TOWARDS A USER INTERFACE PROTOTYPING SYSTEM

Mark Green
Unit for Computer Science
McMaster University
Hamilton, Ontario
Canada

ABSTRACT

In this paper we present an approach to the prototyping of user interfaces. In this approach a prototype user interface is constructed from a set of building blocks. Each of these building blocks simulates a small part of the user interface. The set of building blocks available to the user interface designer is documented in a building block catalogue.

The basic properties of building blocks and how they are connected to form a prototype are discussed along with the format and contents of the building block catalogue. A program, called PROSYS, that can be used in prototype construction is briefly described. Several examples are used to illustrate our approach.

KEYWORDS: user interfaces, human factors, software prototypes

1. Introduction

The user interface is one of the most important components of a computer system and one of the hardest to construct. One of the major problems in user interface design is our inability to predict the response of users to a particular design. Quite often the inappropriateness of a user interface is not detected until after the finished product is delivered. At this point correcting the problems in the design is an expensive undertaking.

Is the situation any different in other design oriented fields, for example architecture or engineering? The answer to this question is both yes and no. The designers in these fields are also unable to predict the acceptability of their designs. But, they counter this uncertainty by using a methodology that includes several evaluation steps. In some fields, such as architecture, there is a large body of successful designs that the designer can draw upon for ideas. In this way the architect is rarely w rking in a vacuum. This is not the case in user interface design since there are very few successful designs for the designer to study. And, unlike architecture we have not developed an effective means of documenting these designs. Another tool used by the designers in other fields is the prototype or mock-up. This allows the designer to evaluate his design before it is put into practice. These prototypes can vary from simple rough sketches to detailed working models. Computer science is the only major design field that does not view the prototype as an important design tool.

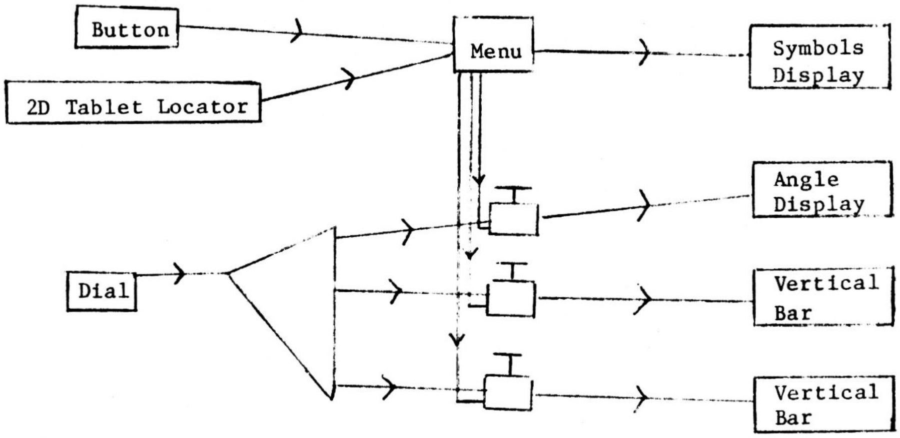

Fig. 1 Example Prototype

Why are prototypes used in other design fields? A prototype is constructed in the design phase of a project, usually before the final design has been decided upon. The prototype helps the eventual users of the product (in our case a user interface) visualize how the product will operate. The users can then give their comments on the acceptability of the product. In this way design errors can be corrected in the design phase. Prototypes can also be used in the evaluation of several competing designs. This can be very helpful in a field where there are few previous designs to draw upon.

What determines a good prototype? In our opinion a good prototype should possess the following three properties:

1) The prototype must be an accurate model of the real system. It need not model the complete system, but for the parts it does model it must be accurate. If this were not the case the user's comment would not be applicable to our design.

2) The cost of the prototype must be considerably less than the cost of the real system. If this were not the case we might as well construct the real system and use it as the prototype.

3) It should be easy to modify the prototype. This allows the designer to experiment with different alternatives and to incorporate and test the changes motivated by user experience.

In this paper we will consider the prototyping of user interfaces. In particular we will present a methodology and supporting computer system to aid in the production of user interface prototypes. In the development of this methodology the above three properties of good prototypes were always kept in mind.

2. An Approach to Prototype Construction

Our approach to prototype construction is referred to as the building block approach. In this approach a prototype consists of a number of basic building blocks connected by data paths. Each of the building blocks in the prototype implements one input technique, output technique, interaction technique, or control structure. A building block simulates one small part of the user interface. The data paths carry data and control signals from one building block to another. The user interface designer constructs a prototype by selecting the building blocks required, and connecting them with data paths. A program, called PROSYS, is used to construct, and edit the prototypes.

