

Using HTML to Create Early Prototypes

Jaya Vaidyanathan, Jason E. Robbins, David F. Redmiles

Department of Information and Computer Science

University of California, Irvine, CA 92697, USA

+1 949 824 8043

{jaya, jrobbins, redmiles}@ics.uci.edu

ABSTRACT

This paper discusses the use of HTML for creating early prototypes of interactive systems. The HTML prototyping technique uses an iterative approach for the creation of early prototypes. Low fidelity prototypes are iteratively refined into higher fidelity versions. The prototypes are easily accessible to project stakeholders. We evaluate the HTML prototyping technique with respect to criteria published in the CHI community.

Keywords

Early Prototyping, HTML, Iterative Design, Usability

INTRODUCTION

There is a growing debate in the HCI community about the comparative virtues of low and high fidelity prototypes. Low fidelity prototypes are more useful during the early prototyping phase where a large number of alternatives need to be explored. High fidelity prototypes on the other hand, help designers and customers better visualize the interface. An ideal technique would allow the designers to specify just enough detail to make and evaluate key design decisions regarding functionality, screen flow, and presentation.

This paper describes a technique for building low fidelity, low cost early prototypes that are iteratively refined into higher fidelity versions. HTML is used as a means for creating early prototypes that can be viewed in any web browser.

We are certainly not the first researchers who have recognized the value of low fidelity early prototyping and tried to reap its benefits. SILK is an interactive sketching tool that allows graphic designers to convert rough sketches into more high fidelity toolkit-specific representations [5]. The designers of the CHI Information Kiosk used iterative development to convert initial low fidelity prototypes into higher fidelity versions until the final product itself was developed [7]. "Cardboard computers" have also been suggested for creating low fidelity mock-ups of interfaces [2]. Wizard of Oz techniques are also commonly used for

early prototyping [1]. But the kinds of applications to which these techniques apply are limited. Commercial prototyping tools such as Visual Basic, HyperCard, and Macromedia Director make it difficult to focus on functionality and screen flow independently of presentation.

THE HTML APPROACH

In our approach the designer uses HTML pages to prototype user interface screens and HTML links to represent possible interactions. The prototyped screens are easy to construct because they evolve from unformatted text. No drawing or sketching ability is needed. Clicking on links allow evaluators to "run" the prototype in a web browser. Since the prototype is done in HTML, it is very clearly a prototype rather than an actual system; this helps evaluators uncouple functionality, screen flow, and presentation.

Our approach is most applicable to "screen-oriented" systems that have complex interactions among simple screens. For example, embedded systems such as automated teller machines (ATMs) have dozens of screens with carefully designed transitions. Efficiency and usability of these systems depends on well-designed screens and screen flow.

One key feature of our approach is the separation of functionality, screen flow, and presentation aspects of the prototype. Below we demonstrate a step-by-step method that addresses these aspects in turn.

Gas-Cash-Dash Example

The "Gas-Cash-Dash" system is a hypothetical point-of-sale system for self-service gas stations. Gas-Cash-Dash offers standard pay-at-the-pump features: read credit cards, dispense fuel, print receipts, and sell car washes. It also displays advertisements while dispensing fuel, offers banking services, and sells lotto tickets. These new capabilities add complexity to the system's UI dialog. We built an HTML prototype to explore and refine this dialog.

Effort sponsored by the Defense Advanced Research Projects Agency, and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-2-0021 and F30602-94-C-0218, and by the National Science Foundation under Contract Number CCR-9624846. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory or the U.S. Government.

Our prototyping method consists of the following five steps. The prototypes produced in each step can be found at <http://www.ics.uci.edu/pub/arch/htmlproto/>.

Step 1: Textually outline system functionality and content. Some of the text for this step may be copied from early requirements or vision statements. Check if critical functions are missing. E.g., we found that our initial system outline lacked the capability to print banking receipts.

