# The Design and Experimental Evaluation of a Tool to Support the Construction and Wizard-of-Oz Testing of Low Fidelity Prototypes

Christopher Hundhausen, Stephen Trent, Anzor Balkar, and Mohamed Nuur
*Visualization and End User Programming Laboratory*
*School of Electrical Engineering and Computer Science*
*Washington State University*
*Pullman, WA 99164-2752 USA*
*{hundhaus, strent, abalkar, mnuur}@eecs.wsu.edu*

## Abstract

*Because they are easy to create and modify, low fidelity prototypes are commonly used in early evaluations of user interface designs. Designers typically use either pen-and-paper or various computer-based tools to create and test low fidelity prototypes. However, our informal analyses of these technologies indicate that they do not optimally support the two key, complementary tasks of (a) prototype creation and (b) wizard-of-oz testing. To address this problem, we have developed WOZ PRO, a pen-based software environment for the quick and easy creation and testing of low fidelity user interface prototypes. When designing interface screens in WOZ PRO, designers can easily (a) propagate a design change to other related screens, and (b) specify the set of screens that are reachable from a given screen. In a wizard-of-oz test, WOZ PRO reduces the cognitive load on the wizard by allowing navigation only to those next screens that are valid. An experimental comparison of WOZ PRO and pen-and-paper provides evidence of WOZ PRO's potential, and insight into how the design of WOZ PRO might improved.*

## 1. Introduction

Central to the user-centered design process is the iterative design and evaluation of user interface prototypes. Such prototypes are typically created at various levels of *fidelity*, which refers to how well the prototype reflects the functionality, completeness, and polish of the final product. Because they are easy and cheap to create and modify, low fidelity prototypes can help designers to obtain invaluable design feedback on early user interface designs.

User interface designers typically construct low fidelity prototypes using one of three technologies:

- simple art supplies (e.g., pen, paper, and scissors);
- general-purpose graphics and presentation software not specifically designed for low fidelity prototyping (e.g., PowerPoint®, Photoshop®); and
- custom software specifically designed for low fidelity prototype creation (e.g., SILK [1], DENIM [2]).

In our own experiences with developing low fidelity prototypes with these existing technologies, we have observed that none of them appears to be optimized for the key, complementary activities of (a) rapidly creating a user interface prototype, and (b) running wizard-of-oz tests in which a human wizard simulates the user interface prototype as test users interact with it. With respect to these existing technologies, two problems in particular stand out:

1. *Design change propagation is cumbersome*. When iteratively designing low fidelity screen sketches, designers often need to go back and make design changes that impact multiple screens. With art supplies, this requires designers either to erase and redraw elements on multiple screens, or to recreate the screens altogether. With some computer-based tools, designers can avoid some of this repetitive work by exploiting cloning and template features. However, in our experience, such features do not make it easy to quickly apply design changes to only a subset of the screens that compose a prototype.
2. *Running wizard-of-oz studies incurs a potentially high cognitive load*. When developing a new user interface, designers typically run wizard-of-oz studies on prototypes with dozens, and sometimes hundreds, of screens. In order to be responsive to a test user's actions, human wizards not only have to have a solid understanding of the user interface they are simulating; they must also have a good organizational scheme, so that they can quickly navigate to the correct screen in response to the test user's actions. This places on human wizards a potentially high cognitive load, which can be eased only through the use of ad hoc screen labeling or hyperlinking schemes.

The problems with existing technologies identified above raise two research questions:

| | |
|---|---|
| RQ1: | *Can we develop a computer-based typing tool that promotes better performance in low fidelity prototyping construction and testing tasks than simple art supplies—the current "gold standard" medium for low fidelity prototyping?* |

RQ2: *What might the design of such a computer-based tool look like?*

To address these questions, we present here a new pen-based low fidelity prototyping tool, together with an experimental study that compares our tool to pen-and-paper. Our new pen-based tool, WOZ PRO (Wizard-of-Oz Prototyper), has been designed specifically for interaction designers who engage in the common tasks of (a) iteratively designing low fidelity user interface prototypes and then (b) subjecting them to wizard-of-oz testing. As we shall see, in an experimental study, WOZ PRO compared favorably to simple art supplies, and we gained insight into how to design an a computer-based tool to support wizard-of-oz testing even better.

After reviewing related work, the remainder of this paper describes the design of WOZ PRO, presents and discusses our experimental evaluation, and concludes by summarizing our contributions and identifying directions for future research.

