Forming Interactivity:

A Tool for Rapid Prototyping of Physical Interactive Products

Daniel Avrahami, Scott E. Hudson
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213
U.S.A.
nx6@cs.cmu.edu, scott.hudson@cs.cmu.edu

Abstract

The current practice used in the design of physical interactive products (such as handheld devices), often suffers from a divide between exploration of form and exploration of interactivity. This can be attributed, in part, to the fact that working prototypes are typically expensive, take a long time to manufacture, and require specialized skills and tools not commonly available in design studios.

We have designed a prototyping tool that, we believe, can significantly reduce this divide. The tool allows designers to rapidly create functioning, interactive, physical prototypes early in the design process using a collection of wireless input components (buttons, sliders, etc.) and a sketch of form. The input components communicate with Macromedia Director to enable interactivity.

We believe that this tool can improve the design practice by:

a) Improving the designer's ability to explore both the form and interactivity of the product early in the design process,
b) Improving the designer's ability to detect problems that emerge from the combination of the form and the interactivity,
c) Improving users' ability to communicate their ideas, needs, frustrations and desires, and
d) Improving the client's understanding of the proposed design, resulting in greater involvement and support for the design.

Keywords

Rapid prototyping, user-centered design, interaction design, design process, interactive models, 3D physical forms, passive RF

Introduction

In this paper we present the first version of a tool for rapid prototyping of interactive physical products, primarily in the design of different types of electronic handheld devices (e.g. – remote controls, cell phones, PDAs). The tool is composed of a set of inexpensive, wireless, physical input components (buttons, sliders, etc.) that allow, by communicating with Macromedia Director, interaction with a prototype of the interface on the screen (see Figure 1) . In this first version of the tool, each component has pins on its backside by which it can be attached to a foam model (or a model made of other soft materials). This attachment mechanism makes it easy to attach, detach, and reposition components on the model and also maintains the designer's ability to modify the form while the components are attached. A non-functional prototype together with a set of these input components and a Director interface becomes a functional interactive physical prototype without the need for any special technical knowledge. We named this tool *Switcharoo* to reflect both the underlying technology of this tool and the designer's ability to fluidly explore different component layouts.

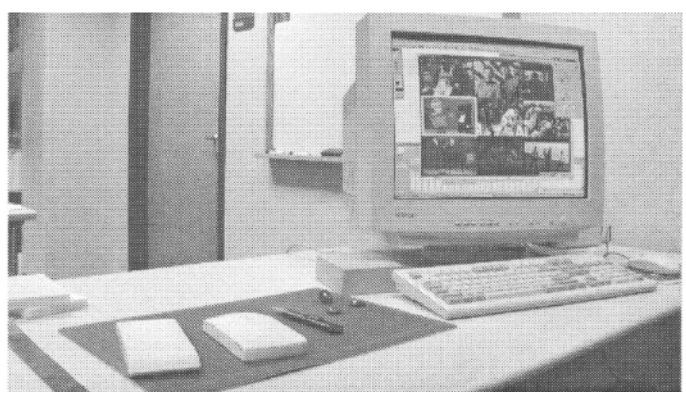
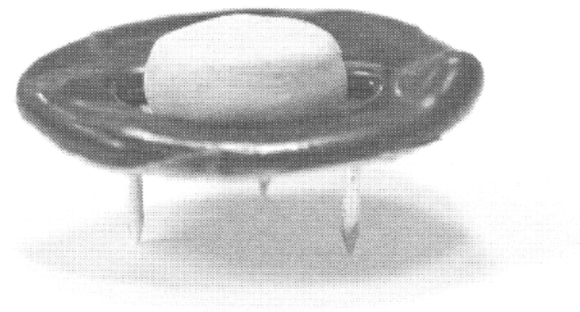


Figure 1: Switcharoo tools: blue foam, X-ACTO knife, input components, antenna, computer and Macromedia Director

This tool is:

- **A design exploration tool**: allowing designers to explore the form and interactivity of the product simultaneously,
- **A design-testing tool**: helping designers identify problems that may emerge from the combination of form and interactivity,
- **A data gathering tool**: allowing users to better express their needs, ideas, frustrations and desires, and
- **A communication tool**: improving clients' understanding of the proposed design, resulting in greater involvement and support for the design.