A prototype for part of a user interface is shown in Fig. 1. The purpose of this part of the user interface is to set the values of three numeric parameters. One of these parameters is an angle and its value is displayed as the angle between two lines. The other two parameters are absolute numbers, the length of two lines on the display are used to represent their values. To change the values of these parameters one dial is available. This means that only one parameter can be changed at a time. A menu and tablet are used to select the parameter that is currently being modified. In the visual representation of a prototype we use geometrical figures to represent the building blocks and lines to represent the data paths. We will use this example to illustrate a number of the features of our prototyping system.

In our prototyping system there are four basic building block types. These types are: input technique, output technique, interaction technique, and control. An input technique is a building block that produces graphical data. To produce this data it must interact with the user through the use of an input device. In our example the input techniques are Button, 2D Tablet Locator, and Dial.

An output technique is a building block that consumes graphical data. In the process of consuming the data the output technique produces an image on the graphics display. This image is a representation of the data received by the output technique. In our example the building blocks Symbol Display, Angle Display, and Vertical Bar are output techniques.

An interaction technique is a building block that transforms graphical data. It is a combination of input and output techniques and usually corresponds to a section of the user-computer dialogue. The building block Menu is the only interaction technique in our example.

The control building blocks are used for structuring and controlling the prototype. There are control building blocks for controlling the flow of data along a data path, splitting a data path into several new data paths, or joining several data paths into one. A control building block does not interact with the user. The two types of control buidling blocks in our example are split and valve. The split building block divides the output of the Dial building block into three streams. The flow of data along each of these three streams is controlled by a valve.

The designer selects the building blocks from a catalogue. This catalogue is divided into four sections, one section for each of the basic building block types. There is an entry in the catalogue for each of the building blocks available in PROSYS. The format of the catalogue entries is described in Section 3. Usually the designer will find several building blocks that can be used to model a particular part of his user interface. The choice between these building blocks will be based on human factors considerations, the available graphics devices, and experimentation.

A data path can only carry one type of data value or control signal. Each building block has one or more connectors that a data path can be connected to. A connector also has a data type, this type must agree with the type of the data path connected to it. A data path can only carry data values in one direction. The connectors on a building block are labelled as either input or output connectors. At any point in the development of a prototype a building block can be disconnected from a data path an replaced by one that generates or consumes the same type of data.

There are seven data types in our prototyping system. Six of these data types are for graphical information, and one is for control signals. The values of the control data type are enable and disable. Most of the building blocks in our system have a control connecter called "enable". In order for the building blocks to function the value at this connector must be enable.

The graphical data types are: discrete, numeric, multidimensional, multidimensional sequence, and symbols. The simplest of these data types is discrete. The values of this type are selected from a finite set of symbols. Typically only two or three distinct values are used. This data type roughly corresponds to the button virtual device in the Core Graphics System [GSPC 79].

The values of the type text are alphanumeric character strings. This type roughly corresponds to the keyboard virtual device in Core Graphics System..

The values of the numeric data type are real numbers. This is the type of value produced by the Core valuator device.

The multidimensional data type is a collection of numeric values. The order of these values is typically not important. The most common use of this data type is the representati n f two and three dimensional points.

A multidimensional sequence is an ordered sequence of multidimensional numeric values. This data type can be used to represent the data points in a graph, or the value produced by an inking or sketching operation. In these examples the ordering of the values indicates their time sequence.

The symbols data type represents a collection of symbols or icons positioned in two or three dimensional space. The symbols are an abstract representation of the objects in the problem domain of the user interface. For example, an individual symbol may be a component on a circuit diagram, the entire circuit diagram is a value of type symbols. This data type can also be used to represent a menu where each of the commands is a separate symbol. The individual symbols in a value of this type are composed of a symbol name and a position in two or three dimensional space. If multiple symbol instances are meaningful in an application then a given symbol name may appear more than once in a symbols value.

The advantages of the building block approach are:

1) Prototype construction is both quick and easy. Most of the building blocks required for a prototype will have already been written so the construction process reduces to the selection and connection of building blocks.

2) It is easy to modify a prototype made up of building blocks. Most changes involve only replacing a building block by one with similar input and outputs, or re-routing a data path.

