

# Rapid Prototyping Methods to Develop Applications for Seniors

*Young Sam Ryu*

*Tonya L. Smith-Jackson*

*Robert C. Williges*

Grado Department of Industrial and  
Systems Engineering, Virginia Tech  
Durham Hall 250  
Blacksburg, VA 24060  
ryu@vt.edu

Grado Department of Industrial and  
Systems Engineering, Virginia Tech  
Durham Hall 250  
Blacksburg, VA 24060  
smithjack@vt.edu

Grado Department of Industrial and  
Systems Engineering, Virginia Tech  
Durham Hall 250  
Blacksburg, VA 24060  
williges@vt.edu

## Abstract

This paper presents a case study of rapid prototyping for iterative prototype design and formative evaluation of computer interfaces for the development of applications for seniors. A combination of three different tools was used -- paper sketches, MS-PowerPoint, and Macromedia Flash. Each tool was used for different phases and roles in the prototyping process. Macromedia Flash served as the main prototyping tool for iterative design and formative evaluation, and our experience showed that Flash can support the rapid prototyping process by reducing the tradeoff that occurs between ease of learning and use and prototype function. The prototyping process involving senior users was found useful in detecting usability problems, especially language issues and problems with touch screen interface elements involving physical interaction. Also, in order to develop the environment for designing software for seniors, early involvement of senior users to develop conceptual models and refinement of design language to facilitate participatory are necessary.

## 1 Introduction

Winograd (1995) predicted that there will be a paradigm shift from programming environments to design environments that help developers “satisfy end-user’s cognitive needs and help deal with contextual issues such as the aesthetics, practical, and social properties of the application and the users” (p. 65). Ten years after he published that prediction, we have observed changes of the trends in software developmental environments, such as the prominence of interactive, object-oriented, and visual programming. In addition to the changes in practical aspects in the environment, the growth and popularity of the Internet has raised the end user’s accessibility to various software applications. Consequently, the broadness or diversity of the possible end-user population has increased significantly. The paradigm shift is driven by the enlargement of the end user population and the technological innovation of the development environment.

There are numerous tools and techniques for software designers and developers, but Stephanidis (2001) proclaimed that higher level tools have better chances to develop user interfaces for all kinds of users. However, those tools may not be easy to use for creating prototypes in a rapid manner. Rapid prototyping can be defined as an early development of a prototype for testing functionality, interaction, or even appearance of the product (Wilson, Jonassen, & Cole, 1993). Techniques of rapid prototyping and iterative design emerged to meet the need of evaluating potential designs early in the design process; this allows for developers to evaluate proposed designs during development “to ensure that usability gets built into the evolving interaction design” (Hix and Hartson, 1993, p. 251).

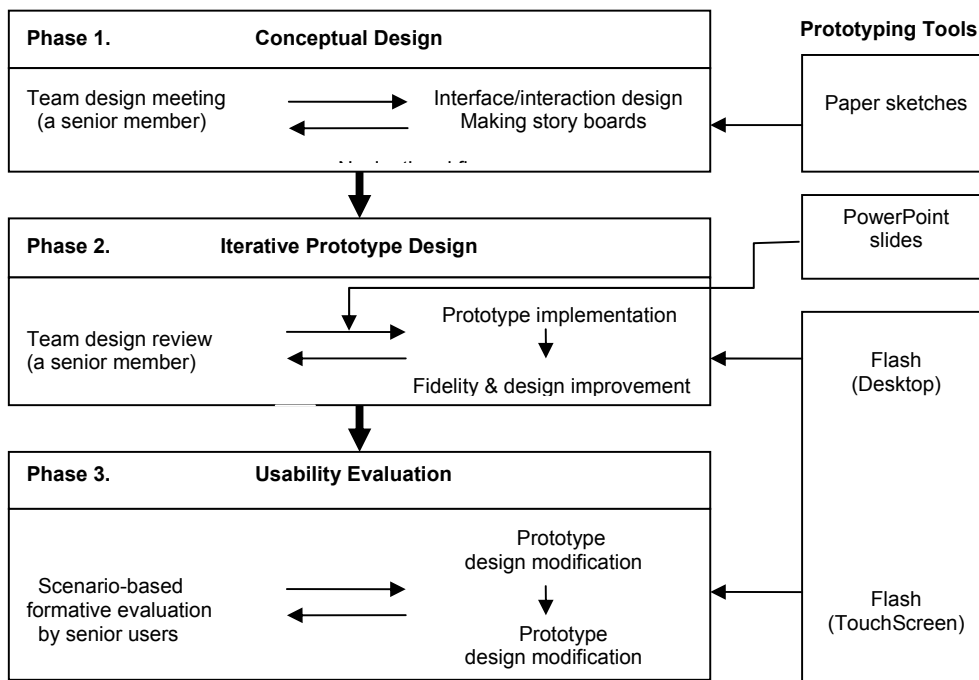
Numerous tools exist for software application developers utilizing languages such as Java, Visual C++, and Visual Basic. These powerful tools are used in the design of highly functional applications and to manipulate visual features in detail. However, because such tools are intended for targeting final products, these may not be easy to use for creating prototypes. In fact, such tools offer considerably more functionality than may be needed for usable design; considerable learning time may be required for use by software developers or designers (Engleberg & Seffah, 2002). For this reason, selecting such tools for rapid prototyping may not be efficient for people who are not highly trained in the use of these tools.