Background

In the last quarter of the 20th century, design practice has gone through major changes resulting from scientific and technological advancements, increased consumer awareness, the dissolution of mass markets and the emergence of niche markets. One of the main changes was the shift in focus placing the consumer, not the product, in the center of the design process. This approach, known as user-focused or user-centered design, requires that products satisfy consumers' needs, goals and desires. User-centered criterion for a successful product is often described in terms of a balance that needs to be maintained between the different aspects of the product, such as Usefulness, Usability and Desirability (Sanders 1992), or Technology and Style (Cagan and Vogel 2001). Practicing user-centered design means being involved with the users continuously throughout the design process: from gathering user data, through development and testing, to deployment and post-testing. An early framework for user-centered design included the following key principles: early focus on users and tasks, prototyping and user-testing, and iterative design (Gould and Lewis 1985). These key principles are still at the heart of many user-centered processes and were a big part in the motivation for the creation of this tool. We shall now describe a number of concepts, methods and techniques that are used in user-centered design and are of particular relevance to the four uses of our tool that were listed above.

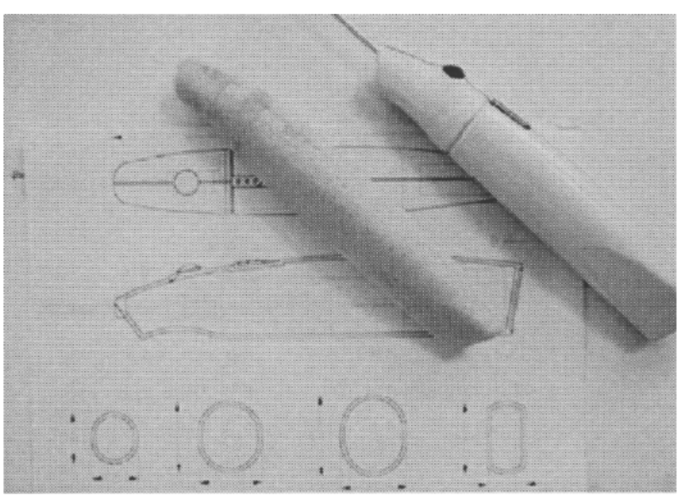


Figure 2: Sketch, model and prototype of a medical device. Whether low- or high-fidelity, they play an important role in the design process. (Drach-Ganchrow Design)

1. Rapid prototyping

One of the most common and important concepts in user-centered design is that of rapid prototyping, referring to the quick generation of prototypes in the early stages of design. Rapid prototypes are normally used for exploring the design space, are used to test basic interaction, elicit early reactions from users and for identifying problems when changes are still cheap to make (Rettig 1994). They are typically of low fidelity (e.g. - rough, made of cheap materials) and usually of limited scope (i.e. - look at specific aspects of the product or explore partial solutions). While high fidelity prototypes are very similar to the finished product (e.g. – in look, weight, material, or functionality), there is evidence that low fidelity prototypes are as successful in finding design issues as high fidelity prototypes (Virzi, Sokolov and Karis 1996). Figure 2 shows an example of the transition in the prototyping of a product from low to high fidelity.

2. User testing and iterative design

The principal of user testing is key to successful user-centered design. User testing should begin as early as possible (using rapid prototypes, as described above) and continue throughout the design process. During user testing, it is inevitable that problems in the design will be found. After these problems are fixed, user testing should be conducted again. This iterative cycle - design, test and measure, and redesign - has been shown to improve systems from many different domains.

