Wizard of Oz Interfaces for Mixed Reality Applications

Steven Dow[1], Jaemin Lee[1], Christopher Oezbek[1], Blair MacIntyre[1], Jay David Bolter[2], Maribeth Gandy[3]
College of Computing[1], School of Literature, Communication and Culture[2],
Interactive Media Technology Center[3]
GVU Center, Georgia Institute of Technology
Atlanta, GA 30332, USA
{steven, jaemn, blair}@cc.gatech.edu, oezbek@inf.fu-berlin.de, jay.bolter@lcc.gatech.edu,
maribeth@imtc.gatech.edu

ABSTRACT
One important tool for developing complex interactive applications is "Wizard of Oz" (WOz) simulation. WOz simulation allows design concepts, content and partially completed applications to be tested on users without the need to first create a completely working system. In this paper we discuss the integration of wizard interface tools into a Mixed Reality (MR) design environment and show how easier creation and evolution of wizard interfaces can lead to an expanded role for WOz-based testing during the design evolution of MR experiences. We share our experiences designing an audio experience in an historic site, and illustrate the evolution of the wizard interfaces alongside the user experience.

KEYWORDS
Wizard of Oz, design tools, authoring, prototyping, mixed reality, ubiquitous computing, experience design

ACM CLASSIFICATION KEYWORDS
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces - *User-centered Design, Prototyping*

INTRODUCTION
Designing and prototyping interactive applications for physical spaces, such as ubiquitous computing and mixed reality (MR), can be difficult, time-consuming and costly because of the difficulty of building and deploying the sensing technologies and complex application logic needed to create these mixed physical-virtual systems. Often designers who develop computer interfaces in physical environments do not have working sensors during the prototyping stages of design; indeed, during these early stages, it may not even be clear what sensors are needed. This difficulty is compounded for media-rich experiences, such as tours of historic sites, since compelling content is expensive and time-consuming to create. Under these conditions, adopting a human-centered approach to understand user behavior and test prototypes can be challenging. The *Wizard of Oz* (WOz) simulation method, where some (or all) application and sensor functionality is simulated by a wizard behind-the-scenes, is a very common and practical tool for testing MR system prototypes.

The WOz method has been explored successfully in many computing contexts. The method has been used to simulate and test the speech and direct manipulation interface for Turvy, an intelligent agent that learns based on user interaction [6]. In SUEDE, WOz is used to simulate realistic system behaviors during the design and evaluation of speech-based interfaces [3]. By simulating perfect voice recognition with the wizard, SUEDE can insert errors to simulate the performance of the expected technology and enable statistically meaningful user evaluation. The Topiary project uses WOz to simulate location sensing during the design of location-enhanced applications [4]. In general, previous uses of WOz for application design have been engineered for one specific task. With Topiary, for example, the wizard's role is to simulate the position of the user, forcing the designer to work out complex application logic. While this approach is very useful for testing specific designs that can be laid out as state-diagrams, it is of limited use during early design stages when the content and behavior of the system is undecided, or when there are many interacting states or complex application logic. Finally, WOz has be shown to be useful in deployed experiences. In Desert Rain, for example, dedicated wizards were present during a mixed reality performance to facilitate the experience for users [1].

As these examples illustrate, WOz simulation can play a significant role throughout the prototyping, testing, development, and deployment of an application. Unfortunately, WOz interfaces can be hard to create, depending on the capabilities of the programming environment used for the application, and the architecture of the application itself. We believe that if support for rapid creation of WOz interface tools is integrated into the application prototyping environment, the WOz method would be utilized more often, leading to more frequent testing of design ideas and, hopefully, to better end-user experiences. Ideally, for each design prototype or concept, different wizard interfaces would be created to support focused evaluation. Therefore, in our work, we have integrated flexible tools to support WOz interfaces directly

CHI 2005, April 2–7, 2005, Portland, Oregon, USA.
ACM 1-59593-002-7/05/0004.

into a design environment for prototyping mixed reality experiences. The case study presented below details the development of an MR experience prototype, and illustrates the WOz-based evaluation strategies that were used.

WIZARD OF OZ INTERFACE TOOLS

At a high level, the design process for computer applications situated in the physical world is similar to desktop or web development: understand users and tasks, brainstorm ideas, prototype initial concepts, test user scenarios, and develop systems. Clearly this is not a straightforward process; these phases overlap and iterate many times before a final system design is reached.

