

This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.



For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

An effective prototyping method for delivering interaction design in industrial design education

Tek Jin Nam and Steve Gill
University of Wales Institute, Cardiff

Abstract

Increasingly industrial designers are required to engage in the design of computer-embedded products. Due to the interactive nature of these products, efficient prototyping is becoming more critical in the design process. This paper presents a designer-oriented prototype development method that facilitates the construction of fully functional physical prototypes. Unlike the multi-media authoring tools commonly used to date, this method employs a cheap and widely available software application: Microsoft PowerPoint with embedded Visual Basic. The method has been designed for use by programming non-literate designers and allows interactive physical prototypes to be created and tested without extensive specialist knowledge. This was achieved in part by simple modifications to a standard computer keyboard. Our case study on the use of this method in the teaching of information ergonomics demonstrated that students could effectively produce software-hardware hybrid prototypes with minimum effort in a short time scale. This method offers an excellent means of collaboration between designers and professionals from different domains. The prototyping method enables rapid and iterative design cycles in design education, research and practice.

Keywords: prototyping, interaction design, industrial design, design tools, computer-supported design, ergonomics & design

Introduction

With the rapid development of computer technologies, more and more products are computer-embedded, or *information appliances* (Norman, 1999). Mobile phones, digital cameras, digital audio players, home automation devices and so on are all examples of such information appliances. Many people refer to these products' emergence as ubiquitous computing, pervasive computing or smart environments (Ark and Selker, 1999).

Because of the interactive nature of these products the trend has enhanced the importance of Information Ergonomics in general and Interaction Design in particular. Modern education must address these issues effectively in order to maintain and enhance its relevance. Several domains, including computer science, psychology, and ergonomics, have recognised the need to include interaction design in their education and a number of universities around the world are actively engaged in the research and delivery of interaction design theories and

practices (ACM SIGCHI Curriculum Development Group, 1996).

In industrial design education, an awareness of information ergonomics for computer-embedded products is becoming ever more crucial. Among the systems-based major projects carried out by students of University of Wales Institute, Cardiff's (UWIC) Industrial Design course, about 70% develop concepts for information appliances. The industrial design domain must deal not only with the static and physical properties of these products but also the dynamic and interactive properties. Despite the strong emergence of computer-embedded products however, many industrial design courses have to date struggled to provide a systematic approach to delivering interaction design content.

One of the most important factors in delivering interaction design in an industrial design curriculum is the selection of an appropriate prototyping tool. Several prototyping tools and methods have

previously been investigated in the industrial design domain. Sharp (1998:49) reviewed three product interface simulation techniques: prototypes using immersive Virtual Reality, touch-screen virtual prototypes, and physical prototypes driven by virtual simulation. Using a prototyping technique based on touch screen technology, he showed that a virtual simulation prototype of a microwave oven was highly effective in predicting users' performance with the real product. Since the virtual simulation was based on touch-screen technology however, it could only allow two dimensional input and display and so the range of products that can be simulated with this technique is limited. For some products a physical interactive prototype is the only way to fully examine the product interface.

The other major issue is that there is no dedicated prototyping software available. There are packages capable of producing sophisticated interaction prototypes, *Director* and *HyperCard* to name but two. While these packages offer a number of benefits, there are problems associated with them. One of these is that more time can be spent understanding the software than the fundamental principles of interface design.

In order to address this problem we have researched and developed ways in which students can design and build effective interface prototypes with minimum software knowledge. We hope thereby to be in a position to concentrate on the study of the specialism: i.e. human-product interaction. In the following sections we briefly discuss the appropriateness of tools currently used in interaction design and then describe our own prototyping methods. We next offer a case study in which the resultant method was used for the delivery of Information Ergonomics in our university and conclude with a discussion on the potential for future development.

Prototyping of computer embedded products with existing tools

The lack of dedicated software for prototyping computer-embedded products has led to the use of multimedia authoring tools like *Toolbook* (Hustedde, 1996), *Director* (Gross, 1999) and *Hypercard* (Goodman, 1998). The

most popular of these among professional interaction designers is *Director* by *Macromedia*.

Director uses a theatre metaphor with a time line to combine graphical, text, sound and video objects in a series of frames (Director, 1999). Interactive features can be added using a program-specific scripting language called *Lingo*. The program is generally used to create interactive media for the World Wide Web, CD-ROM, information kiosks, presentations, and interactive TV. *Director* is a capable and complex program able to cope with most multi-media authoring, but its broad range of capabilities makes the user's learning curve longer and steeper.

One of the problems with the use of **multimedia authoring** tools for the prototyping of **computer-embedded** products is that with most authoring tools multi-media interactivity is largely based on mouse or touch screen. Computer-embedded products on the other hand usually have their own interface elements such as buttons, switches, dials or small display panels. Physical feedback to the user of interface elements plays an important role in the success (or otherwise) of the interface design.

