Usability: Gaining a Competetive Edge
IFIP World Computer Congress 2002
J. Hammond, T. Gross, J. Wesson (Eds)
Published by Kluwer Academic Publishers

A Framework for Rapid Mid-Fidelity Prototyping of Web Sites

Daniel Engelberg and Ahmed Seffah
Human-Centred Software Engineering Group, Concordia University, Canada
dan.engelberg@sympatico.ca; seffah@cs.concordia.ca

Abstract: This paper presents a prototyping framework situated mid-way between low fidelity and high fidelity. The framework is used after requirements definition and early design but before development. The approach provides a solution to the classical trade-off between the ease of production associated with low-fidelity approaches and the realism associated with high-fidelity techniques. Within this framework, we present a generic mid-fidelity prototyping method supported by a tool, MS-PowerPoint, which we have found well adapted for mid-fidelity prototyping.

Key words: rapid prototyping, PowerPoint, low fidelity, medium fidelity, high fidelity, Internet, web sites, user-centred design.

1. MOTIVATION

Our purpose in presenting this paper is to share a rapid prototyping method we initially developed for our own use, over the course of about 5 years of application in industrial projects. The framework grew from a desire to prototype the interactive and navigational aspects of user interfaces as quickly as possible, with a minimum investment of learning time and with no programming skills. This is not the case with many current prototyping tools. It was important for us that our prototype would have high enough fidelity to clearly communicate the details of the interface to clients, users and developers.

2. LOW, MID AND HIGH FIDELITY TOOLS

Only two levels of fidelity of rapid prototypes are commonly recognized in the literature: low fidelity and high fidelity (e.g. Isensee & Rudd, 1996; Virzi et al., 1996). Unfortunately, low and high fidelity are loosely defined and each covers a broad range of fidelity levels, leading to confusion in comparing different tools. For example in the traditional classification, two different "low-fidelity" tools could create prototypes of radically different fidelity.

Therefore, we have found it useful to define an intermediate "mid-fidelity" category, which has properties distinct from the two extremes. There are no formal references in the literature to this intermediate category, apart from two studies referring to the use of medium-fidelity prototypes for flight simulators and pilot training (Metalis, 1993; Becker & Hamerman-Matsumoto, 1989).

Table 1 summarizes the characteristics of these three proposed fidelity levels, which represent a continuum from low to high.

Fidelity	Appearance	Optimal uses	Advantages	Limitations
Low	Rough sketch; highly schematic and approximate. Little or no interactive functionality.	Early design: conceptualizing and envisioning the application.	Low cost: useful communication vehicle; proof of concept.	Limited usefulness after requirements established; limitations in usability testing
Mid	Fairly detailed and complete but objects are presented in schematic or approximate form. Provides simulated interactive functionality and full navigation.	Designing and evaluating most interactive aspects, including navigation, functionality, content, layout and terminology.	Much lower cost and time as compared to high fidelity; detail is sufficient for usability testing; serves as a reference for the functional specification.	Does not fully communicate the look and feel of the final product; some limitations as a specification document.
High	Lifelike simulation of the final product; refined graphic design. Highly functional, but the back end might be simulated rather than real.	Marketing tool; training tool; simulation of advanced or highly interactive techniques.	High degree of functionality; fully interactive; defines look and feel of final product; serves as a living specification.	Expensive to develop; time consuming to build.

Table 1: Comparison of different levels of prototype fidelity

In our three-level framework:

- Low-fidelity prototyping tools and methods are used for early design just after requirements analysis, to help conceptualize and envision the interface at a high level. These tools often support rough sketching of interface screens by freehand drawing with a mouse or tablet pen.
- Mid-fidelity prototyping tools are used after early design, for the purposes of detailed design and usability validation. They present detailed information about navigation, functionality, content and layout, but in schematic ("wireframe") or approximate form.
- High-fidelity prototyping tools permit the creation of a lifelike simulation, normally for marketing purposes or sometimes for user tests, before the final version has been developed. High-fidelity prototyping tools tend to target developers, and are often general-purpose development tools. Due to the efforts required, high-fidelity prototypes are usually not "rapid"; nevertheless the expression RAD (rapid application development) is widely used in the field.

A re-classification of existing tools into our proposed three levels of fidelity is beyond the scope and purpose of the current paper. Therefore in the following overview of existing tools, for the moment we wish to sidestep the issue of whether they are low, mid or high fidelity in the new framework. We therefore refer to existing tools under their originally defined fidelity level in the two-level system.

