

d.stix: Testing and Analyzing with Wireless Prototypes

Leith Abdulla, Dan Wilson
Stanford University HCI Group
Computer Science Department
Stanford, CA 94305-9035, USA
{eleith, dwilson1}@stanford.edu

ABSTRACT

Computation is no longer tied to the desktop, and rapid prototyping tools shouldn't be either. It can be difficult to test prototypes of information appliances that are still tethered to a desktop inside a lab. Thus, we built d.stix, a wireless prototyping toolkit to enable designers to rapidly prototype the wireless devices of tomorrow. Our work most strongly contributes to the 'test' stage of the design-test-analyze prototyping cycle. The results of our user study demonstrate that wireless prototypes allow for new testing environments and new user insights.

Author Keywords

Wireless prototyping, toolkits

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Prototype testing is critical to the iterative design process. User feedback is essential data that guides designers on what they are iterating, and how they do it. In our design classes, we have seen groups rapidly prototype PDAs, digital color mixers, digital cameras and mp3 players with d.tools. However, when conducting user studies, students dragged around laptops, desktops, wires, and ethernet cables. This affected setup time for demos, required additional design to 'hide' wires and PCs, and constrained demos to take place indoors and near outlets.

d.stix allows designers to build devices with interactions that allow users to completely rotate a GPS prototype or to test a digital color mixer in the middle of a classroom. These interactions lead to new insights from testers that would have otherwise been missed with a prototype constrained by wires.

TOOLS FOR TESTING

d.stix is a wireless approach for d.tools [1], a prototyping toolkit that enables designers to rapidly prototype ubiquitous computing devices. The d.tools prototyping approach consists of two steps: designing an interaction model visually through a desktop application and testing this design physically through smart hardware that is the physical counterpart of the virtual model.

One major constraint of d.tools (and others [2,3]) is that the toolkits require power and a communication cable for the physical prototype itself. This constraint tethers the prototype to the designer's desk, greatly limits mobility in what may eventually be a wireless device, and constrains the testing environment itself. Being able to perform user studies in a realistic environment can improve feedback from testers. Thus, we redesigned d.tools' hardware architecture to support a wireless infrastructure.

IMPLEMENTATION

Hardware: d.stix is based on the embedded Gumstix computer, running a customized version of Linux. The system is powered by an Intel Xscale 400 MHz processor, and Wi-fi connectivity is provided by a Netgear compact flash card (see Fig. 1). We power d.stix with three 9-volt batteries, and we use an i2c bus for communication between devices.

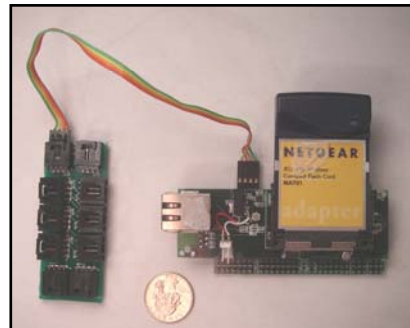


Figure 1: The d.stix hardware.

Software: d.stix prototypes are designed using the d.tools eclipse plugin, which affords the visual programming of a device's interaction model using a statechart editor. Software running in the d.stix wirelessly communicates with d.tools using UDP. This communication ensures the physical device is always in the proper state according to the interaction model.

USER STUDY

We ran a comparative study to observe the differences between testing wired and wireless prototypes. We built a Google Maps navigator that scrolls a Google Maps window through tilting the device itself in the desired direction (see Fig. 2). Both a wireless and PC-tethered version of the appliance were constructed and presented to users to test in

a small room setting. We ran one study of a group of six HCI students who were familiar with d.tools. We observed and interviewed each user. Our questions centered around their thoughts on how we could improve the prototype. Three themes for wireless testing emerged:

New user insights: One tester found difficulty with orienting our device properly and suggested that the next design iteration should include visual cues to indicate compass direction. No one questioned orientation for the tethered prototype, because users assume the back of the device was where the wires came out. Another user attempted to tilt the device 360 degrees and discovered that the system could not gracefully handle the input. No one attempted this type of motion with the wired device.

Increased user mobility: During the tethered study, the users formed a queue next to the prototype, waiting for their turn to use the device one at a time. Users testing the wireless prototype adapted a more social posture as they passed the device around the room, observing each other and conversing at a comfortable distance.

Comfortable testing environment: The users found it simpler to “walk up and use” the wireless device without worrying about avoiding the wires and managing cables. They also considered the wireless device to be less fragile.



Figure 2: The tilt-sensing Google Maps navigator.

RELATED WORK

Rapid Prototyping Systems for Product Design: As mentioned earlier, this work builds directly upon d.tools, developed by Hartmann et. al [1]. It also draws inspiration from the Phidgets [2], Calder [3], and iStuff [4] frameworks. Phidgets’ “Physical Widgets” communicated with each other through a wired protocol, but the authors mentioned they were experimenting with wireless protocols as an aside. The Calder paper contains a deep discussion of architectural issues that arise with wireless components, but does not discuss the role that wireless prototypes play in the reflective prototyping process. The iStuff framework facilitates distributed, wireless interactions in a room environment. We focus less on distributed interactions and are instead enabling the iterative development of single

wireless prototypes, with the eventual goal of allowing the prototype to be tested far away from a lab environment.

Prototyping methodology: Virzi et Al [5] analyzed the relative effectiveness of low and high fidelity prototypes in uncovering usability problems. Our work targets a specific prototyping capability and its effect on testing; untethering a prototype of a wireless product can be perceived as increasing its overall fidelity, but the effect is quite specific.

FUTURE WORK

The impact of complete prototype portability on the testing process remains to be explored, as d.stix prototypes currently do not have the capability to leave the host PC’s wireless network. For the prototype to be taken deeper into the field, we need a way of moving the d.tools statechart model onto the d.stix itself. This will involve the creation of Gumstix-compatible software that can interpret d.tools statechart files. Additionally, interfaces might be created to allow designers to make these portable prototypes leverage all that d.stix has to offer (wireless internet, email, ftp, and linux binaries), further enhancing prototype realism.

REFERENCES

- 1 Hartmann, B., S. R. Klemmer, M. Bernstein, *et al.* Reflective physical prototyping through integrated design, test, and analysis. In Proceedings of *UIST 2006: ACM Symposium on User Interface Software and Technology*, 2006.
- 2 Greenberg, S. and C. Fitchett. Phidgets: easy development of physical interfaces through physical widgets. *UIST: ACM Symposium on User Interface Software and Technology*, *CHI Letters* 3(2). pp. 209–18, 2001.
- 3 Lee, J., D. Avrahami, S. Hudson, *et al.* The Calder Toolkit: Wired and Wireless Components for Rapidly Prototyping Interactive Devices. In Proceedings of *ACM Symposium on Designing Interactive Systems*: ACM Press. pp. 167–75, August, 2004.
- 4 Ballagas, R., M. Ringel, M. Stone, and J. Borchers. iStuff: a physical user interface toolkit for ubiquitous computing environments. *CHI: ACM Conference on Human Factors in Computing Systems*, *CHI Letters* 5(1). pp. 537–44, 2003.
- 5 Virzi, R. A., J. L. Sokolov, and D. Karis, Usability problem identification using both low- and high-fidelity prototypes. In Proceedings of the *SIGCHI conference on Human factors in computing systems: common ground*. Vancouver, British Columbia, Canada: ACM Press. pp. 236-43, 1996.