3. Gathering user data

Gathering user data, often referred to as "Requirements Analysis", is the first step in the design process, aimed at understanding who the users of the product will be, what they need, want, think, and dream. Many techniques are used for this purpose: interviews, observations, focus groups, and so on. Velcro Modeling (Sanders 1992) is an example of a data gathering technique that emerged from the need to provide users with a way to communicate in a three-dimensional form.

4. Clients as design-team members

Realizing that the design and the process that was taken to reach it would not succeed unless understood and valued by the clients, many design studios make an active effort to bring the clients into the design process. One way to reach this goal is to integrate the clients into the design team and to view them as fully participating members. One designer told us that he would establish a relationship where he "works **with** the clients, not **for** the clients". Participatory design sessions with the clients, joint user-data analysis sessions, etc., all assist in achieving this goal. Some recommend involving a small representative group of the different stakeholders (e.g. - clients, technologists and users) in these types of sessions (Karat, Dayton 1995).

While the process of designing interactive products can employ most, if not all of the methods described above, a set of issues arise from a divide between exploration of form and exploration of interactivity. While different design firms employ different design processes, aspects of form often precede aspects of interactivity. Interactive prototypes that reflect both the form and the interactivity are typically fewer in number than prototypes that reflect only one or the other. The main cause for this divide is that the creation of working prototypes is typically expensive, yet the components are often not reusable for other prototypes. Working prototypes take a long time to manufacture and require specialized skills and tools. While some studios may have an in-house hardware shop with skilled electrical engineers, other studios are usually left with the option to have the prototyping outsourced, extending the prototyping cycle even longer.

There are a number of implications of this divide on the designer's ability to employ a user-centered design process:

As discussed before, the goal of iterative design is to try and catch as many problems as possible at every stage of the design. With fewer iterations of interactive prototypes however, fewer problems can be found and fewer improvements can take place. Furthermore, testing of interactive prototypes starting late in the process, makes it less likely that significant improvements and changes can take place even if testing suggests that they are necessary.

While a non-functional prototype can tell a lot about the physical aspects of the product, the ability to envision the interactivity of a product from a non-interactive prototype is not possessed by all. This creates two problems. The first problem is that responses from users regarding the interactivity of the product tend to be speculative. The second problem is that if clients are unable to fully understand the design and the interaction, they may not actively participate in the design process, risking their appreciation of the design.

Finally, with interactive physical prototypes not available early in the process, designers often resort to software prototypes and test the interaction using a mouse and keyboard. While an on-screen prototype of a physical product can tell us a lot about the interface, many problems cannot be revealed through the interaction with a mouse and keyboard. For example, in a project that was done in our department several years ago, an interface for a graphing calculator was tested on the screen. The users, interacting with the interface using the mouse and keyboard, repeatedly tried clicking on the screen of the calculator instead of on its buttons. When testing with an actual working prototype finally begun, a large number of issues were discovered that were not uncovered using the software-only prototype.

It is our belief that the tool proposed here can significantly reduce the divide between exploration of form and exploration of interactivity and by that alleviate the issues that arise from this divide.

Illustrations of Use

Let us give a few examples of how our tool might be used:

1. A design exploration tool

After creating a number of paper sketches of the form and interface of a new handheld device, the designer creates a set of 3D sketches made of foam and a Director software prototype of the interface. In the next stage the designer lays out Switcharoo input components on the foam sketches using them to interact with the interface. Thinking that three buttons might be too far apart, the designer quickly re-positions the components and tries out the new layout.