Under the original two-category framework, low-fidelity tools proposed for the early design of graphical interfaces have ranged from simple pen and paper (Landey, 1995; Rettig, 1994) to scripting-based prototyping tools such as Visual Basic and HTML. SILK (Landay & Myers, 1995) and PatchWork (van de Kant et al., 1998) have been suggested as intermediate solutions for early design. Based on the SILK approach, DENIM offers another dedicated tool for the early design of Web sites (Lin et al., 2000).

For high-fidelity prototyping of Web sites, Jung and Winter (1998) summarize CASE (Computer Aided Software Design) tools, including Microsoft FrontPage, Adobe PageMill, Aimtech's Jamba, W3DT, and application development tools such as Sun Microsystem's Java Workshop, Microsoft's Visual J++, Symantec's Visual Café, Rational Rose and Oracle Designer. In addition, Graphical User Interface (GUI) builders such as Visual Basic and Borland Delphi, and their associated user interface toolkits such as MFC and JDK, provide direct manipulation for visually combining and organizing the basic GUI widgets onto screens and windows. Although very powerful, these high-fidelity tools require considerable learning time and technical support, and offer considerably more functionality and detail than necessary for the purposes of usable design.

As explained by Myers et al. (2000), it is difficult to find a user interface software tool which is both easy to learn and highly functional. Tools that are easy to learn and use tend to have basic functionality and low versatility, and tools offering advanced functionality and high versatility tend to be hard to master and use effectively.

Our solution to this trade-off lies in the use of mid-fidelity tools such as PowerPoint. A well-designed mid-fidelity tool can simultaneously offer ease of production and versatility, by focusing on what is essential for prototyping a usable design, and by supporting detailed design through direct manipulation without programming.

3. PROTOTYPING METHODS

There is little detailed reference to any structured methodology for user interface prototyping in the literature. This lack of detail is probably in part because most design methodologies are still in their early stages, and perhaps also because prototyping methods are seen as contextual to the tool. As explained by Isensee and Rudd (1996) and still applicable today, "…the optimum methods of prototyping have not yet been agreed on." One of our motivations for this paper was therefore to document a generalizable prototyping method.

Wilson and Rosenberg (1988) were among the first to define the major classes of prototyping method. Their main categories are the top down approach and the bottom up approach.

Top down approaches define the interface progressively in increasing levels of detail, starting with the task analysis and conceptual design and proceeding through to the detailed design. These approaches are especially appropriate for projects where the functional requirements are known in advance, which should be the case with most user-centred approaches.

Bottom up approaches are not strongly based on a needs or task analysis, and are therefore more highly iterative than top-down approaches. These approaches start by creating a "best guess" design and then by progressively changing and refining that design until it meets the usability requirements. This trial-and-error approach is appropriate when it is impossible to clearly know the functional requirements in advance. Although this context is rare, it could occur in the design of a radically new tool that supports a task that does not yet exist.

Our own method, described in Section 5, is largely top-down, but incorporates iterative loops allowing for bottom-up feedback.

4. BENEFITS AND VALIDITY OF RAPID PROTOTYPING

Rapid prototyping is known to offer many advantages across all levels of fidelity. The following summarizes the most important benefits of rapid prototyping (Isensee & Rudd, 1996; Rudd, Stern & Isensee, 1996; Wilson & Rosenberg, 1988):

- Cost savings in the total life cycle
- Improved usability and quality of the final application
- Permits usability testing before coding
- Improved communication of the design concept to the client and end-users, and communication of the functional specifications to graphic designers and developers.

In several controlled experimental studies (Virzi et al., 1996; Wicklund et al., 1992; Virzi, 1989), it was found that “low-fidelity” prototypes (probably mid-fidelity according to our classification scheme) are as effective as high-fidelity prototypes for validation with users. In other words, rapid prototypes are able to capture almost all of the usability issues that could be communicated in a fully developed system.

5. THE GENERIC MID-FIDELITY PROTOTYPING FRAMEWORK

In this section we present our generic rapid prototyping method for mid-fidelity prototypes. The generic method is presented separately from its application using PowerPoint. This separation allows the method and tools to be evaluated separately and facilitates reuse of the method with other tools.