Relatively new application development software tools such as Macromedia Flash or Director are available in the software development community for developing interactive Internet applications including various formats such as text, graphics, animation, video, audio, and hyperlinks (e.g., see <http://www.macromedia.com>). For example, Flash allows both designers and developers to create animations using vector graphics across many platforms and devices (e.g., PC, Macintosh, Internet Explorer, Netscape, AOL and etc.). Also, Flash can be a powerful prototyping tool because it provides relatively easy interactive and visual authoring. This characteristic is compatible with the transition from the programming environment to the designing environment. Isensee and Rudd (1996) argued that there is no agreed upon optimum method of prototyping, as confirmed more recently by Engelberg & Seffah (2002) who pointed out that few references exist describing user interface prototyping methodologies.

Our project team understood the importance of choosing an easy-to-use rapid prototyping tool to develop user interfaces for various groups of users, including senior citizens. Macromedia Flash 5 was selected to help facilitate development efforts and to assist in the process of prototype design and formative evaluation of an information display related to a smart house application called Senior Healthwatch. This paper aims to introduce our prototyping approach and methodology to share the lessons learned from the development of the Senior Healthwatch system to deliver useful information about the health condition of older residents in a Smart House (Kwahk, Smith-Jackson, & Williges, 2002; Ryu, Williges, Smith-Jackson, & Kwahk, 2002).

## 2 Methods

Figure 1 describes the prototyping procedure along with the tools used. This procedure involves three phases including conceptual design, iterative prototype design, and formative usability evaluation.



**Figure 1:** Prototyping procedure for *Senior Healthwatch* system along with the prototyping tools

### 2.1 Phase 1: Conceptual Design

A meeting for conceptual design was conducted by the project team. The project team consisted of two professors, a post-doctoral researcher, and two graduate research assistants. For the conceptual design step, low-fidelity paper based screen sketches were used to determine organization of menus and pages, interaction type, abstract interface

design, and navigational flows. Throughout this step, the designers could get the sense of visible or tangible elements to be built into a prototype.

## 2.2 Phase 2: Iterative Prototype Design

For the next step of iterative prototype design, Macromedia Flash 5 was selected as a prototyping tool by the project team. To be familiar with the selected prototyping tool, members of the design team took a series of online self-training courses for Flash 5 provided by Virginia Tech. It took 2 hours a day for 5 days for the design team to complete 37 training modules (15 minutes for each module).

After the training, the design team began to build prototypes based on the results of the conceptual design phase. Though this iterative prototype design phase lasted three months, ten different versions of prototypes were developed and improved upon based on weekly design review meetings by the project team. Verbal descriptions were used during the meetings to communicate features to be revised among other project team members and the design team. PowerPoint slides were used to demonstrate design features to the design team precisely as the fidelity of the prototype increased. For example, color or detailed screen layouts as depicted in Figure 2 could be delivered to the design team in a more straightforward manner than by verbal description. If the fidelity of the images used in PowerPoint prototype was high enough, those images were imported into Flash prototype. As iterations of the prototyping were repeated in this phase, the level of fidelity became higher in terms of visual features and functions.