2. A design-testing tool

Testing a new design for a remote control, a number of users are brought into the studio to try out the new design. The users are asked to use foam models with Switcharoo components to interact with a television interface simulated on a computer (see Figure 3a). One of the users complains that the remote control

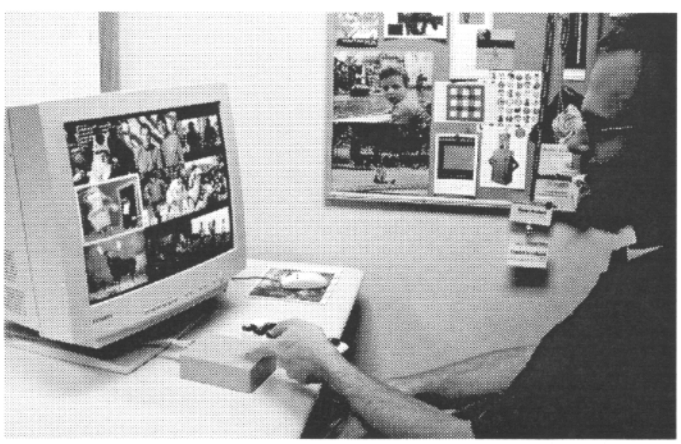


Figure 3: Illustrations of use: (a) Remote-control interaction (b) Game-controller interaction

is too wide. Using an X-ACTO knife the designer modifies the prototype and asks the user for his response. All that is done without the need to remove the Switcharoo components from the prototype.

3. A data gathering tool

A studio was hired to design a game controller for the next Macintosh computer. The designers create a number of rough game controller-shaped models and a simple game simulation in Director. Three teenagers are invited to choose Switcharoo components and place them as they wish on the models. The teenagers then interact with the simulation using their prototypes and discuss their choices and desires with the designers (see Figure 3b).

4. A tool for communicating designs to clients

The design team is holding a meeting with the clients to decide between two possible directions for the design of a medical tool. The team presents high quality sketches and Stereolithography models. Making sure that the clients understand the two proposed designs before making a decision, the team invites the clients to interact with a software simulation using two interactive prototypes that contain Switcharoo components. The clients choose one of the designs, feeling confident that the design not only looks good, but is also intuitive to use.

These illustrations demonstrate how our tool integrates common design techniques with new technology to allow a fluid performance of different user-centered design tasks.

Related Work

This work is part of a growing body of tools created to assist designers in their process. SILK (Landay and Myers 1995) is a tool aiming to address similar issues to the ones mentioned above in the domain of graphical user interface design. In SILK, designers sketch interfaces on a computer and through automatic widget recognition, these sketches become interactive (see also DENIM from Lin, Newman, Hong, and Landay 2000). Similarly to SILK, our tool attempts to provide designers with the ability to work in the fluid realm of sketches (3D sketches in our case), while making those sketches interactive. Another related system presented recently is the Phidgets toolkit (Greenberg and Fitchett 2001) providing developers with a set of input and output components conceptually similar to those discussed here. By wiring up the components to a PC (using a USB connection), these components can communicate with Visual Basic to create functional interfaces.

We shall now describe the process that we followed in designing this tool, then describe the technology used in this tool (both hardware and software) and finish with a few conclusions and plans for evaluation and future work.

Tool Design

In order to build a successful tool that will fit into a work process of a particular audience, one must first understand that process, the existing knowledge, and the culture of the audience. Realizing the limitation of our knowledge of Product Design process and that of interactive products in particular, we started the process of designing this tool with a series of interviews and conversations with professional product designers and industrial-design professors from: Carnegie Mellon University, Bezalel Academy of Art and Design Jerusalem, and The University of Art and Design Helsinki. In these interviews we learned about the work process these designers follow, the materials they use, and their relationships with clients and users. We then engaged in discussions with the designers about how a tool, such as the one we present here, could be incorporated into the process in the most beneficial way. Next we observed a number of industrial design classes in order to better understand the available tools and the materials used. During the design of the tool we also continuously involved four designers, who provided us with useful comments and suggestions. These interviews, observations, conversations, the ongoing participation, and our own familiarity with the culture of the design community (mostly from the interaction design perspective) all influenced the design of this tool. We will now elaborate on the main design decisions that were influenced by our initial study.