## 2. Related Work

Over the last several years, considerable effort has been dedicated to developing computer-based prototyping environments that incorporate the "free-form" attributes of arts supplies. The emphasis of this body of work has been to harness the speed, flexibility, and design memory offered by a computer-based system with the high *provisionality* [3] of the arts supply medium.

Ozlab [4] was designed specifically to support wizard-of-oz studies of software with speech recognition capabilities. Ozlab works on two networked computers: one computer is used display the user interface (UI) screen to the test user, while the second computer is used by the wizard to manipulate the UI screen in response to test user input. Like WOZ PRO, Ozlab supports the generation of UI screens; however, unlike WOZ PRO, Ozlab does not support prototyping of UI screen transitions or navigation.

SILK [1] and DENIM [2] are pen-based low fidelity prototyping environments that allow designers to sketch user interface screens and create storyboards to specify the transitions between screens. Whereas SILK is designed to support general user interfaces, DENIM focuses on websites. In addition, unlike DENIM, SILK supports a widget recognition engine that attempts to recognize, and assign functionality to, sketched interface components, thus allowing low fidelity prototypes to evolve into high fidelity prototypes.

WOZ PRO has much in common with both SILK and DENIM, including support for (a) sketching user interface screens with a stylus, (b) specifying screen transitions via storyboards, and (c) running user studies from within the environment. However, whereas WOZ supports facilitator-driven (wizard-of-oz) user tests, SILK and DENIM support user-driven user tests in which users interact directly with the prototypes. Moreover, in our experience, SILK and DENIM can initially prove difficult for designers to use, because they require designers to become familiar with a specialized gesture language.

PatchWork [5] is a low fidelity prototyping environment built upon the metaphor of "patches"—pieces of paper that can be cut to size and placed on a background screen. Patches can be linked to other screens. In "run" mode, a test user can interact with a PatchWork prototype by clicking on patches to navigate to the linked screens. Like WOZ PRO, PatchWork supports the construction of a network of linked screens. Unlike WOZ PRO, PatchWork is user-driven, rather than facilitator-driven.

Implemented for use with a digital whiteboard, FreeForm [6] is a sketch-based low fidelity prototyping environment designed to support group collaboration in the design process. Similar to SILK, FreeForm attempts to translate sketched text and user interface components into corresponding objects in Microsoft Visual Basic forms. However, FreeForm does not allow a designer to run user studies from within the environment.

InkKit [7] is a tablet PC-based low-fidelity prototyping environment that allows a designer to sketch screens, and to specify transitions by associating sketched controls with transitions to other UI screens. Like SILK, InkKit implements a graphical transformation engine that "beautifies" text, widgets, and other components sketched on the UI screen. Similar to FreeForm, InkKit attempts to translate sketched UI components into corresponding components in Microsoft Visual. However, as is the case with both SILK and DENIM, InkKit does not support facilitator-driven wizard-of-oz studies.

SketchWizard [8] is specifically catered to prototyping and testing pen-based UIs that support handwriting and gesture recognition. Like Ozlab, SketchWizard is designed to support wizard-of-oz studies via two networked computers. One computer displays UI screens to a test user, while the other is covertly operated by a human wizard who updates UI screens in response to the test user's actions. SketchWizard supports a sophisticated wizard interface that enables the wizard to browse the test user's event stream, quickly translate sketch objects into higher fidelity objects, respond to user interaction by constructing and displaying new screen elements, and record the test user's interactions for future playback.

SketchWizard has much in common with WOZ PRO—most notably, its focus on supporting facilitator-driven wizard-of-oz studies. However, whereas SketchWizard is designed specifically to simulate gesture recognition and beautification systems, WOZ PRO focuses on simulating simple click-based navigation. In addition, SketchWizard allows wizards to respond to user input on the fly, whereas WOZ PRO requires designers to specify transitions prior to a user test. As such, the WOZ PRO runtime wizard interface is far simpler than that that of SketchWizard.

In addition to the computer-based systems just reviewed, a line of empirical research into low fidelity prototyping is related to the work presented here. Virzi *et al.* [9] and Sefelin *et al.* [10] show that low-fidelity prototypes elicit just as much feedback as their high-fidelity counterparts, regardless of the software development stage in which they are used. In a similar vein, Walker *et al.* [11] conducted an experimental study to determine whether or not presentation medium might have an effect on prototype efficacy. In this study, the same hand-drawn prototype was presented both on paper and on a computer. The study concluded that there were no significant differences between the two versions with respect to either the feedback they elicited or the usability issues they uncovered..