In human-centered design work, it is necessary to evaluate throughout the design process, and in MR environments, a range of WOz techniques can help answer a variety of questions throughout the design process. WOz interfaces must be flexible, simple to prototype, and usable by the dedicated wizard operator. The usability of the wizard interface (e.g., What does the human wizard operator see and hear? What is their cognitive load?) is important if meaningful evaluations of the user experience are to be performed. The wizard operator must be capable of performing their assigned tasks, usually some combination of observing the user, observing the environment, simulating sensors and/or controlling content. Throughout the design of the experience, the wizard interface should be redesigned to support the user testing needed at the time. Similar to SUEDE, we have included tools to facilitate the rapid construction of wizard interfaces, but we attempt to support a wider range of interfaces.

Our human-centered design of a location-based audio experience ("The Voices of Oakland"), described in the case study section, provides insight to the importance of flexible WOz support during the evolution of applications in physical spaces. Both the end-user audio tour and the wizard interface used to facilitate the development of the tour were designed in DART, the Designer's Augmented Reality Toolkit (see [5] for a detailed discussion of DART). Motivated by the lack of design tools for media-rich applications in physical environments, we created DART to empower media designers to build experiences for mixed physical/virtual worlds within the familiar design tool of Macromedia Director.

The key to WOz simulation in DART is its use of a simple event broadcast/subscription model (*cues* and *actions*) to communicate between entities in an application. A *cue* is an event representing a high-level user interaction or internal state change that an entity in DART may express interest in (subscribe to). For example, DART has cues that are linked to the passage of time, user position, sensor values, and mouse and keyboard interaction. An *action* is a response performed by an entity when it receives a cue. For example, DART has actions for starting/stopping content and for manipulating the 3D world. There may be multiple cues that trigger one action or multiple actions subscribed to one cue, and the cues/actions are easily substituted and iterated during the design process.

Only small changes are needed to prepare the user application for wizard control in DART (primarily the inclusion of a so-called "puppet" script to enable remote wizard control). Among other things, this script allows cues (i.e., events) to be transparently forwarded between the wizard and the controlled "puppet" application, enabling the WOz application to control the user application using its existing cues and actions. The WOz interfaces are built using the same tools and techniques as the user applications, allowing the WOz application to monitor and leverage the same sensors and shared data. Application logic for the user is programmed using DART's existing suite of events (cues/actions) and/or by creating custom events with Lingo (Director's programming language).

DART leverages the broadcast/subscription model and transparent distributed communication[1] to automatically generate simple WOz interfaces, if desired. The puppet script monitors the running user application, notifying the wizard when actions become subscribed to cues. A wizard interface (with buttons labeled with the cue name) can be created on the fly, and automatically modified as subscriptions change. When a button is pressed, the cues are broadcast on the user application, causing the subscribed actions to occur. Cues created by the WOZ application and those triggered by the application logic are identical in the DART system, allowing the wizard to work with partially (or even completely) finished applications. Designers may also create custom wizard interfaces using existing design features in Director, integrating the automatically-generated button set if they desire. WOz cues can be triggered by any user-designed interface element, not just the automatically generated buttons.

Synthetic trackers (and other sensors) can be added to the wizard interface to simulate real trackers (the Phidgets project demonstrated the utility of simulated sensors in a design environment [2]). These sensors are indistinguishable from the real sensors to the user application, allowing for a smooth transition to working sensors. As with cues, simulated sensor data can be used in parallel to real data, even allowing noisy sensors to be monitored and adjusted by the wizard.

A user experience can also be controlled by multiple wizard interfaces, allowing two or more wizard operators to observe the environment. The flexibility of DART is important for rapidly creating the appropriate wizard interface at different levels of the design.

CASE STUDY: THE VOICES OF OAKLAND

To demonstrate the power of flexible wizard interfaces in the design of an experience for a physical environment we will discuss "The Voices of Oakland" project, an audio-based tour based on historic Oakland Cemetery in Atlanta. In the audio experience, we are attempting to enhance the environment with dramatic stories from the lives of cemetery residents, so that visitors can better understand the history of Atlanta and

[1] The Virtual Reality Peripheral Network (http://www.cs.unc.edu/Research/vrpn/) is integrated into Director by the DART Xtra, and used for distributed communication and sensor integration.