1. Prototyping materials

Different materials are used for prototyping at different stages of the process. As the process progresses, a smaller number of prototypes is generated using finer materials and techniques. At the early stages of the process, sketching on paper and foam modeling emerged as the most common techniques for exploring different options and aspects of the form (one of the designers referred to foam modeling as "sketches of form" and another as "3D sketches"). Our tool had to support the ease, comfort, and speed with which designers create sketches of form. This observation motivated us to choose foam as the primary supported prototyping material (although the tool can be used with other materials such as clay or wood).

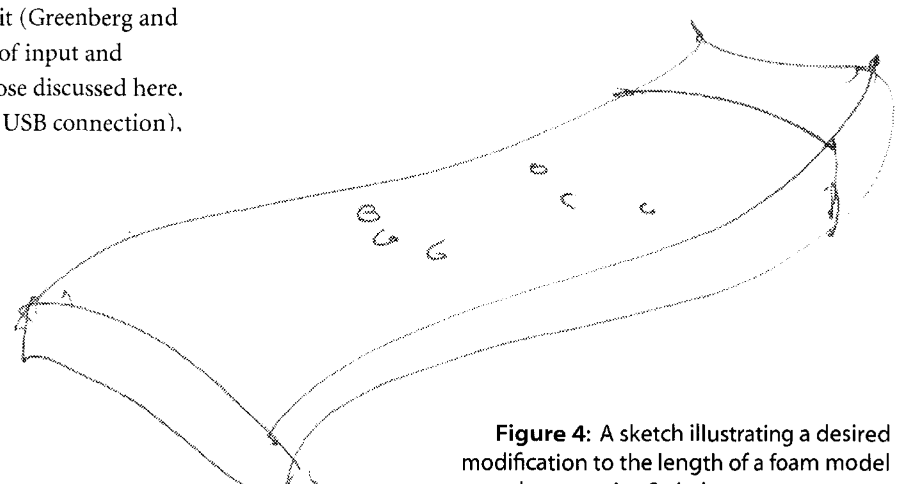


Figure 4: A sketch illustrating a desired modification to the length of a foam model that contains Switcharoo components

2. Input components

We realized that in order to match the rapid nature of form sketching, our input components had to be small, simple, and support different input techniques. We also came to the conclusion that the input components should be wireless, particularly at the "low fidelity" prototyping phase. This is important so that the technology does not overpower the form and interfere with the experience of the interaction. Following these considerations we selected a "passive RF" technology for the Switcharoo input components (a detailed description of the technology is provided later). This means that the input components do not require batteries or other sources of power allowing them to be small, wireless, and very simple to use. In this first version of the tool we provide buttons, sliders and a five-way joypad. Realizing that there are a large number of additional input techniques that also need to be supported, we plan to provide an extended set of input components with the next version of the tool.

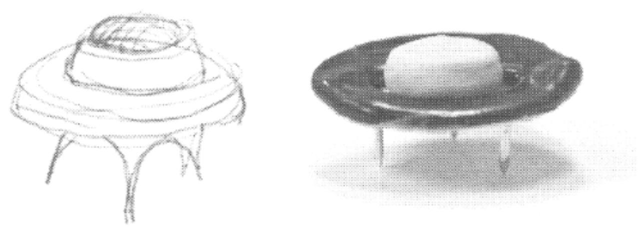


Figure 5: A sketch of, and an actual Switcharoo button with a 3-pin attachment mechanism