## 3. WOZ PRO Environment

WOZ PRO is a pen-based software environment that enables one both to create low fidelity prototypes, and to run wizard-of-oz tests on those prototypes. We developed WOZ PRO through an iterative, user-centered design process [12].We created a low fidelity prototype of a preliminary design using the SMART® Notebook whiteboarding software and a SMARTBoard® electronic whiteboard. We then iterated through two rounds of wizard-of-oz testing, each involving three participants. In these tests, participants were tasked with (a) designing a low fidelity prototype of a website according to specifications we provided, and then (b) running a wizard-of-oz study on the prototype.

Running wizard-of-oz studies in which participants themselves ran wizard-of-oz studies posed two notable challenges. First, participants were initially confused about which tool to use to develop and test their prototypes—SMART® Notebook, or our low fidelity prototype developed in SMART® Notebook. To resolve this issue, we resorted to draping a towel over all interface of components of the SMART® Notebook software, leaving in view only the screens that composed our prototype. Second, when it came time for participants to run a wizard-of-oz study on their prototypes, they initially became confused about their role as wizards—a role that we (as the researchers) had been playing for the first part of the study. We therefore had to take care, in our instructions, to explain that their job in the second part of the study was to run a study on their prototype with us (the researchers) as test users. Once we refined our explanation of this, participants caught on quickly.

### 3.1 Overview of Environment

After three design iterations on our low fidelity prototype, our design was stable enough to implement as a high fidelity prototype. Developed for the Tablet PC® using the Microsoft .NET® 2.0 framework and the Tablet PC® SDK, our high fidelity prototype of WOZ PRO contains three main modes that are clearly denoted by the tabs at the bottom of the interface (see Figure 1):

1. *Design Mode* allows the user to create, clone and edit individual interface screens by sketching with the pen, and inserting image files as interface elments. In addition, the user can propagate edits made on one screen to other similar screens, as illustrated below.
2. *Edit Screen Transitions Mode* allows designers to edit a state-transition network (STN) of their screen sketches, which specifies the set of screens that are reachable from each screen.
3. *Run Mode* allows human wizards to run wizard-of-oz studies by presenting screen sketches to test users, and navigating screen sketches in response to test users' actions. WOZ PRO constrains the choice of next possible screens to only those that are valid according to the STN defined by the designer.

### 3.2 Sample Usage Scenario

To illustrate how WOZ PRO supports the iterative design and testing of low fidelity prototypes, we now walk through an example usage scenario in which we design and test a prototype website for Pedro's PC—a fictional online store that sells computer hardware. The first step is to sketch out the eight screens of the website. Since all of the screens have a similar appearance, we can make judicious use of the "clone screen" feature; we create each subsequent screen by modifying a clone of the previous screen.

Suppose that, after creating all eight screens, we realize that we have made a mistake: Pedro's PC does not sell

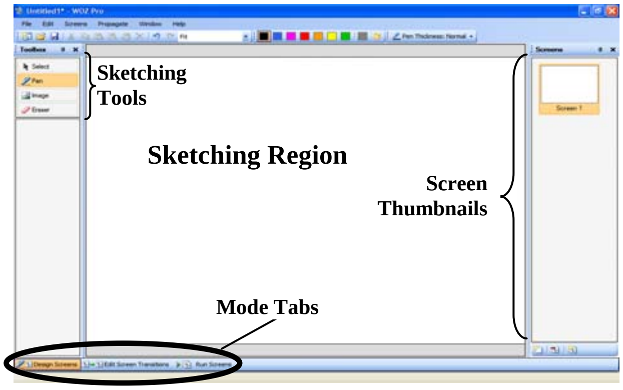


**Figure 1**. Annotated Snapshot of WOZ PRO Environment

CPUs, so we need to remove that item from the navigation bar on the left. Rather than having to remove that element from all eight screens, we can instead remove it from any one of the screens, and then choose "Propagate last change(s)…" from the "Edit" menu. This brings up a list of the last changes we have made, along with a list of screens to which we can apply the change (see Figure 2a). We choose the most recent change—a deletion action—and select all screens as the target of the change. After clicking on "OK," WOZ PRO automatically applies the change to all of the screens.

We are now in a position to define the transitions between screens. We first switch to "Edit Screen Transitions" mode by clicking on the corresponding tab (see labeled area of Figure 1). We are presented with a STN of our prototype; by default, screen transitions are defined between each pair of screens that were created in sequence. However, in our Pedro's PC website, navigation is more complex. To bring the STN into accordance with the Pedro's PC website, we first drag and drop all screens onto the canvas (Figure 2b). We then drag out transitions between appropriate screens, yielding the STN shown in Figure 2c.