The method is used after sketching a preliminary design or vision. The method itself does not provide a cookbook recipe for usable design; rather it provides a framework within which designers can easily create usable prototypes, assuming some prior familiarity with the principles of usability. The main steps of the method are described in Figure 1.

The current description uses menu-based architectures as a typical design assumption, since this is the most common type of system designed today for the Web.

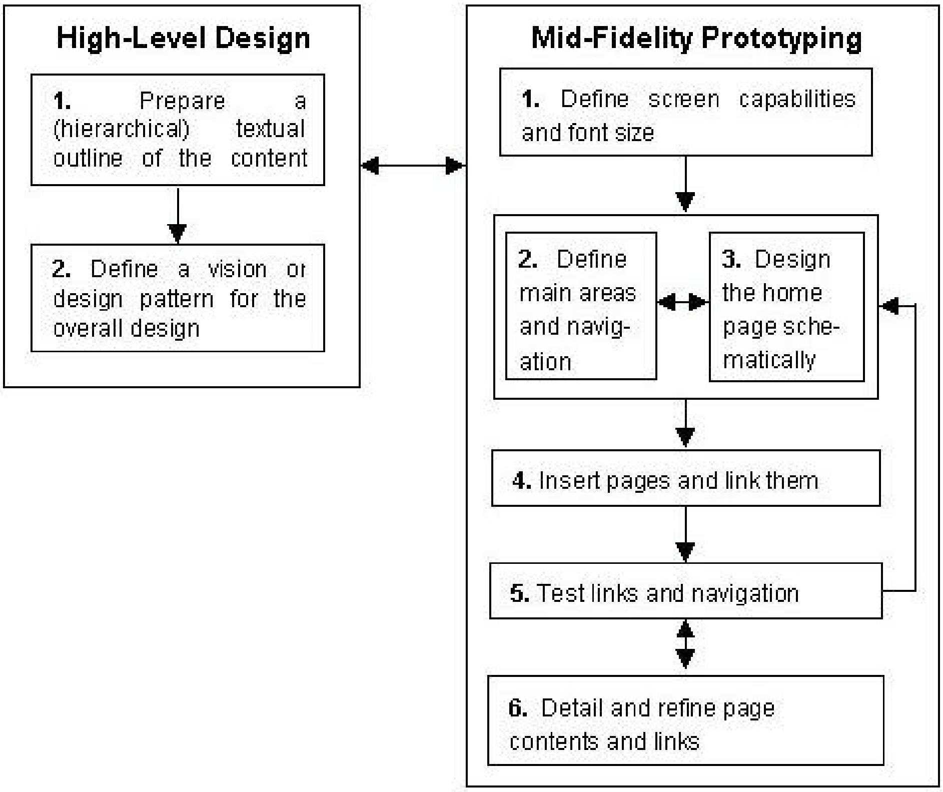
High-Level Design
1. Prepare a (hierarchical) textual outline of the content
2. Define a vision or design pattern for the overall design
Mid-Fidelity Prototyping
1. Define screen capabilities and font size
2. Define main areas and navig-ation
3. Design the home page sche-matically
4. Insert pages and link them
5. Test links and navigation
6. Detail and refine page contents and links

Figure 1: Main steps of the method

Before starting to create the mid-fidelity prototype, it is important to prepare a preliminary (high-level) design. This stage of design is described by Torres (2002) as "Conceptual design and architecture". Early design tools such as Patchwork and DENIM can be very useful at this stage, but pen and paper sketches are adequate. Two steps are involved:

1. A hierarchical (tree-format) outline of the content and functionality of the information architecture is created (Rosenfeld & Morville, 1998). The outline serves as a reference to help structure the early stages of defining menus, navigational structure and page content. The outline accelerates the graphical prototyping and ensures that crucial elements are not forgotten.
2. A rough overall design is sketched, including the general navigation scheme (Mayhew, 1999; Torres, 2002). Solutions can be drawn from design patterns in order to accelerate this step (Borchers, 2001). This step is inherent to usable design but is considerably more complex than can be explained in this paper.

5.1 Define screen capabilities and font size

The prototype should simulate the screen resolution and screen size at which the final site will be viewed. Some prototyping tools allow the user to set the screen resolution, but in other tools it is necessary to approximate font size by “eyeballing” (trial and error tests) in full-screen mode. If the prototyping tool does not allow the designer to specify screen resolution, the use of a smaller font will simulate the increased space available at higher screen resolutions.