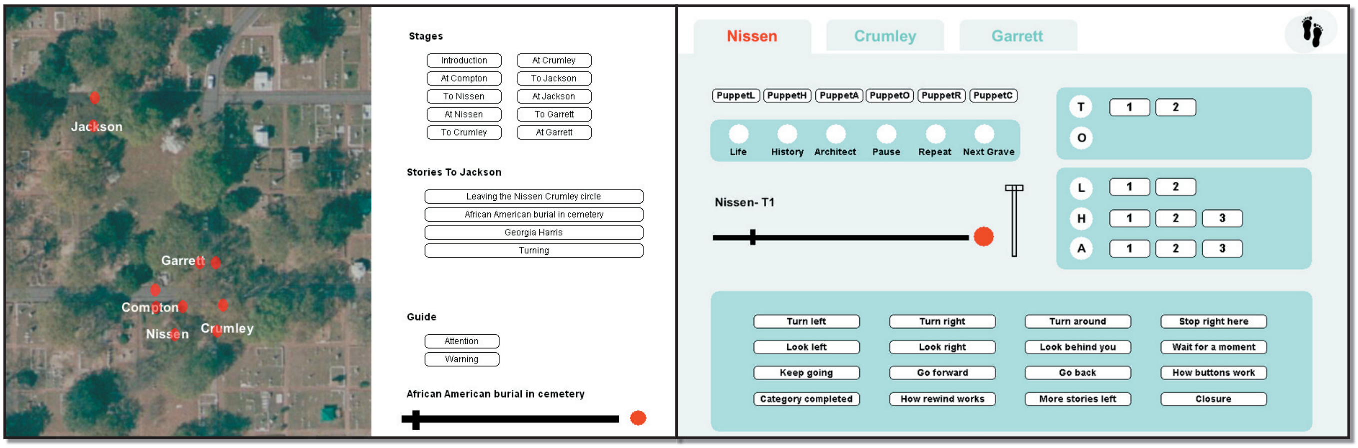

Figure 1: a) The first iteration of the WOz interface for "The Voices of Oakland" with a choice of button interaction or map-based simulation of GPS technology. b) The second iteration of the WOz interface, custom built to match the organization of the stories. In the third iteration, the WOz interface on the right was used primarily to monitor the user's interaction with button controls.

the South. We are particularly interested in exploring a blend of linear and non-linear storytelling as participants wander through this historic place. We envision the final application using GPS location sensing in conjunction with other sensors, and to have sophisticated application logic to create unique and engaging experiences for visitors.

First iteration

For our first generation of "The Voices of Oakland" project, our writers prepared a dialectic dramatic script. We quickly recorded the script using our own voices for the characters and imported the audio clips into DART, grouping them into scenes around each location. Each audio actor subscribed to a unique cue that would trigger an action causing the sound to play, and a collection of cues were used to control the movement from one scene to another. Initially we created application logic to determine when to play audio based on a user's position, but working with the GPS devices proved to be difficult due to their poor accuracy. Therefore, we implemented a map-based WOz interface with an overhead map of the cemetery and red circles representing the location of audio segments (Figure 1a). The wizard could manipulate an icon of the user on the map, moving them near zones of audio, and thus triggering the audio to play on the participant's machine.

The wizard interface also included a set of automatically-generated graphical buttons, as described in the previous section. The combination of map and buttons let the wizard choose to either simulate the GPS tracking or push buttons to directly trigger the media content. Finally, the audio segments and application logic for the experience were replicated on the wizard interface so that the wizard operator could hear exactly what the user heard.

We ran a short pilot study using the two script writers as wizards, since they were experts on the content. On the wizard side, we observed that our operators never used the map portion of the interface to "position" the user into the audio zones. Our wizard operators claimed that using the buttons, as opposed to the map, put them closer to the content and were much easier to understand and control. For example, adjustments could be made on the fly to the ordering and timing of the content. At this early stage in the design, the designers were not concerned with tracking the user to trigger the audio correctly. The participants enjoyed the experience, but wanted more control over the amount and kind of content at each location. They also reported not being aware that a wizard was controlling the content.

Second iteration

After the first design iteration and pilot testing, several changes were made to the experience. The script was divided into smaller dialogues and put into categories (the resident's Life, relevant History, nearby Architecture). This way, the user could listen to small segments of stories within a category and then dig deeper, or choose a different category. To simulate user-control, we supplied the user with a hand-held controller with physical buttons that corresponded to the categories of audio that could accompany any particular grave site. We recorded the audio with professional actors so that the content sounded rich and authentic.

The wizard was redesigned for this iteration of the prototype (Figure 1b). We removed the map portion of the wizard interface and the location-based content activation, because we decided it was too early for this type of interaction (the map interface will be useful again after the application logic is programmed and we simply want to test the experience with simulated high-accuracy GPS). The wizard interface was custom designed to match the organization of the audio stories. The user's button presses did not trigger the audio, but were routed to the remote machine and appeared on the wizard interface (near the top of Figure 1b). The wizard still controlled what audio was played, but made their decisions largely based on which buttons the user pressed. We chose to route the button interaction through the wizard because we had questions about the level of control the user should have and what users would expect from the button device.

We tested this experience informally during an outdoor Fall Festival in the cemetery. We had about fifteen different guests (members of the public) try the experience, and four different wizards operate the tour over a period of several hours. In general, the wizards simply relayed user button presses, indicating to us that users would be fine using the button interface to control the audio content at a particular grave. We found that the users enjoyed the experience and liked having the button control over the audio. Some users talked about their desire to be able to freely explore the cemetery on their own, rather than always being guided by the characters.