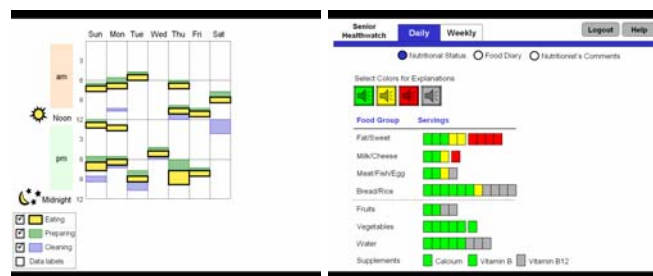


Figure 2: Examples of PowerPoint prototype as a communication vehicle

## 2.3 Phase 3: Formative Usability Evaluation



Figure 3: Formative evaluation session using a touch-enabled LCD display

Two separate sessions of formative evaluations were conducted for usability evaluation of Senior Healthwatch. To record the evaluation process, a high-quality video camera and a computer-based screen capturing tool (Camtasia Recorder 3.0.2) were used. Four members of the senior advisory board participated in the first session, and eight more seniors from the Blacksburg Electronic Village volunteered for the second session. A 17" touch-enabled LCD

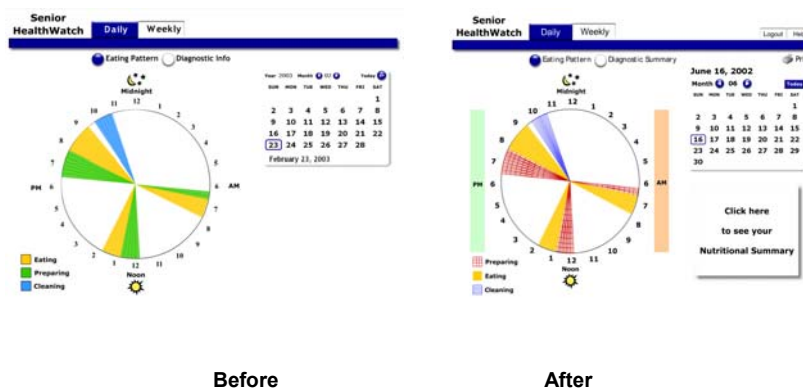
display (Planar CT1744-NUT) was used to present the prototype, which allowed the prototype to accept user inputs only through the touch screen without using a keyboard or a mouse as depicted in Figure 3. A stand-alone desktop PC was used for running the prototype. Each participant completed the same series of structured task scenarios individually followed by an interview. User comments were collected and forwarded to the design team. Based on the observations of the participants as they performed tasks and their comments from interviews, the prototype was iteratively revised through the two sessions.

### 3 Results

#### 3.1 Interface Changes Based on Prototyping

##### 3.1.1 Static interface

Existing design guidelines for website designers targeting senior users (Kwahk, Smith-Jackson, & Williges, 2001) were used to check interface elements of the prototype. Although Senior Healthwatch is not a web site but an application that uses networks and a database, the design guidelines could cover the static interface features such as font size, color, and placement. For example, double coding of color and pattern for the same item were suggested for color-blind users, and red-green color combinations for the items in the same categories was avoided because of daltonism (or red-green colorblindness) as shown in Figure 4.



**Figure 4:** Interface change example - Double coding of color and pattern (also referred to as redundant coding) for daily eating pattern representation.

##### 3.1.2 Language

To select relevant language for menus, buttons, and titles, several changes were suggested during formative evaluation. Since the Senior Healthwatch system involves four different user groups (i.e., senior residents, remote family members, medical doctors, and nutritionists), the language and wordings were selected to be understood by all the user groups. It was reasonable to design on the basis of senior residents' language, since their language tended to be more non-technical and, consequently, could be understood by all other user groups. Thus, the formative usability evaluation phase was meaningful in checking the language selection for menus, buttons, and titles.