The font size affects how much space a menu requires, how much information will fit into the content zone, and how many top-level items can fit into a horizontal menu. The main fonts to define are for menu items, page titles, subtitles, textual content, field labels and buttons.

5.2 Define main areas and navigation strategy

This step is performed in parallel with the definition of the home page. Figure 2 provides an example.

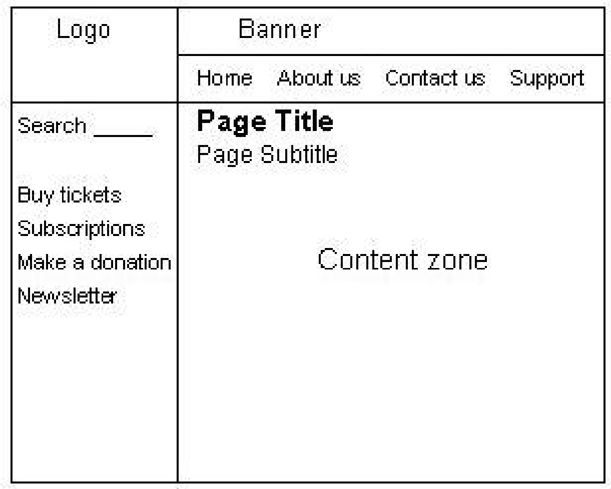

Figure 2: An example of definition of main zones and navigation approach

This step consists of defining the page layout and basic behaviour, in particular for the elements listed in Table 2. In a first step, the designer places repeating elements (such as menus) in a dedicated and normally fixed-size zone. Some prototyping tools such as PowerPoint have a “page background” or “master” function that allows users to identify elements that repeat on all pages.

The general navigation approach (menus and any other tools) should be defined at this point, by referring to the architecture and navigation needs described in the textual outline prepared in the first step. Normally menus are placed horizontally at the top and/or vertically at the left.

Finally, the menu content is defined (i.e. the specific items to be used in the menus), by referring to the textual outline.

Graphical element (zone)	Main properties to define
Repeating header	Location, dimensions
Menus (horizontal and vertical)	Location, dimensions, number of embedded levels, behaviour, font
Main content area	Location, dimensions, font
Adjunct content areas	Location, dimensions, font
Promotional information areas	Location, dimensions, font
Additional navigation tools	Location, dimensions, behaviour
Page title	Location, dimensions, font
Page subtitle (if relevant)	Location, dimensions, font

Table 2: Main graphical elements to define

5.3 Design the home page schematically

This step is performed in parallel with defining the main zones and the navigation approach. It consists of inserting a page into the prototype for the home page, and defining the page schematically, in particular for the menu and navigation system.

5.4 Insert pages and link them

Firstly, a destination page in the prototype is inserted for each main link on the home page (e.g. menu elements), and for linked items on secondary pages. The most important thing at this point is to place a title on the secondary page immediately as a reminder of its purpose. The blank page is mainly a placeholder for linking to the page from hyperlinks elsewhere in the prototype. It is not yet necessary to start designing the detailed contents of each page.

Secondly, if not performed automatically by the prototyping tool, the menu defined on the home page can be added to each secondary page, opening the appropriate configuration of sub-levels as needed. Some secondary pages might of course use menus that are different from the home page.

Thirdly, if not performed automatically by the prototyping tool, menu items can be provided with hyperlinks and other links to the target pages.

Finally, schematic content is added to the pages. At this point it can be sufficient to indicate the main zones by drawing text boxes or rectangles with descriptive titles.

5.5 Test links and navigation

After a navigation structure has been created, the prototype is tested informally by navigating with the hyperlinks through typical paths. This step helps the designer determine whether the navigation is fluid and efficient. Afterward, links and pages can be changed or added if necessary.

Since the design process is iterative, it is possible that at this step, or at any other step, issues will arise that will require reconsidering the general approach or some specific element of the design. If so, it is possible to return to this previous design step and loop back through the same sequence of procedures, adjusting the prototype as necessary.

5.6 Detail and refine the page contents

Once the "skeleton" or navigational framework of the design has been completed in the previous steps, the page contents and interaction are refined. In forms and interactive dialogues, field labels and sample field are also added. Graphics may also be useful. Objects can be aligned at any stage. This is the time to fine-tune the alignment of objects and zones. See Figure 3 for an example of the results of this step.