3) It is possible to incrementally construct the prototype while the design is being developed. As the designer selects interaction techniques the corresponding building blocks can be added to the prototype. In this way the designer can evaluate and experiment with different techniques while he is producing his design.

As can be seen the major aim of the building block approach is to make prototype construction and modification as efficient as possible.

3. The Building Block Catalogue

The major component of our prototyping system is the building block catalogue. This catalogue has two main purposes. First, it defines the building blocks that are available to the user of the prototyping system. In this way it serves as the prototyping system's user's manual. Second, it is a source of design ideas the designer can draw upon in the development of his user interface. Thus, the building block catalogue is useful outside of the prototyping system.

The building blocks in the catalogue have been taken from a number of sources. One valuable source is the standard text books on interactive computer graphics; [Newman and Sproull 1979] and [Foley and Van Dam 1982]. Another source is the articles that have been written on particular interaction techniques, such as [Evans, Tanner, and Wein 1981]. A careful study of successful user interfaces has suggested a number of building blocks. The building block catalogue is not static, new entries are added as they come to our attention.

In this section we will look at the general stucture of the building block catalogue, the format of its entries, and outline its contents.

The building block catalogue is divided into four sections, one section for each type of building block. These four major sections are further divided into subsections based on properties of the building blocks. All the entries in the catalogue have the same basic format, but the details of this format will vary from section to section due to the differences between the building block types.

The general format of a catalogue entry is shown in fig. 2. An entry is divided into a number of fields. The name field contains the name of the building block. A building block name is composed of several words that describe its operation or purpose.

```
Name:
Classification:

Parameters:

Connectors:

Description:
```

Fig. 2 General Catalogue Entry Format

The classification field contains a classification code that can be used to uniquely identify this particular building block. The classification code is made up of several numbers separated by periods. The first number is the major catalogue section the building block belongs to. The last number is the building block's position within a subsection of the catalogue. The intervening numbers are the subsections of the catalogue containing this building block. Since the classification code is based on the properties of the building block, it can be used to find other building blocks with similar properties.

Each building block can have a number of parameters. These parameters are used to specify the properties of individual instances of the building block. Some example parameters are the position on the screen of an output technique, scaling factors for the values produced by an input technique, or the size of an image. These parameters are defined in the parameters field of the catalogue entry.

The connectors field contains the definitions of the building block's connectors. For each connector its name, type, and direction must be specified.

The description field contains a verbal description of the behavior of the building block. This description tells the designer how the building block interacts with the user. It does not mention how the building block is, or could be implemented.

Name: Vertical Bar
Classification: 1.3.2

Parameters:
    screen_position : multidimensional
    offset : numeric
    scale : numeric

Connectors:
    value : numeric, input
    enable : control, input

Output Device:
    Any output device capable of drawing and erasing lines in a fraction of a second.

Description:
    In this output technique a vertical bar is used to represent the value of a number. The height of this bar is proportional to the value on the connector 'value'. The parameters 'offset' and 'scale' specify the value at the bottom of the bar and the rate at which the bar changes height. The parameter 'position' gives the screen position of the bottom left hand corner of the bar.

Fig. 3 Output Technique Catalogue Entry

3.1. Output Techniques

An output technique is a building block that consumes information, and produces a graphical representation of it on a display device. From this we can conclude that two major characteristics of an output technique are; the type of the data it displays, and the nature of the display device.

This section of the catalogue is divided into six subsections, one for each of the graphical data types defined in section 2. The subsection for each type contains all the output techniques that can be used to display data of that type. Each entry in the output techniques section has a field called output device. This field contains the display devices required to use this output technique. An example output technique entry is shown in Fig. 3.

Output Technique Section

1.1 Discrete
    On/Off Display
    State Display

1.2 Text
    Text Window
    Scrolling Display
    Page Display

1.3 Numeric
    Print Number
    Vertical Bar
    Horizontal Bar
    Angle Display
    Symbol Size

1.4 Multidimensional
    Bar Chart
    Pie Chart
    Marker
    Scaled Symbol
    Vector

1.5 Multidimensional Sequence
    2D Plot
    3D Plot
    Scatter Plot

1.6 Symbols
    Spatial
    Connected Icons

Table 1 Output Technique Building Blocks

A list of the output techniques currently in our catalogue is shown in Table 1.

3.2. Input Techniques

An input technque is a building block that produces information by interacting with the user through the use of an input device. Two major characteristics of an input technique are; the type of the data produced, and the input device used.