##### 3.1.3 Touch screen interface

Although the planned final platform of the Senior Healthwatch system was a touch-enabled LCD display, which implies that touching the screen with finger input only, the touch-enabled LCD display was not used until the formative evaluation session. Most of the problems detected through formative evaluation were the features related

to the validity of the touch screen input. For example, the size of buttons and the on-screen keyboard were revised because the width of a finger is much bigger than the mouse pointer in the typical desktop application. Additionally, the senior participants reported that the interface should support users with hand tremors by making the distance between buttons further apart. Another example was button touching feedback sound. The built-in feedback sound of the display emitted the sound wherever the screen was touched, regardless of the location of the point of contact. However, the users needed the sound feedback only when they touched the control buttons or menu, because older adults usually made errors when selecting those buttons. Therefore, the design team turned off the built-in feedback sound and implemented a camera shutter sound for each control button through the prototype.

### 3.2 Comparison of Prototyping Tools

In terms of the usage of prototyping tools, Table 1 shows a short comparison of the tools used in this project. Because each prototyping tool was used for different phases and objectives of the Senior Healthwatch development, this result cannot be generalized for other cases. For example, PowerPoint is able to create interactive functions that we did not use. Because we used PowerPoint only as a communication vehicle, not as an actual prototyping tool, PowerPoint is described as having no interactive feature in the visual feature column of the Table 1. Since the main prototyping tool for Senior Healthwatch was Flash, further lessons about using Flash as a prototyping tool will be discussed later in this paper.

**Table 1:** Comparison of prototyping tools used for *Senior Healthwatch*

	Paper	PowerPoint	Flash
<b>Phase</b>	Conceptual design as storyboard vehicle.	Iterative prototype design as communication vehicle.	Iterative prototype design & formative evaluation.
<b>Visual Feature</b>	Rough sketch. No interactive feature.	Fairly detailed. No interactive feature was used although it is possible.	Refined graphic design. Highly interactive features.
<b>Advantage</b>	Low cost. Useful communication vehicle.	Lower cost than flash. Better communication than paper in terms of look and feel. Images can be imported into Flash.	Highly functional. Fully interactive. Defines feel and look of final product.
<b>Dis-advantage</b>	Far from the look and feel of the final product. Images cannot be imported into PowerPoint or Flash.	Lack of interactive functions. Limited for remote evaluation.	High cost. Relatively more time and effort to build.

## 4 Discussion

Hix and Hartson (1993) listed a number of important features in a rapid prototyping tool that Flash supports such as the ability to easily develop, modify, and link screens; support of various interfaces and input/output devices; modification of links; integration with various media; ease of use; and good customer support.

**Table 2:** Comparison of training time for highly functional prototyping tools through online self-training courses

	Flash 5.0	Director 8.5	Visual Basic
<b>Modules &amp; Total Time</b>	37 Modules 9 Hours	56 Modules 14 Hours	72 Modules 18 Hours

\* Each module takes about 15 minutes.

A feature not supported by Flash is that of importing external procedures and programs. Flash uses its own programming language called ActionScript, requiring the developer to put codes in ActionScript format. However, the design team was able to acquire a fair amount of existing interface libraries and code from Flash developer resource sites such as Flash Kit (<http://www.flashkit.com>). In addition, the syntax and level of difficulty for use of ActionScript were easy enough for all members of the design team who were not computer programmers to learn and use. Compared to other highly functional prototyping tools such as Visual Basic, Visual Studio, and HTML with

CGI script, Flash has the advantage of ease of learning and use. To support this argument, we need to explore how long it takes for design teams to become proficient at the different tools. According to the online self-training courses provided by Virginia Tech, Flash takes the least time to complete the training modules (See Table 2). However, this argument cannot be generalized simply, because many possible factors may affect the level of proficiency through training process. Many tools that are easy to learn and use offer low versatility and tend to have basic functionality, but Flash supports a high level of functionality and versatility because Flash was primarily intended for developing interactive web applications using the Internet, including animation and even online games.

One advantage of Flash over other highly functional tools is the easy support of remote usability evaluation. A Flash prototype file can run on a web server without any additional technical support, but it can also be run on a stand-alone desktop as used in the formative evaluation of this project. Since the target platform of Senior Healthwatch was a touch-screen display, a laboratory-based evaluation was conducted. However, remote evaluation using Flash interactively over the Internet might be a good advantage over other tools for other practitioner's applications and prototyping.

