The user can then bring up details about the place of interest or return to the map screen.

Preparing Test Materials
The first step in creating the tour guide prototype with BrickRoad is to prepare all the materials that will be used by the prototype, such as images, audio clips, and text. Examples of such materials are an image of a labeled floorplan to serve as the map for the tour guide, an image to represent the person in relation to the map, various photographs of the places of interest, and audible prompts.

All these materials are imported as design resources in the *Resource Area* (see the right of Figure 1). These materials can be organized by *category*. For example, all the appropriate softkey labels are added to the "Softkeys" category. The designer also imports a map image into the *Canvas* (see Figure 1). The position of the *Device* on the map represents the user's current location. The designer can choose to display the map area enclosed by the *Device* on the phone. All media files (images and audio clips) are stored on the user's phone for quick access during a test.

Conducting a Test

During a test, a wizard walks behind a user and maintains the user's location on the map by crossing [5] the *Device* over the map with a tablet pen. The wizard can drag or cross items listed in the *Resource Area* to put them on the *Device*, where they can be arranged and sent to the phone.

Semantic Zooming

BrickRoad has two zoom levels that let a wizard switch between authoring content and simulating location. Tapping on the *Device* zooms in to the *content authoring mode*, in which a wizard can send *Device* content to the phone by clicking on the Send button. Tapping outside the *Device* zooms out to the *location simulation mode*. Both actions take place by interacting with the *Device* because the state of the phone is a wizard's center of attention.

Shadows

Shadows were inspired by our need to create a log replayer after our studies with the wayfinding application prototype [6]. We used the replayer to determine how often certain messages had to be sent, how location and time factored in study participants' responses, and which types of messages seemed more successful. BrickRoad combines the essential capabilities of our custom prototype and the replayer.

Sending the *Device* to the phone causes a *Shadow* to appear at the location of the Device and a clone of the *Shadow* to appear at the end of the *Timeline* (see the bottom of Figure 1). *Shadows* are snapshots of previous *Device* contents. In addition to *Trails* (i.e., trajectory arrows) placed when the wizard updates a user's location, *Shadows* record the progression of interaction between the user and the application during a test.

Shadows fade over time to reduce the screen clutter, but can be retrieved when a wizard taps on one. When tapped, the *Shadow* on the Canvas becomes opaque and the *Timeline* scrolls to show the associated *Shadow*. A wizard can tap on a *Shadow* in the *Timeline* to send its contents to the *Device*. This allows reusing content based on both temporal and spatial locality.

End-User Input

When a user presses a softkey, BrickRoad automatically shows a busy cursor on the phone until new content is sent. In the BrickRoad wizard interface, the pressed button and any selected checkbox or list items flash on the *Device*. *Input Shadows* are created in the *Canvas* and *Timeline* with the associated screen elements highlighted (see Figure 1). In the example application, the wizard zooms into the *Device*, toggles the map display off, drags the "Pick a place" text, list items of each of the nearby places of interest, and "Select" and "Dismiss" labels to the *Device*, and then sends the content to the phone.

When the user selects a place, the wizard clears the *Device*, toggles the map back on, draws a path from the center of the screen to the selected place, and drags the "Details" and "Back" labels to the softkey area. When the user presses the "Details" softkey, the wizard clears the screen, toggles the map off, drags the photo and text description of the place to the *Device*, and "Dismiss" and "Back" button labels to the softkey area.

Iterating on Design

As one way to solidify test findings for following design iterations, BrickRoad allows designers to combine multiple materials into a *Composite* that can then be manipulated as a whole. A *Composite* provides a mechanism to consistently lay out screens and reduce the need to repeatedly assemble associated materials during tests. A designer can create a *Composite* by assembling materials from the *Resource Area* and from *Shadows*.

BrickRoad is also able to capture and export test data, including wizard and user interactions (location changes, user input and application output), to XML formatted documents for subsequent tests or analyses.

DISCUSSION AND FUTURE WORK

Based on our as well as others' experiences, there is an interesting tradeoff between design and wizard cost while applying WOz approaches. In this work, we intentionally took a different direction from previous prototyping tools and explored the end that requires a higher wizard cost but a lower design cost. The wizard load in BrickRoad is higher than that of previous tools. However, BrickRoad imposes lower load for preparing a test, by avoiding the need to specify interaction logic. So far, BrickRoad is able to investigate interesting applications such as indoor navigation. With BrickRoad, we intend to find out how far this type of "spontaneous design at test" can go. This can provide useful evidence for future exploration of design tools for Ubicomp.

Addressing the Three Design Goals

Here we discuss how the design of BrickRoad supports the three goals that we discussed in the introduction.