3. Attachment mechanism

The attachment mechanism of the input components to the form prototype had to be simple but also secure enough to allow the prototype to be used. Originally we had considered using Velcro for attaching the input components to the prototype (similar to the technique used in Velcro modeling). This was to be done by covering the prototype with the loop side of the Velcro and the input components with the hook side. While this technique supports rapid exploration of different input component layouts and keeps the input components secure to the prototype, it prevents the designer from easily modifying the form once it is covered with the Velcro. The designer's ability to modify and transform the form even when using our tool came up in the interviews as crucial for the success of the tool - currently a designer can, for example, present a foam model to a user and based on the user's comments, modify the prototype using a knife in a matter of seconds. Figure 4 shows a sketch drawn by one of the designers we interviewed, illustrating a desired modification to a foam model containing input components. This led us to abandon Velcro and instead use pins as the attachment mechanism. By using pins we maintain the designer's ability to modify the form even while the components are attached to it. Similar to Velcro, pins allow the designer to quickly attach, detach, and re-position the components while providing a secure attachment to the prototype. Following a suggestion from one of the designers, we use 3 pins on the back of every component to achieve a secure connection and prevent rotation. Figure 5 shows an actual component next to a sketch of the component with the attachment mechanism as was originally proposed by the designer.

4. Software component

One design goal that we understood to be critical for the success of this tool was for the software component not to become a hurdle or an obstacle. We knew that the software environment should be one that designers are familiar and feel comfortable with. From our daily interaction with designers we identified Macromedia Director as one of the software prototyping environments designers felt most comfortable with. We also knew that a very large proportion of the design community uses Mac OS and not Microsoft Windows. Following these observations we based our software entirely on Macromedia Director and Java, which can both work on a Mac as well as on a PC. The Java component only supports the operation of the tool; the designers need not interact with it. (A detailed description of the software component is provided later.)

Underlying Technology

In this section we describe the technology used in the input components, the creation of composite components, and the software component.

1. Input components technology

The input components provided with this tool use passive RFID (Radio Frequency Identification) tag technology. "Passive RF" means that a tag, or a button in this case, does not have its own power source such as a battery or power cable. Instead, a transmitting antenna connected to a PC or a Mac is used (in this version of the tool we used an off the shelf short-range antenna from InterSoft with a range of approximately 10cm). When a passive RF tag is placed within the antenna's range, a physical phenomenon called "inductive coupling" occurs. This phenomenon means that the radio waves sent from the antenna generate power in the tag (a version of this technology is used in retail stores as an anti-theft system with the antenna positioned by the doors). By drawing different levels of energy over time, the tag can send its ID to the computer. Since passive RF tags do not require power, they can be made very small. In this version of the tool we used RF tags that were custom manufactured by InterSoft and were relatively inexpensive. All of these custom RF tags include a switch. The antenna will only power a switched tag when the tag is within range and the switch is closed. This, for instance, allows us to know when a button was pressed within the range of the antenna. For a different technique using the pressing of a button to generate power, see Paradiso and Feldmeier 2001.

2. Software component technology

As we described before, the input components provided with Switcharoo allow interaction with Macromedia Director. Here are the steps that take place allowing this interaction: When a component is activated (e.g. - a button pressed) within the range of the antenna, the component's ID is sent from the antenna to a Java server. The server then passes this ID on to the running Macromedia Director movie. A custom lingo event *'on tagReceived'* (similar to an *'on mouseUp'* event) is then generated with the component's ID as an argument. In order to set up a Director movie that the components can communicate with, the designer needs simply include one Script member that we have written (allowing Director to generate the *'on tagReceived'* events). After that, the designer simply creates regular frame scripts and cast-member scripts in the movie to handle the incoming events as they would with other Director events. It is important to note that the designer can assign more than one input component to do the same operation, allowing the creation of several prototypes that interact with the same interface.

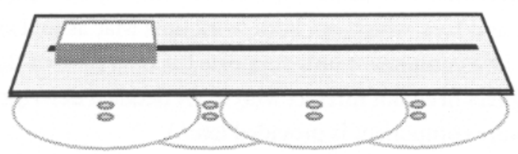


Figure 6: A diagram of a slider composed of 4 switched RF tags

3. Composite input components

While switched RF tags can only show two states (for example, pressed or not pressed), by using a number of tags together we can simulate more complex input techniques such as sliders, knobs, thumb-wheels, etc. Figure 6 shows a diagram of how we construct a slider out of four switched tags. Even though the simulated slider consists of only four states, it should be sufficient for simulating the interaction. Figure 7 shows a diagram of a five-way joypad (a similar interaction technique to that found on a Compaq iPAQ).