The use of programming language within authoring tools is generally essential in order to achieve full interactivity with prototypes, because the basic features of standard authoring tools are not sufficient for implementing the operations of computer-embedded products. This means that users not only need to learn the basic functions of the authoring package but also complex programming routines. Designers therefore have to learn a great deal before they can make any effective use of these languages as a design tool. Even when this has been achieved the performance and scalability of package-based programming language tends to be limited.

One of the greatest obstacles in building an operational prototype is the difficulty of building a hardware prototype. Traditional methods require sophisticated language (i.e. C++) with additional electronic components, such as interface or communication devices.

The use of low-level communication protocols is required to control the communication between the software simulation and the hardware component. Interface elements and graphical representation is usually limited since it must be managed within a low level programming environment.

A new prototyping method

From the review of existing prototyping tools used in the design domain, we have identified the following requirements for a designer-oriented prototyping method.

- Scalable to complex level for products with sophisticated operations
- Capable of producing detailed graphical representation of products
- Ease of use for programming non-literate designers
- Allow a flexible and evolutionary approach to interface design.
- Utilise a widely available software application
- Provide a good metaphor for the development of interface concepts.

Based on these requirements we devised a prototyping method using *Microsoft Powerpoint* in conjunction with its own embedded *Visual Basic* programming language. We soon discovered that by combining the two features a uniquely designer-oriented prototyping method could be constructed.

The approach is different from a standard *Visual Basic* programming environment because users can make use of a variety of presentation features so that the necessity for

complex programming is dramatically reduced. Users first develop an interaction model using standard modelling techniques such as state diagrams. The model is transferred into a slide show using hyperlinks to join the various slides. Each slide represents a certain **state** of the product interface and the hyperlink represents an event change. This basic slide show using state transitions can then be evolved into a dynamic prototype using the programming features. For example, when a user presses a button to show an altered status in the product, the associated *Visual Basic* script can control the visibility of a graphic element in the presentation. Figure 1 illustrates an example of script changing the states of a product in a single slide. By changing the visibility of the icon graphics, it is possible to show multiple product status in a single slide.

One of the strengths of this prototyping method is that the application is widely available and performance is satisfactory even on low specification computers. It also allows an evolutionary approach since it is possible to start with a simple slide show developed later to a fully functional prototype including dynamic interactivity. Text, graphics, sound and video objects, can easily be included using standard presentation features. Graphic or multimedia objects can be manipulated using the object-oriented structures that *Visual Basic* offers. Above all the simplicity of the method allows designers to quickly focus on the interactive aspects of his or her design.

Having developed this approach our next goal was to look at methods of building a physical hardware prototypes driven by software

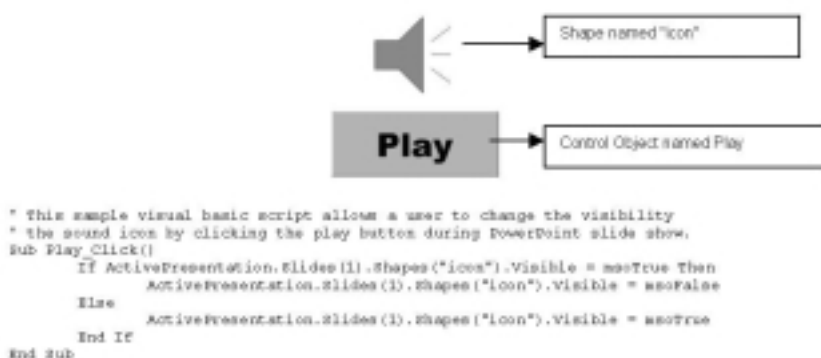


Figure 1 Sample Visual Basic script allowing dynamic interactivity within a single slide



Figure 2 Modified Keyboard

simulation Using operational hardware prototypes, a designer can examine tangible aspects of interface elements, such as the size and arrangement of the controls. We found that this could be easily achieved by connecting the interactive software simulation to a hardware mock-up through a special keyboard. The keyboard intercepts external interface events and converts them into key strokes (Figure 2). The hardware mock-up can be built easily as it uses simple electronic components, such as standard key switches.

A CASE Study: Information Ergonomics Module Delivery at UWIC

UWIC's BA Industrial Design and BSc Product Design & Manufacture courses share a level 2 module called *Information Ergonomics*. It was in this module that the feasibility of our prototyping method was examined.

The aim of the Information Ergonomics module is to teach students how to deal with a range of human-product interface issues for computer-embedded and other products. Students are expected to deliver critically developed information-based products considering various static and dynamic interactivity issues. The module is divided into three main phases. In the first phase, students are introduced the overall concept of interaction design, in the second phase they learn a variety of techniques required to carry out a project and in the third phase they are given a practical project in which they are

required to produce a demonstrable prototype. It should be clear then that in order to achieve the aims of the module, effective and easy prototyping tools are essential. The module is run over the course of a single semester, which leaves little time to consider the details of the tool itself. It follows then that the tool needs to be consistent throughout the module and scalable to the final outcome – that is, a functional hardware prototype.