In the final step, we run a wizard-of-oz test on our prototype. Clicking on the "Run Screens" tab (see encircled area of Fig. 1a), we enter a mode similar to "presentation" mode in PowerPoint®. The key difference is that the set of next possible screens from any given screen is constrained, reflecting the transitions in the previous step. In particular, navigating to a screen in response to a test user's actions merely requires us to click on the navigation menu in the lower-left portion of the screen, and then to select the next screen from a constrained menu of choices each of which is previewed in a smaller inset window (lower left-hand corner of Figure 2d),. Thus, in contrast to art supplies and other computer-based tools, a human wizard using WOZ PRO does not have to struggle to remember, and then navigate to, valid choices. Instead, the navigation menu in WOZ PRO contains only those screens that can be reached from a given screen, easing the cognitive load on the wizard.

## 4. Experimental Evaluation

To evaluate the WOZ PRO Environment, we conducted an experimental study with two main hypotheses:

H1: *WOZ PRO will promote faster, more accurate construction of user interface prototypes than art supplies*

H2: *In wizard-of-oz prototype studies, WOZ PRO will promote more efficient screen transitions and fewer screen transition errors than art supplies*.

To test these hypotheses, we conducted a between-subjects experimental study with two conditions defined by prototyping tool: WOZ PRO (WOZ) and art supplies (ART). To gauge the effectiveness of the prototyping tools being compared, we defined the following four dependent measures:

1. Time to construct prototype
2. Average time delay between screen transitions
3. Number of screen transition errors

In addition, participants completed an exit questionnaire that elicited their subjective opinions on the low fidelity prototyping tool they used. An analysis of the questionnaire results is beyond the scope of this paper.