Name: 2D Tablet Locator
Classification: 2.4.1

Parameters:
    origin : multidimensional
    scalex : numeric
    scaley : numeric

Connectors:
    value : multidimensional, output
    enable : control, input

Input Device:
    Any input device that produces a two dimensional value can be used. The best choice is either a tablet or a mouse.

Description:
    This input technique produces a two dimensional value that is treated as a point. The parameter 'origin' specifies the origin of the coordinate space. The parameters 'scalex' and 'scaley' specify the size of the coordinate space in terms of the input device's coordinate space.

Fig. 4 Input Technique Catalogue Entry

The input technique section of the catalogue is divided into six subsections, one for each of the graphical data types. As in the output techniques section, all the building blocks in a subsection produce the same type of data. Each entry in this section has an extra field called input device. This field contains the input devices required by the input technique. In most cases this field will contain the name of a generic or virtual device, but in some cases a specific hardware device may be required. An example input technique entry is shown in fig. 4.

The current contents of this section of the catalogue are summarized in Table 2.

3.3. Interaction Techniques

An interaction technique is used for transforming information. A building block of this type has one or more input connectors and one or more output connectors. The types of the output connectors are not necessarily the same as the types of the input connectors.

Input Technique Section

2.1 Discrete
    Push Button
    Function Button

2.2 Text
    Keyboard

2.3 Numeric
    Numeric Keypad
    Dial
    Vertical Tablet
    Horizontal Tablet

2.4 Multidimensional
    2D Tablet Locator
    3D Tablet Locator
    Light Pen Locator

2.5 Multidimensional Sequence
    Rubber Band Line
    Equal Time Inking
    Equal Space Inking

2.6 Symbols
    Keyboard Select
    Light Pen Select (Pick)
    Tablet Select
    Character Recognizer

Table 2 Input Techniques Building Blocks

A common example of an interaction technique is a command menu. This interaction technique is used to select a command from a list of commands that have been presented to the user. In our example prototype we show one form of a command menu building block. The catalogue entry for this building block is shown in fig. 5. The main output of any command menu is a set of control signals that are used for controlling other parts of the user interface. The second output of this building block is of type symbols. It represents the set of commands to be selected from, and the menu's tracking cross. The inputs to this building block are a multidimensional value and a discrete value. The multidimensional value is used for positioning the menu's tracking cross. The discrete value indicates when the user is pointing at the command he wants. In this building block the transformation being performed is called selection. The inputs to the transformation are values of type multidimensional and discrete, and the output is a set of control values. There are several other kinds of command menus, they differ mainly on the types of their input values.

Name: Locator Based Command Menu
Classification: 3.1.1.2

Parameters:
  menu_layout : symbols

Connectors:
  cursor_position : multidimensional, input
  select : discrete, input
  menu_display : symbols, output
  $command_1$: control, output
  $command_2$: control, output
  . . .
  $command_n$: control, output
  enable : control, input

Description:
  A command menu is used to select a command from a list of commands that have been presented to the user. The parameter 'menu_layout' gives the names and positions of the commands in the menu. This parameter along with a tracking cross make up the output 'menu_display'. The 'command' output for the selected command has the value enable, while all the other 'command' outputs have the value disable. The input 'cursor_position' indicates the position where the user is pointing. When the value of the input 'select' becomes 'on' this position is used to determine the command the user has selected.

Fig. 5 Interaction Technique Catalogue Entry

As can be seen from this example interaction techniques can be more complicated than either input or output techniques. This makes it harder to develop a classification scheme for interaction techniques. Currently, we are using a two level classification scheme. The first level is the transformation performed by the interaction technique. This level is open ende , new transformations can be added at any time. The second level is the type of the major output produced by the interaction technique. All seven data types need not appear under every transformation, and in some cases several types are grouped together. The current classification scheme for interaction techniques, and a summary of the contents of this section of the catalogue are shown in Table 3.