Step 2: Break the outline into screens using a single HTML file and horizontal lines. Each screen contains comments on screen content, not final text. For each screen, list possible events that cause transitions to other screens. Check the amount of content on each screen and the availability of needed events. We realized that our system would need to detect when the customer removes a receipt.

Step 3: Define links between events and destination screens. At this point the prototype can be “run.” Check the length of typical usage scenarios and the effectiveness of navigation aids. We improved navigation by adding navigation options, e.g., “Press Cancel to Return to Main Menu.”

Step 4: Roughly format the screens using HTML tables to address presentation. Replace comments on the content of each screen with an initial draft of the text to be presented. The Gas-Cash-Dash system will have a fixed-size display, so we used fixed-size tables. We used GIF images to check critical screens at a higher level of fidelity.

Step 5: Break the prototype into one HTML page per screen. Visualize overall screen flow with site mapping tools (See Figure 1). Use web hit counters and forms to gather metrics and comments. Check for long sequences of screens, dead ends, and parallel structures. We used hit counters and forms that do not require programming or system administrator privileges on the web server.



Figure 1. Site Map Overview of Screen Flow.

EVALUATION

We chose the following criteria for evaluating our approach based on the desirable features for a prototyping environment [3] and on published prototyping evaluation criteria [6].

Appearance: HTML allows for a range of appearance fidelity from unformatted text to precise GIF images.

Screen Flow and Navigation: Existing site-mapping tools can be used to visualize overall screen flow.

Coding Effort: The HTML approach requires no coding. All interactions are captured in terms of hyperlinks between screens.

Iterative Improvement: HTML is a convenient medium for making design changes. The lag between deployment of the prototype and response from evaluators is short.

Availability and Quality of Tools: Commercial HTML editors are widely available. These are suitable for the task of prototyping, although that is not their intended use.

Getting Participants: The HTML prototype is very easy to access. The ability to evaluate the prototype from any location also encourages the stakeholders to provide prompt feedback to the designers.

Interactive Performance: If desired, processing delays can be simulated with minor script-writing effort.

Conceptual Simplicity: The HTML approach involves very few concepts, namely screens and links. These are familiar to anyone who has used HTML before.

CONCLUSIONS AND FUTURE WORK

Based on our evaluation we conclude that the HTML approach is an easy way to prototype and evaluate screen-oriented systems. It is expected to be particularly useful in allowing designers to address functionality and screen flow independently of presentation. In future work, we plan to integrate it with the notion of designers' expectations [4].

REFERENCES

1. Dahlback, N. et. al. Wizard of Oz Studies – Why and How, in *Proceedings of IUI '93*, ACM, 193-200.
2. Ehn, P. and Kyng, M. Cardboard Computers: Mocking-it-up or Hands-on the Future, in *Design At Work*, Ed. Greenbaum, J. and Kyng, M., Lawrence Erlbaum Associates Inc., Hillsdale, NJ, 1991, 169-195.
3. Hewett, T.T. Towards a Rapid Prototyping Environment for Interface Design: Desirable Features Suggested by The Electronic Spreadsheet, in *Proceedings of the HCI '89*, British Informatics Society Ltd., 305-314.
4. Hilbert, D.M. and Redmiles, D.F. An Approach to Large-Scale Collection of Application Usage Data Over the Internet, in *Proc. of ICSE '98*, ACM, 136-145.
5. Landay, J.A. and Myers, B.A. Interactive Sketching for the Early Stages of User Interface Design, in *Proceedings of CHI '95*, ACM Press, 43-50.
6. Collings, P.A. and Mackay, J. How to Evaluate Prototyping Tools (Allowing Designers to Design), in *Proceeding of OZCHI '92*, CHISIG, 62-69.
7. Salomon, G. A Case Study in Interface Design: The CHI '89 Information kiosk, in *Readings in Human-Computer Interaction: Toward the Year 2000*, Ed. Baecker, R.M., et. al., Morgan Kaufmann Publishers, 1995, 25-34.