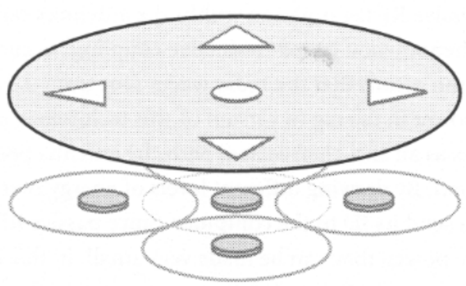


Figure 7: A diagram of a 5-way joypad composed of 5 switched RF tags with buttons

Conclusions and Future work

We have presented the first version of Switcharoo, a tool that tries to alleviate the issues that arise from the divide between exploration of form and exploration of interactivity in the design of physical interactive products. Basing our tool on common design techniques and skills, we provide designers with the ability to create advanced prototypes without any special technical knowledge.

While we received very positive informal feedback as to the desirability of the tool, its usability and usefulness need to be evaluated. We recently received offers from three product design professors at Carnegie Mellon University to give the tool to their students to use as part of their courses. Feedback provided by these students will be invaluable for the improvement of the tool. Our other plans involve improvements to the technology that was used in this version of the tool; we plan to obtain a stronger antenna with a longer range, create smaller input components, and support a larger set of input techniques by creating more composite components (such as thumb-wheels, jog-dials, etc.). Finally we plan to support more prototyping materials using a larger set of attachment mechanisms.

Acknowledgements

We would like to thank Joe Ballay, Bruce Hanington, Jodi Forlizzi, Francine Gemperle, Carl DiSalvo and Eric Anderson from Carnegie Mellon University, Ami Drach from Drach-Ganchrow Design and Bezalel Academy of Art and Design Jerusalem, and Katja Battarbee from The University of Art and Design Helsinki. This work was supported in part by the National Science Foundation under grants IIS-9800597 and IIS-0121560.

References

Cagan, J and Vogel, C.M. (2001) *Creating Breakthrough Products: Innovation from Product Planning to Program Approval*, Financial Times - Prentice Hall Publishing

Gould, J.D. and Lewis, C. (1983) "Designing for Usability: Key Principles and What Designers Think" *Proceedings of CHI '83*, pp. 50-53

Greenberg, S. and Fitchett, C. (2001) "Phidgets: Easy Development of Physical Interfaces through Physical Widgets" *Proceedings of UIST '01*

Karat, J. and Dayton, T (1995) "Practical education for improving software usability" *Proceedings of CHI '95*

Landay, J.A. and Myers, B.A. (1995) "Interactive Sketching for the Early Stages of User Interface Design" *Proceedings of CHI '95*

Lin, J., Newman, M.W., Hong, J.I. and Landay, J.A. (2000) "DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Designä" *Proceedings of CHI '00*, pp. 510-517.

Paradiso, J. and Feldmeier, M. (2001) "A Compact, Wireless, Self-Powered Pushbutton Controller" *Proceedings of UbiComp '01*, pp. 299-303

Rettig, M. (1994) "Prototyping for tiny fingers" *Communications of the ACM*, April 1994, Volume 37 Issue 4

Sanders, E.B.-N. (1992) "Converging Perspectives: Product Development Research for the 1990s" *Design Management Journal*, Fall 1992.

Virzi, R.A., Sokolov, J.L. and Karis, D. (1996) "Usability Problem Identification Using Both Low- and High-Fidelity Prototypes" *Proceedings of CHI '96*, pp. 236-43