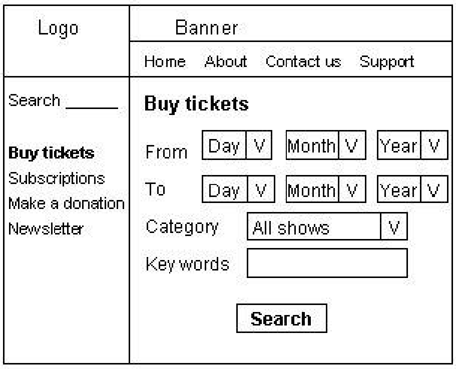

Figure 3: An example of detailed contents

5.7 General techniques

The following additional methods apply across all of the prototyping steps:

- Because the goal of a mid-fidelity prototype is simply to communicate the concept and create the illusion of the target system, it is usually not essential to model every single page in the target site. Particularly for large sites, the designer can put less detail in the less important sections and pages. Where the same page format repeats across a large number of pages, a single page can be provided as an example.
- It is possible to simulate virtually any interaction technique using simple approaches. For example, complex database architectures can be simulated simply by providing sample data in the fields. The prototype does not need to actually use advanced technology to present advanced techniques – it only needs to paint the picture of how the results would look.
- In addition to the prototyping process itself, it is important to validate the prototype before sending it out to the developers, in the following steps:
  1. The prototype is compared to the textual or graphical outline to ensure that it correctly implements the vision and definition of needs.
  2. All links are tested.
  3. The prototype is validated with the client.
  4. Usability tests are performed with target users.

6. APPLYING THE METHOD WITH POWERPOINT

The following refinements of the generic design steps take into account technical aspects specific to PowerPoint functionality.

In the first step (Defining type fonts): PowerPoint does not inherently define a screen resolution, so the designer must adjust the font and object size to simulate different resolutions and screen sizes. To simulate a screen resolution of 800x600 pixels, a font size of about 12-14 points is used for content and menu items, and other objects are sized in proportion. To simulating a target resolution of 1024x768, the standard text size for content and menus should be about 10-12 points.

In the second step (Define main zones areas and navigation approach): Repeating elements are put in the slide master. The slide master is a valuable functionality in PowerPoint. It allows the designer to enter elements that will be repeated on all pages. Normally the master is used for the horizontal header bar at the top with the company’s logo and a repeating horizontal menu. The master can also be used for a one-level repeating vertical menu, but it cannot be used for dynamically expanding vertical menus using a simple text-link approach, because the master allows for only a single configuration of elements.

If the design uses a dynamically expanding vertical menu at left, then the menu is copied and pasted into every page, expanding different sub-sections as appropriate for the context. It is best to copy and paste a version of the menu where the menu hyperlinks have already been defined, so as to avoid having to re-program them on each page.

In designing menus, a text box is used for menu items, normally without an outline. A separate text box is used for each item, rather than putting all the menu items in a single text box. It is important to use separate text boxes so as to allow different hyperlinks for each menu item.

In the fifth step (Testing links and navigation): Links and navigation can be tested in PowerPoint by entering Slideshow mode. Links are active in this mode, but not in editing mode.

General techniques: In addition to the above step-specific techniques, we have found the following techniques to be useful:

- Repeating elements can be copied and pasted from other slides. In copying elements from one slide to another, PowerPoint automatically aligns pasted elements to the same position as on the slide from which they were copied.
- Sequences of slides can be animated, using the “Slide Transition” function in the Slide Show menu, to model certain types of interactive effects.
- Collaborative approaches: PowerPoint can be used in real time in a conference room for collaborative design sessions, brainstorming, and design validation sessions.

6.1 Benefits and weaknesses of PowerPoint

Based on our experiences in industrial projects, PowerPoint provides the following benefits as a mid-fidelity prototyping tool:

- The tool takes only a few hours to learn for the purposes of prototyping, and does not require any technical or programming expertise.
- Prototypes can be developed very quickly, often close to the same amount of time that it would take to draw the screens by hand.
- The tool can simulate virtually any type of graphical interaction quickly, easily and directly.

Although PowerPoint is a very solid and versatile prototyping tool, we have found from experience that it has some weaknesses. In particular, as the prototype becomes larger, it becomes increasingly more difficult to modify globally. This is partly due to the fact that there are no global utilities for managing links. In addition, there is no inherent functionality for creating dynamic menus; therefore each different configuration of a menu needs to be drawn by hand on a different page. Finally, it is difficult to prototype pages that extend beyond one screen length. Nevertheless, the benefits of PowerPoint outweigh the weaknesses.