Third iteration

In the third version of "The Voices of Oakland", we lowered the cognitive load of the wizard by allowing the user's buttons presses to trigger the changes in audio directly, instead of routing the commands through the wizard. The wizard tells the system when the user approaches a new grave, and triggers appropriate directional audio segments to help the user navigate from one grave to another if they appear lost. While at a particular grave, the user is in control of the audio content, but the wizard can override system if necessary. The appearance of the wizard interface remained essentially unchanged, and served to give the wizard feedback on the state of the system.

We have begun a formal user study and asked the first four participants to take the guided tour and to think aloud during the experience to express ideas or concerns they had along the way. Most of the participants talked about wanting to hear about nearby graves and other points of interest. The wizard interface would be ideal for expanding content and testing out non-linear story elements. Again, we would not have to program the application logic, and the wizard could choose from a range of audio clips when the user approaches (or looks interested in) particular graves.

In this version, we have added sensors to the system (GPS and head-orientation tracking) to obtain as much information as possible about the user's position and head orientation. During each trial, we are capturing the data from these sensors, synchronized with a log of the user and wizard interactions, to better understand how to create the complete application logic, and to understand when the content is not working (such as when the user seems not to understand the navigational instructions given by the narrator).

DISCUSSION

In "The Voices of Oakland" project, we have used different WOz strategies. Early on, the wizard was required to monitor the environment, judge the user context, and provide the best media content for the situation. If there was too much unstructured content to choose from, this strategy could breakdown. Similarly, simulating multiple simultaneous high-precision trackers (such as user location and orientation) would very difficult for a wizard to control, and prevented us from exploring the use of spatialized audio content in this experience. Based on user feedback, we provided the user with buttons to trigger and control some of the audio content, while letting the wizard advance the experience between graves and help when users get lost.

Our hope is to create an effective user experience through extensive user testing. Once we have a collection of compelling content and a target experience, the logic for the application will be reverse engineered, with the help of the sensor data and button interaction captured from successful wizard-controlled experiences. DART will allow us to test the application logic by playing back the captured data from the user tests into the experience, or by simulating the sensors using different derivatives of the wizard interface (such as the map to simulate user location). As the sensing technology becomes available (e.g., more accurate GPS), the wizard's role will continue to shift. The wizard could disappear or continue to be used during user testing to react to unimplemented system input such as hand gestures, or even the emotional (affective) state of the user. In each case, the wizard interface can be redesigned in a specific way to optimize the wizard's ability to observe the state of the experience and control the content or system state in some way.

Wizard of Oz simulation has great potential for facilitating the many facets of design, especially for applications in physical environments. The cognitive and motor ability of one or more wizard operators can be exploited to simulate many sensors and/or intelligent computer logic. By integrating the design tools for the wizard interface directly into the design environment for the user experience, designers can quickly and effectively take advantage of WOz testing and develop new strategies for iteratively solving design problems.

ACKNOWLEDGMENTS

We extend a special thanks to Danny Muller, Susan Bryant, Kevin Kuharic, and John Avery. This work was supported by NSF CAREER Grant 0347712.

REFERENCES

1. Koleva, B., Adams, M., Taylor, I., Benford, S., Fraser, M., Greenhalgh, C., Schnädelbach, H., vom Lehn, D., Heath C., Row-Farr, J. (Nottingham and Blast Theory) "Orchestrating a Mixed Reality Performance" In *ACM Conf. on Computer-Human Interaction (CHI'01)*, 2001, pp 38-45.
2. Greenberg, S., Fitchett, C., "Phidgets: Easy Development of Physical Interfaces through Physical Widgets," In *ACM Symp. on User Interface Software and Technology (UIST'01)*, 2001, pp 209-218.
3. Klemmer, S., Sinha A., Chen J., Landay J., Aboobaker N., Wang A., "SUEDE: A Wizard of Oz Prototyping Tool for Speech User Interfaces," In *ACM Symp. on User Interface Software and Technology (UIST'00)*, 2000, pp 1-10.
4. Li, Y., Hong, J., and Landay J. "Topiary: A Tool for Prototyping Location-Enhanced Applications," In *ACM Symp. on User Interface Software and Technology (UIST'04)*, 2004, pp 217-226.
5. MacIntyre, B., Gandy, M., Dow, S., Bolter, J.D., "DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences", In *ACM Symp. on User Interface Software and Technology (UIST'04)*, 2004, pp 197-206.
6. Maulsby D., Greenberg S., Mander R., "Prototyping an Intelligent Agent through Wizard of Oz", In *ACM Conf. on Human Factors in Computing Systems (InterCHI'93)*, 1993, pp 277-284.