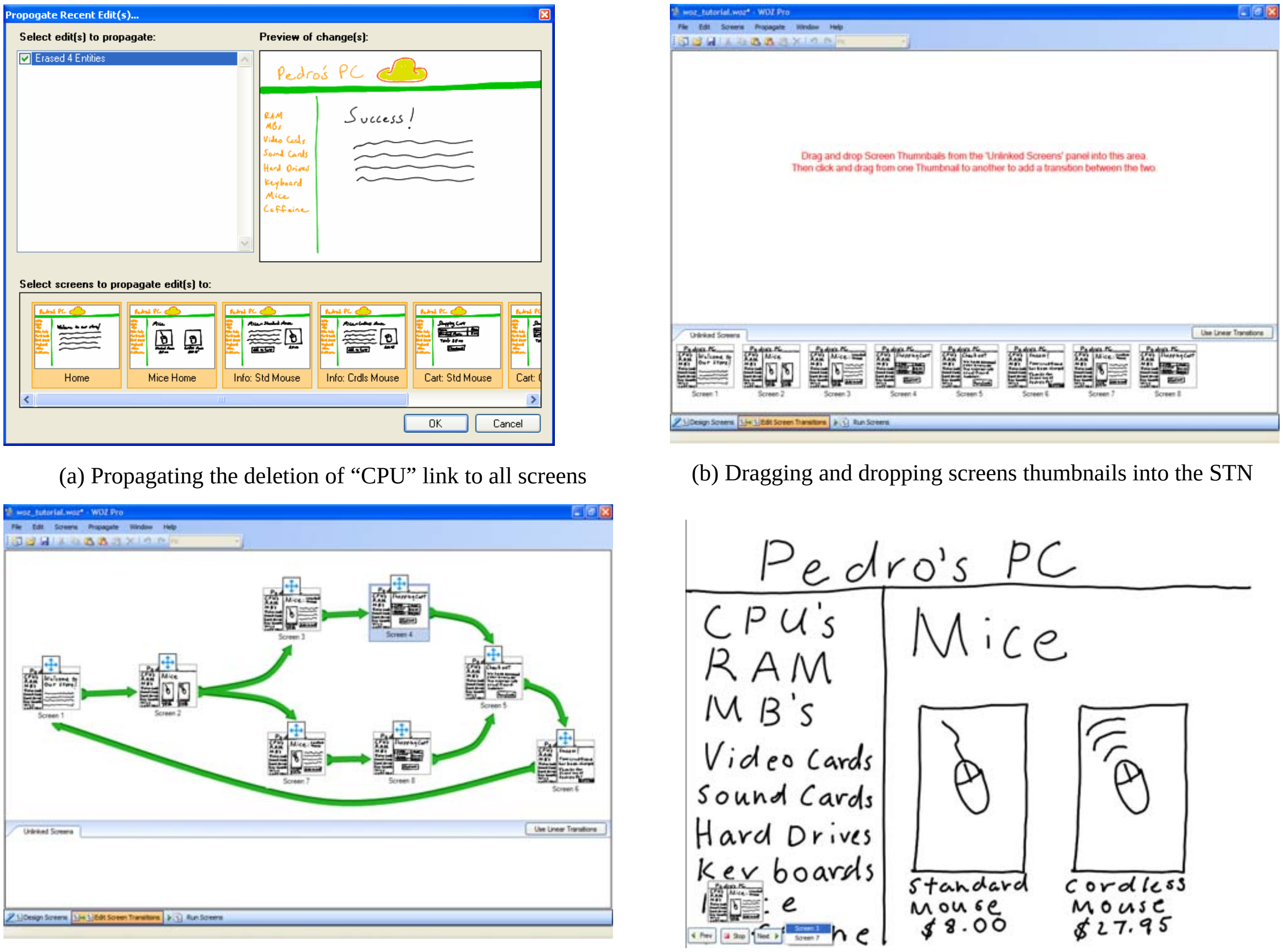
### 4.1 Participants

We recruited 19 computer science students and recent graduates to participate in the study (15 males, 4 females). Twelve of these participants were recruited from the Spring, 2007 offering of CptS 443/580, the undergraduate and graduate human-computer interaction course at Washington State University. These students received course credit for their participation. Seven additional participants who had completed the CptS 443/580 course at Washington State University during the previous two years were recruited from the local area. These individuals were given a $25 honorarium for their participation. All study participants had a basic understanding of low-fidelity prototyping and wizard-of-oz studies due to their past or present enrollment in the CptS 443/580 course.

### 4.2 Materials and Tasks

Study participants were required to create and test, in wizard-of-oz fashion, two target websites—one in the tutorial segment of the session, and another in the study segment of the session. For the tutorial segment of the session, we provided participants in both conditions with a web browser running a simple low-fidelity (sketched) version of a computer peripherals purchasing website, along with informationally-equivalent sets of instructions for constructing and testing prototypes with their respective tools (art supplies or WOZ PRO).

For the main part of the study, we gave participants a web browser running a more complex website based on an online banking website developed by Walker *et al.* [11] for their experimental study of prototyping techniques. Through pilot studies, we adapted the website of



(a) Propagating the deletion of "CPU" link to all screens

(b) Dragging and dropping screens thumbnails into the STN

(c) Final STN created in "Edit Screen Transitions" mode

(d) Previewing a Screen in "Run Screens" Mode

**Figure 2.** Snapshots from a Sample Session with the WOZ PRO Environment

Walker *et al.* by reducing its complexity such that our study participants could create and test the website within the allotted time of each experimental session; however, most of the usability problems of the site were retained. Our final version of the website consisted of 21 screens and 157 possible transitions.

Participants completed two types of tasks in the study: (a) prototype creation tasks, and (b) wizard-of-oz demonstration tasks. Prototype creation tasks involved sketching screens, defining transitions between screens, and organizing screens for presentation to a test user. For the wizard-of-oz demonstration, participants simulated five tasks in their website while the experimenter played the role of a test user.

In the ART condition, we provided participants with sheets of standard 8½ × 11 inch paper for creating and presenting their website prototypes. Sketching was performed with black permanent markers. White out pens and correction tape were provided for correcting errors. In contrast, in the WOZ condition, participants used the WOZ PRO software (see previous section) to create and present their website prototypes. Participants worked with WOZ PRO on a Hewlett-Packard Compaq tc4200 tablet PC with a 14" screen, 512 MB of RAM and an Intel Centrino 2.0 GHz processor. For both tasks, participants accessed the target websites they were to create and test through a web browser running on an identical Hewlett-Packard Compaq tc4200 tablet PC. They performed their work in an area directly adjacent to that PC.

We used the Morae® Recorder to record participants' work sessions. In particular, we created both (a) lossless recordings of the target screens participants viewed as they constructed their websites, and (b) the screens that participants actually created and presented (either with art supplies or WOZ PRO). The picture-in-picture feature of Morae® Recorder was used to synchronize participants' work with the target screens that they viewed. By viewing these recordings, we were able to gather our dependent measures.