Winograd (1995) indicated that there are issues to be addressed in the movement toward user-oriented software design environments from programming environments. The issues include

- *responsive prototyping media,*
- *user conceptual models,*
- *design languages, and*
- *participatory design.*

The first issue of utilizing responsive prototyping media is very easily performed with the sequence of various prototyping methods we used, shown in Table 1. However, being able to elicit user conceptual models could raise some criticism, because senior users were not actually involved at the stage of conceptual design, as shown in Figure 1. To elicit users' conceptual models, cognitive models or interface metaphors, the designer and user should be engaged in creating a world, so that the user-oriented software design environments could be developed. Through the formative evaluation phase of our project, participatory design concept was employed and found to be very useful, especially for the implementation of language and touch-screen interfaces. Design language refers to the vehicle to communicate functionalities to users in a natural and nonintrusive way (Rheinfrank, Hartman, & Wasserman, 1992). This issue could be the most challenging one for the environment designing software for seniors. For example, programmers or designers may have difficulty using a base language and libraries of common program elements, such as a tool bar or radio buttons, with which senior users are not really familiar. The enhancement of design language for senior users could facilitate the participatory design and expand the debugging environment.

## **5 Conclusion**

The procedure for prototyping described in this paper presents a method for rapid prototyping with the combination of three different tools, based on our experience. Each tool was used for different phases and roles in the prototyping process so that other practitioners could derive guidelines for relevant usage of the prototyping tools in their application development cycles. The prototyping process involving senior users was found to be useful to detect usability problems, especially regarding language and touch screen interface elements involving physical interaction. We have found that Flash can support the rapid prototyping process by reducing the trade-off that occurs between ease of using prototyping tools and the functionality of prototype. Also, in order to develop the environment for designing software for seniors, early involvement of senior users to develop conceptual models and refinement of design language to facilitate user-participatory design are necessary.

## **Acknowledgements**

This study was conducted at Virginia Tech with support from Carilion Biomedical Institute. The effort was directed by the Medical Automation Research Center at the University of Virginia. We would also like to thank reviewers for their helpful comments and the members of the Blacksburg Electronic Village Seniors and the Blacksburg Parks and Recreation Senior Center for their support.

## References

- Engleberg, D., & Seffah, A. (2002). A framework for rapid mid-fidelity prototyping of web sites. *In Proceeding of IFIP World Computer Conference*, Montreal, Canada.
- Isensee, S., & Rudd, J. (1996). *The art of rapid prototyping*. Cambridge, MA: International Thompson Computer Press.
- Kwahk, J., Smith-Jackson, T. L., & Williges, R. C. (2001). From user-centered design to senior-centered design: Development of a health information portal for seniors. *In Proceeding of Human Factors and Ergonomics Society 45th Annual Meeting*, 580-584.
- Kwahk, J., Smith-Jackson, T. L., & Williges, R. C. (2002). An application of neural network modeling to diagnose eating behavior of seniors. *In Proceeding of Human Factors and Ergonomics Society 46th Annual Meeting*, Baltimore, 180-184.
- Rheinfrank, J. J., Hartman, W. R., & Wasserman, A. (1992). Design for usability: Crafting a strategy for the design of a new generation of xerox copiers. In P. Adler & T. Winograd (Eds.), *Usability: Turning technologies into tools*. New York, NY: Oxford University Press, Inc.
- Ryu, Y. S., Williges, R. C., Smith-Jackson, T. L., & Kwahk, J. (2002). Comparison of design requirements methods for monitoring eating patterns of seniors in a smart house. *In Proceeding of Human Factors and Ergonomics Society 46th Annual Meeting*, Baltimore, 844-848.
- Stephanidis, C. (2001). User interfaces for all: New perspectives into human-computer interaction. In C. Stephanidis (Ed.), *User interfaces for all: Concepts, methods, and tools* (pp. 3-17). Mahwah, NJ: Lawrence Erlbaum Associates.
- Wilson, B. G., Jonassen, D. H., & Cole, P. (1993). Cognitive approaches to instructional design. In G. M. Piskurich (Ed.), *The astd handbook of instructional technology* (pp. 21.21-21.22). New York: McGraw-Hill.
- Winograd, T. (1995). Environments for designing. *Communications of the ACM*, 38(6), 65-74.