A Low Threshold for Conducting a Test

BrickRoad enables a low barrier of entry to testing interactive behavior. It does not require any interaction logic to be specified, which can otherwise be complex to design and difficult to implement for location-enhanced applications. To start a test, a designer needs only to load all the materials on the phone and the wizard interface. During a test, all the materials can be sent to the user's interface, e.g., a mobile phone, at any time with only a few, simple steps. All interaction involves a series of pen tapping, dragging, or crossing. As it is the wizard's responsibility to respond the user's input and update locations, the wizard's task load can be high during a test. BrickRoad offloads some of the wizard's work by providing a set of built-in interactive widgets for handling basic input automatically, such as checkboxes, menu buttons, selectable list items and a dynamic map.

High Flexibility for Experimenting with Alternative Designs
BrickRoad allows a wizard to spontaneously respond to the user input as well as location changes, by assembling the contents to be sent to the user on-the-fly. When a user input event takes place, a busy cursor on the end-user's device appears to give the user the impression that the system is processing. A wizard has the chance to handle all the application logic by assembling the materials in the *Resource Area* as well as reusing the content in the *Timeline*. The association between the *Shadows* in the *Canvas* and *Timeline* allows a wizard to quickly retrieve and reuse temporally and spatially adjacent content. In addition, freeform sketching is supported for handling tasks that are complex to do automatically but easy to simulate manually. For example, a wizard can simply draw the shortest-path route or a turning arrow on the *Device* and then send the drawings to the end-user UI.

Design Iteration Support
A designer can analyze the *Shadows* to find interaction patterns, which in turn can be used to refine design ideas. *Shadows* on both the *Canvas* and the *Timeline* record previous interactions to give a wizard the sense of the progression of interaction and also a basis for later analysis. A designer can solidify their findings into *Composites* that help increase UI consistency across similar interactions and streamline interactions for later test iterations. We plan to incorporate more powerful analysis techniques in the future.

Transitions to Rapid Prototyping Tools
BrickRoad is aimed at earlier design stages that previous tools are not designed for. In such stages, a designer often has a vague idea about users' needs as well as a target design space, and tends to explore various design inspirations based on users' quick feedback. However, as a design idea matures and becomes more structured, the design environment should be shifted from BrickRoad to tools allow more automation (e.g., [4, 7]). Each test session of BrickRoad can produce a large amount of interaction examples. These interaction examples include rich information about locations of interest and interaction patterns that can be used to infer application logic, which can then be explicitly modeled and represented by existing prototyping tools, (e.g., using storyboards or rules). We plan to enable the transition of BrickRoad prototypes to such rapid prototyping tools.

Informal Feedback and Future Evaluations
We are currently using BrickRoad in our location-based projects and distributing the tool to university students in an HCI course. We have garnered positive feedback on BrickRoad from researchers and students in our group. The BrickRoad client is available on Windows Smartphones. We have tested BrickRoad (using the T-Mobile SDA with a wireless connection) in an indoor environment. We plan to conduct formal user studies on BrickRoad to learn how effective the UI is for a wizard to use during an actual study, and what additional features would be useful to make testing easier and analysis more useful.

CONCLUSION
We created BrickRoad, a WOz tool for spontaneously designing location-enhanced applications by allowing a wizard to simulate both location tracking and application logic. It lowers the threshold to acquire early user feedback by not requiring that interaction logic be specified before a test and by allowing lightweight interactions (e.g., crossing) during a test. It provides flexibility to support design space exploration by allowing a wizard to assemble user interfaces on-the-fly. It supports design iteration by allowing a designer to solidify test findings into reusable composites. We have used BrickRoad to design several location-enhanced applications and believe it is a valuable complement to existing tool support for the early stage design of location-enhanced applications.

REFERENCES
1. Dahlbäck, N., Jönsson, A., and Ahrenberg, L. Wizard of Oz studies: why and how. *Proc. IUI 1993*. ACM Press (1993), 193-200.

2. Dow, S., MacIntyre, B., Lee, J., Oezbek, C., Bolter, J.D., and Gandy, M. Wizard of oz support throughout an iterative design process. *IEEE Pervasive Computing* 4, 4 (2005), 18-26.

3. ekahau. Increased visibility and monitoring of critical assets and people. http://www.ekahau.com/.

4. Li, Y., Hong, J. I., and Landay, J. A. 2004. Topiary: a tool for prototyping location-enhanced applications. In *Proc. UIST 2004*. ACM Press (2004), 217-226.

5. Li, Y., Welbourne, E., and Landay, J.A. 2006. Design and experimental analysis of continuous location tracking techniques for Wizard of Oz testing. *Proc. CHI 2006*, ACM Press (2006), 1019-1022.

6. Liu, A.L., Hile, H., Kautz, H., Borriello, G., Brown, P., Harniss, M., and Johnson, K. Indoor wayfinding: developing a functional interface for individuals with cognitive impairments. *Proc. ASSETS 2006*. ACM Press (2006), 95-102.

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

8. Reilly, D., Dearman, D., Welsman-Dinelle, M., Inkpen, K.M. (2005). Evaluating early prototypes in context: trade-offs, challenges, and successes. *IEEE Pervasive Computing* 4, 4 (2005), 0-18.