## 4.3 Procedure

In order to guard against the possibility that individual differences in sketching speed might confound the results, we had participants complete a sketching pretest prior to the experimental sessions. We then assigned participants to the two conditions such that the mean pre-test performances of the two conditions were optimally matched.

Each participant completed the study individually. In each session, which lasted approximately two hours, participants were first given 30 minutes to work through a tutorial in which they were introduced to the prototyping tool they were to use (art supplies or WOZ PRO), and practiced creating the screens for, and demonstrating, a simple website using that tool.

Participants then moved on to the main study task, for which they were given 90 minutes to create and test the more complex study website as quickly as possible, without sacrificing accuracy. This included 75 minutes to create the website, and 15 minutes to simulate the website as a test user (the experimenter) performed tasks. The instructions partitioned the website creation and simulation tasks into the following four subtasks:

1. *Create the website screens*. This entailed navigating the target website and making sure to sketch out all of the website screens.
2. *Define the screen transitions*. This entailed either using the "Define Screen Transitions" mode of WOZ PRO (WOZ condition), or actually sketching out a state transition diagram (ART condition).
3. *Practice navigating screens*. This entailed either using the "Run Screens" mode of WOZ PRO to practice navigating through possible screen sequences (WOZ condition), or organizing web page sketches so that they could be quickly and easily navigated in a manner consistent with the state-transition diagram developed earlier (ART condition).
4. *Simulate website for a test user*. In this final step, the experimenter entered the room and played the role of a test user by attempting to perform five tasks with the website. Immediately before attempting to perform a given task, the test user would clearly announce the task he was trying to perform. He would then use a pen to point to the link that he wanted to click on, and the participant would be responsible for navigating to the correct next screen. Participants were not informed of what tasks the test user was going to perform ahead of time. Rather, they were asked to prepare for all possibilities.

Participants completed their study sessions by taking up to 15 minutes to complete an exit questionnaire that elicited their subjective opinions on the prototyping tool they used.

## 4.4 Measuring the Dependent Variables

We measured all dependent variables by carefully viewing the recordings. To measure prototype construction time, .we determined the time at which participants started and stopped the three prototype construction subtasks—sketching, defining screen transitions, and practicing screen navigation (dependent variable 1).

In a similar fashion, we computed the time delay incurred for screen transitions during wizard-of-oz tests by noting the start and stop times for each correct screen transition (dependent variable 2). If an uncorrected screen transition error was made, or if no transition was defined by the study participant, then no time delay was calculated for the screen transition.

The recordings were also used to identify errors made by study participants during the simulation of their prototypes (dependent variable 3). We distinguished two types of screen transition errors: (a) presentation of an incorrect screen during the simulation (regardless of whether the participant recognized the error and subsequently corrected it); and (b) missing a transition—if a participant failed to complete all transitions in a sequence, each missed screen was counted as an error.

In addition to counting errors, we also tabulated screen transition errors that were subsequently corrected. A corrected screen transition occurred when the participant recognized that an incorrect transition had been made, and subsequently found the correct transition and associated screen.

## 4.5. Results

Before statistically analyzing our data for each dependent variable, we applied Shapiro-Wilk tests to verify that the data met assumptions of normality. According to these tests, our data for dependent variable met assumptions of normality, whereas our data for dependent variables 2 and 3 did not. This meant that, while we could use parametric ANOVAs to test for differences with respect to dependent variables 1, we had to use non-parametric Kruskal-Wallis tests for dependent variables 2 and 3.

Table 1 presents a summary of our data with respect to each of the three dependent variables. Included in Table 1 are the results of appropriate statistical tests (ANOVA or Kruskal-Wallis) that compared the means of the two conditions. As the table indicates, we detected significant differences between the two conditions with respect to just two of our four dependent measures:

1. The ART participants required significantly less time on average to complete the “define transitions” subtask of the prototype construction task.
2. The WOZ participants fixed significantly more transition errors than the ART participants.

### 4.6 Discussion

.

Our results do not support our hypothesis that WOZ Pro promotes faster prototype construction. With respect to the first subtask of creating screens, we observed that, although the clone screens and propagate change functions were available, WOZ participants tended not to use these features. Perhaps this was because they were not yet familiar with the tool, which they had used for just 15 min. prior to the study task.

In the second subtask of creating screen transitions, WOZ Pro participants spent significantly more time than the ART participants. We can explain this difference by observing the fundamentally different activities in which participants in each condition engaged. In the ART condition, participants hand-sketched state transition diagrams showing the navigation paths among screens. In the WOZ condition, participants used the “edit screen transitions” mode to define state transition diagrams. The state transition diagrams created by WOZ participants quickly became too large and complex to fit on a single screen. We observed that participants had substantial difficulties navigating these diagrams as they attempted to add screens and define transitions.