7. CONCLUSION

The framework for mid-fidelity prototyping described here presents a simple, easy-to-apply structured method for rapid prototyping. This framework consists of a generic method for Web application prototyping supported by the PowerPoint tool. PowerPoint fits well into the gap between low-fidelity and high fidelity prototyping tools.

Although the generic method can be applied with a broad range of prototyping tools, we have found that PowerPoint supports this process well, reducing the possibility of the traditional tradeoff that occur between ease of use and functionality. PowerPoint is especially well adapted for rapid mid-fidelity prototyping, allowing designers to quickly translate their ideas onto the screen, while also permitting interactive content and high-quality page layout.

In future research we will test this framework empirically with usability specialists and Web designers. This will help us refine the method and define elements for improving PowerPoint’s adaptation to mid-fidelity rapid prototyping.

8. REFERENCES

Becker, C.A. & Hamerman-Matsumoto, J. (1989), The Use of Low-Cost Microcomputers as Medium-Fidelity Simulators for Prototype Development, Training and Research, in *Proceedings of the Human Factors Society 33rd Annual Meeting 1989*, v.2, 1068-1071.

Borchers, J. (2001), *A Pattern Approach to Interaction Design*, Wiley.

Isensee, S. & Rudd, J. (1996), *The Art of Rapid Prototyping*, International Thomson Computer Press.

Jung, R. & Winter, R. (1998), Case for Web Sites, in *Proceedings of the 1998 ACM Symposium on Applied Computing*, Atlanta, Georgia, United States, 726-731.

Landay, J.A. & Myers, B.A. (1995), Interactive Sketching for the Early Stages of User Interface Design, in *Proceedings of CHI '95: Human Factors in Computing Systems*, Denver, Colorado, 43-50.

Lin, J., Newman, M.W., Hong, J.I. & Landay, J.A. (2000), "DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design," *CHI Letters: Human Factors in Computing Systems, CHI 2000*, 2(1): 510-517.

Mayhew, D. (1999), *The Usability Engineering Lifecycle*, Morgan Kaufmann Publishers.

Metalis, S. A. (1993), Assessment of Pilot Situational Awareness: Measurement via Simulation, in *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting,* v.1, 113-117.

Myers, B., Hudson, S.E. & Pausch, R. (2000), Past, Present and Future of User Interface Software Tools, *ACM Transactions on Computer Human Interaction (TOCHI)*, 2000, **7**(1): 3-28.

Rettig, M. (1994), Prototyping for tiny fingers, *Communications of the ACM*, **37**(4): 21-27.

Rosenfeld, L. & Morville, P. (1998), *Information Architecture for the World Wide Web*, O'Reilly.

Rudd, J. Stern, K. & Isensee, S. (1996), Low Vs. High-Fidelity Prototyping Debate, *Interactions,* **3**(1): 76-85.

Torres, R.J. (2002), *Practitioner's Handbook for User Interface Design and Development*, Prentice Hall.

Van de Kant, M., Wilson, S., Bekker, M., Johnson, H. & Johnson, P. (1998), Patchwork - A Software Tool for Early Design, in *Proceedings of CHI '98 Conference Companion: Human Factors in Computing Systems*, Los Angeles, California, 221-222.

Virzi, R. (1989), What can you learn from a low-fidelity prototype? In *Proceedings of the Human Factors Society 33rd Annual Meeting, HFES*, Santa Monica, California, 224 - 228.

Virzi, R., Sokolov, J.L. & Karis, D. (1996), Usability Problem Identification Using Both Low- and High-Fidelity Prototypes, in *Conference on Human Factors in Computer Systems (CHI 96)*, Vancouver, Canada, 236-243.

Wiklund, M., Thurrott, C. & Dumas, J. (1992), Does the fidelity of software prototypes affect the perception of usability? In *Proceedings of the Human Factors Society 36th Annual Meeting, HFES*, Santa Monica, California, 399 - 403.

Wilson, J. & Rosenberg, D. (1988), Rapid Prototyping for User Interface Design, in M. Helander (ed.), *Handbook of Human-Computer Interaction*, North Holland, 859-873.