The assignment we chose was a hand-held MP3 player because it was thought to have the right scale and number of interface elements. Students were required to build a software-based prototype using their newly learned information ergonomics theories and techniques. At the end of the module a hardware prototype was developed using the method detailed above.

State diagrams were used as a modelling technique for the MP3 user interface (Figure 3) and translated into a slide show that acts as the basic prototype. In the slide show, each slide represents different states of the interface and the event links between states can be defined using hyperlinks. The slide show is then evolved into a dynamic prototype using *Visual Basic*. In this way dynamic MP3 player features could be demonstrated within a single slide. Next, mouse-based interface was converted into key-based simulation. A model was built and connected to the customised keyboard which translates impulses from



Figure 3 Sample State Diagram of MP3 player interface

switches mounted in the model. The result was a prototype that demonstrated all the operations of the student's MP3 player (Figure 4).

Discussion

The above case study has demonstrated the validity of the prototyping method. Although minimum time was spent teaching the software application, students were able to produce a demonstrable prototype. Although our students had little or no background in programming, acceptance of the technique

was surprisingly high. Part of the reason for this was that material was made available on the Web so that lines of sample code could be cut and pasted into the *Visual Basic* editor for their own use. Students were pleased with the outcome and experiences and the new prototyping method is now a vital component of the module.

To date comparatively little attention has been paid to computer programming in industrial design education or practice, although the rapid growth of computer-embedded products make such skills essential. One of the traditional difficulties of teaching programming to design students is the time it takes for them to see the results of their acquired knowledge. Since the programming skill is directly related to the quality and realism of the prototypes produced, students using this method appear to understand the necessity of the skill more directly. A variety of program samples help them to understand the principles of products' operation and the results of their modifications to the standard sample code can be observed very quickly. As a result, the effort to understand programming techniques can be minimised and design students can focus on user interface issues in greater detail.

This method of prototyping offers an excellent means of collaboration between professionals from different domains. Since many product



Figure 4 A student hardware-software hybrid prototype

features can be easily demonstrated using an interactive prototype, communication between designers, ergonomists and engineers becomes easier increasing the efficiency of multi-disciplinary collaborative projects. In addition, the application of this type of methodology is not restricted to education. Applications abound in both research and practice. For example, design research activities are better facilitated since fuller interface analysis is possible using fully functional prototypes.

Future Research

This work suggests several future directions. One of the most obvious benefits of demonstrable prototypes is in the user evaluation phase of new product development. Using this method a system of analytic product evaluations can be incorporated within the prototype. For example, statistical software may be connected or incorporated to examine the usefulness of certain controls. By measuring the frequency and the time used for each of the interactive elements it will be possible to quantify the efficiency of interaction in a quantitative manner.

More control elements need to be developed for the realistic simulation of physical interface elements. Although various control objects, such as timers and key event handlers are available, it is difficult to realistically simulate some control elements, such as levers and sliders that have tactile feedback. An investigation into mechanisms and methods is required to develop such control objects for use within our prototyping environment.

Consideration might also be given to the ways of further incorporating the prototyping tool into UWIC's Industrial Design programme. By incorporating elements of the prototyping methodology earlier in the programme we hope to create graduates with high ability levels in this important and neglected area. By combining this study with other subjects such as model making and user study yet more depth can be added.

Conclusion

The role of designers in the development of the growing number of computer-embedded products make the study of interaction design ever more crucial. We have presented an efficient and evolutionary prototyping methodology that employs cheap and widely available software applications. Our case study on the use of this method in the teaching of information ergonomics demonstrated that designers could effectively produce a software and hardware hybrid prototype with minimum effort in a short time scale. This work contributes towards design education, research and practice by providing an efficient designer-oriented prototype development method that produces tangible designs with a functioning interactive element.

References

- ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group (1996), *Curricula for human-computer interaction*, ACM Press. Available at <http://www.acm.org/sigchi/cdg/index.html>
- Ark, W.S. and Selker T. (1999) 'A look at human interaction in pervasive computers'. *IBM systems journal*, 38, 4, 504- 507.
- Goodman, D. (1998), *Complete HyperCard 2.2 Handbook*, iUniverse.
- Gross, P. (1999), *macromedia DIRECTOR 7 and LINGO AUTHORIZED*, macromedia PRESS.
- Hustedde, S.F. (1996), *Developing with asymetrix toolbox: applied programming theory*, Wadsworth Pub Co.
- Norman (1999), *Invisible computers*, MIT Press, London.
- Sharp, J. (1998), *Interaction design for electronic products products using virtual simulations*, PhD Thesis, Brunel University, UK.