**Table 1.** Mean Performance with Respect to the Three Dependent Variables by Condition

| Condition | *n* | M | SD | DF | *F*- or<br>*H*-val.* | *P*-val. |
|---|---|---|---|---|---|---|
| *1. Time to construct prototype* | | | | | | |
| *1a. Time to sketch screens* | | | | | | |
| ART | 9 | 46.05 | 7.23 | 1 | 0.31 | 0.585 |
| WOZ | 10 | 44.02 | 8.46 | | | |
| *1b. Time to define transitions* | | | | | | |
| ART | 9 | 16.06 | 5.20 | 1 | 14.44 | **0.001** |
| WOZ | 10 | 28.90 | 8.84 | | | |
| *1c. Time to practice navigating screens* | | | | | | |
| ART | 9 | 4.52 | 2.39 | 1 | 0.68 | 0.420 |
| WOZ | 10 | 5.61 | 3.28 | | | |
| *2. Screen transition time* | | | | | | |
| ART | 9 | 7.33 | 1.32 | 1 | 1.50 | 0.221 |
| WOZ | 10 | 6.60 | 2.72 | | | |
| *3. Screen transition errors made/fixed* | | | | | | |
| *3a. Transition errors* | | | | | | |
| ART | 9 | 2.67 | 1.00 | 1 | 0.00 | 0.967 |
| WOZ | 10 | 3.40 | 2.59 | | | |
| *3b. Transition errors fixed* | | | | | | |
| ART | 9 | 0.00 | 0.00 | 1 | 5.62 | **0.018** |
| WOZ | 10 | 0.15 | 0.19 | | | |

*An ANOVA was used for dependent variables 1; Kruskal-Wallis tests were used for variables 2 and 3.

Although no significant difference existed between the conditions with respect to the third prototype construction subtask of practicing screen navigation, we observed interesting differences in the activities in which participants in each condition engaged. In the ART condition, most study participants spent the majority of the time shuffling screens and placing them in piles on the desk next to the work area. The purpose of the piling was to group related screens in logical sequences that allowed the participants to easily access them during the prototype simulation sessions. In contrast, WOZ participants switched to the “Run Screens” mode and practiced navigating their prototypes.

With respect to our hypothesis that WOZ Pro will promote more efficient screen transitions, we found no differences in the average screen transition time and average number of transition errors. We suspect that no differences emerged here because of a flaw in the WOZ interface’s “Run Screens” mode. WOZ participants were unable quickly and consistently to identify the next screen to which to navigate, because the thumbnail previews proved too small to visually distinguish from each other. In contrast, screen transition time in the ART condition turned on the ability of participants to locate the correct “pile” of screens and mechanically place the correct screen in the work area. This approach produced more consistent screen transition times, as indicated by a comparison of the standard deviations of average transition times of the ART and WOZ conditions (1.3 seconds and 2.7 seconds, respectively).

Finally, we found limited empirical support for our hypothesis that WOZ Pro promotes significantly fewer screen transition errors than art supplies. While there was no significant difference between the two conditions with respect to average number of transition errors, we did find that the WOZ participants were able to correct a small but significantly higher number of screen transition errors than the ART participants. We attribute this difference to the fact that WOZ Pro required participants to explicitly define all transitions, leading to feedback in cases where transitions were incorrectly defined. In particular, when WOZ participants could not make a screen transition because the transition was not defined, they were forced to reexamine their screen transition diagrams and define the correct transitions. In contrast, ART participants had no built-in constraints on their transitions. While their screen transition diagrams may have been correct, there was no mechanism to ensure that they adhered to their transition diagrams, aside from their own memory. Further, ART participants did not receive feedback when they failed to make screen transitions that matched their transition diagrams.

## 5. Summary and Future Work

We have argued that existing computer-based low fidelity prototyping tools do not optimally support the two complementary tasks in which interface designers routinely engage: (a) prototype construction and (b) facilitator-driven wizard-of-oz studies. To address this problem, we have presented WOZ PRO, a new pen-based low fidelity prototyping environment that aims to make creating low fidelity prototypes as quick and easy as using pen-and-paper, while providing advantages over pen-and-paper in both prototype construction tasks (viz., *cloning* and *change propagation*) and wizard-of-oz testing tasks (viz., *constrained transitions*). In an experimental comparison against pen-and-paper, we found that WOZ Pro's requirement that designers create full-fledged state transition diagrams led to significantly slower performance in the task of defining transitions; however, that requirement also led to the identification and repair of significantly more screen transition errors. Thus, having to specify STNs appears to be a mixed bag: they make prototype creation more cumbersome, but they reduce transition errors when it comes time to test a prototype with users.

In future work, we would like to pursue two complementary directions. First, we would like to use insight gained from observing ART participants to improve upon WOZ Pro's interface for specifying screen transitions. After they sketched their STNs, ART participants placed the screens related to each area of the website into separate piles. Then, when they were asked to simulate a task, they simply reached for the pile corresponding to the task. Notably, ART participants did not specify, in their piles, every single transition. Rather, they relied upon the interaction context to generate the next screen. While this technique led to undetected transition errors, we were struck by how easy and natural it appeared to be. In ongoing work, we are exploring ways to replace state transitions with the concept of a "task pile": a container into which users can drag-and-drop screens that are related to a given task.

Second, our study used *transition speed* as a gauge of tool success. However, a key goal of wizard-of-oz studies is not only to test the feasibility of existing user interface designs, but to generate new ideas. Thus, what may matter more than transition speed in wizard-of-oz tests is whether the wizard can spontaneously generate new screens in response to unanticipated actions. In future work, we would like to explore ways in which WOZ Pro might support a greater degree of creativity and spontaneity, so that wizards can respond more dynamically to test users' actions.

## 6. Acknowledgments

This research was supported by a Hewlett Packard "Technology for Teaching" Award (U06TFH0033E).

## 7. References

[1] J. A. Landay and B. A. Myers, "Interactive sketching for the early stages of user interface design," in *Proceedings of the ACM CHI '95 Conference on Human Factors in Computing Systems*. ACM Press, New York, 1995, pp. 43-50.

[2] J. Lin, M. Newman, J. Hong, and J. Landay, "DENIM: Finding a tighter fit between tools and practice for web site design," in *CHI 2000 Conference Proceedings*. ACM Press, New York, 2000, pp. 510-517.

[3] Green, T.R.G., and Blackwell, A. Cognitive dimensions of information artefacts: a tutorial. http://www.cl.cam.ac.uk/~afb21/CognitiveDimensions/CDTutorial.pdf. Accessed 18 February, 2008.

[4] L. Molin, "Wizard-of-Oz prototyping for cooperative interaction design of graphical user interfaces," in *Proc. NordiCHI '04*. ACM Press, New York, 2004, pp. 277-284.

[5] M. van de Kant, S. Wilson, M. Bekker, H. Johnson, and P. Johnson, "PatchWork: A software tool for early design," in *Human Factors in Computing Systems: CHI 98 Summary*. ACM Press, New York, 1998, pp. 221-222.

[6] B. Plimmer and M. Apperley, "Computer-aided sketching to capture preliminary design," in *Proc. Third Australasian Conference on User Interfaces*, vol. 3. ACM Press, New York, 2002, pp. 9-12.

[7] R. Chung, P. Mirica, and B. Plimmer, "InkKit: a generic design tool for the tablet PC," in *Proc. 6th ACM SIGCHI New Zealand Chapter's International Conference*. ACM Press, New York, 2005, pp. 29-30.

[8] R. C. Davis, T. S. Saponas, M. Shilman, and J. A. Landay, "SketchWizard: Wizard of oz prototyping of pen-based user interfaces," in *Proc. 20th Annual ACM symposium on User Interface Software and Technology*. ACM Press, New York, 2007, pp. 119-128.

[9] R. A. Virzi, J. L. Sokolov, and D. Karis, "Usability problem identification using both low- and high-fidelity prototypes," in *Proceedings 1996 SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, New York, 1996, pp. 236-243.

[10] R. Sefelin, M. Tscheligi, and G. Verena, "Paper prototyping - What is it good for? A comparison of paper- and computer-based low-fidelity prototyping.," in *Proc. 1996 SIGCHI Conference on Human Factors in Computing Systems*. ACM Press, New York, 2003, pp. 778-779.

[11] M. Walker, L. Takayama, and J. A. Landay, "High-fidelity or low-fidelity, paper or computer medium?" in *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*. The Human Factors and Ergnomics Society, Santa Monica, 2002, pp. 661-665.

[12] D. Norman and S. Draper, *User-centered system design*. Lawrence Erlbaum Assoc., Mahwah, NJ, 1986.