Interaction Technique Section

3.1 Selection
  3.1.1 Control
    Keyboard Based Command Menu
    Locator Based Command Menu
    Pick Based Command Menu
  3.1.2 Graphical Data
    Locator Based Data Menu
    Pick Based Data Menu

3.2 Geometrical Transformation
  3.2.1 Multidimensional
    2D Translate
    3D Translate
    2D Scale
    3D Scale
    2D Rotate
    3D Rotate
  3.2.2 Multidimensional Sequence
    2D Translate Sequence
    3D Translate Sequence
    2D Scale Sequence
    3D Scale Sequence
    2D Rotate Sequence
    3D Rotate Sequence
  3.2.3 Symbols
    Translate Symbols
    Scale Symbols Position
    Rotate Symbols

3.3 Manipulation
  3.3.1 Symbols
    Dragging
    Symbol Positioning

Table 3 Interaction Techniques Building Blocks

3.4. Control

Control building blocks are used for structuring and controlling the flow of information in a prototype. There are only a small number of basic control building block types, and each of these types is represented by a different geometrical shape in the diagram of a prototype. The control section of the building block catalogue is divided into two subsections; pipe fitting, and control flow. The building blocks in the pipe fitting subsection are used for joining several data flows together, or splitting a data flow into several new ones. These building blocks are used for structuring the flow of information between different parts of the prototype.

Name: Valve
Classification: 4.2.1

Parameters:

Connectors:
- in_value : any_type, input
- out_value : any_type, output
- control_value : control, input

Description:
This building block is used for controlling the flow of information along a data path. If the value of the connector 'control_value' is enable then information can flow from the connector 'in_value' to the connector 'out_value'. Otherwise, the flow of information between the two connectors is blocked. This building block does not alter the information flowing through it in any way.

Fig. 6 Control Catalogue Entry

The building blocks in the control flow section are used for controlling the flow of data. The only building block currently in this section is the valve, it catalogue entry is shown in fig. 6.

The current contents of the control section of the catalogue are summarized in Table 4.

4. The Prototyping Program

The program PROSYS is used to construct and modify user interface prototypes. The major aim of this program is to make the prototype construction process as easy and efficient as possible. In accordance with this aim PROSYS is a highly interactive program that allows the user to intermix prototype construction and testing. At any point in the development of a prototype the user can test all or part of the prototype and make any necessary changes. There are no special editing or testing modes in this program.

Control Section

4.1 Pipe Fitting
- 2-Way Join
- N-Way Join
- 2-Way Split
- N-Way Split

4.2 Control Flow
- Valve

Table 4 Control Building Blocks

PROSYS is written in the object oriented language EDL [Green and Philp 1982]. Each of the building blocks in the building block catalogue is implemented by an EDL object. The data paths in the prototype are simulated by commands that are passed between the objects. PROSYS itself is a small collection of objects that manage the objects in the prototype and provide a convenient mechanism for adding and manipulating building blocks. Since the building blocks are standard EDL objects the user interface designer can easily add any new building blocks that are required for the user interface he is working on. Because the prototype is a collection of objects it can be running while the designer is adding to the prototype.

The major advantages of this type of prototyping program are:

1) Prototype construction and testing can proceed at the same time.

2) The set of building blocks can easily be extended by the user interface designer.

3) The prototype is easy to modify and

experiment with. This encourages the exploration of new design ideas.

5. Conclusions

We have presented an approach to user interface prototype construction that is based on the concept of a set of building blocks. The building blocks available to the designer are documented in a building block catalogue. An interactive program exists for constructing and editing prototypes. At the present time our efforts are directed at expanding the building block catalogue to include more input, output, and interaction techniques. The building block approach to prototype construction is quite general and there is no reason why it could not be applied to other areas of computer science.

Acknowledgements

I would like to thank Ron Baecker and Bill Buxton for many helpful comments during the early stages of this work. I would also like to thank McMaster University for providing the facilities necessary for this work

Bibliography

[Evans, Tanner, and Wein 1981] Evans K.B., P.P. Tanner, M. Wein, "Tablet Based Valuators that Provide One, Two, or Three Degrees of Freedom", SIGGRAPH'81 Proceedings, p.91-96, 1981.

[Foley and Van Dam 1982] Foley J.D., A. Van Dam, Fundamentals of Interactive Computer Graphics, Addison-Wesley, Reading Mass., 1982.

[Green and Philp 1982] Green M., P. Philp, "The Use of Object Oriented Languages in Graphics Programming", Graphics Interface'82 Proceedings, 1982.

[GSPC 1979] "Status Report of the Graphics Standards Committee", Computer Graphics, vol.13, no.3, 1979.

[Hanau and Lenorovitz 1980] Hanau P.R., D.R. Lenorovitz, "Prototyping and Simulation Tools for User/Computer Dialogue Design", Computer Graphics, vol.14, no.3, p.271, 1980.

[Newman and Sproull] Newman W.M., R.F. Sproull, Principles of Interactive Computer Graphics, 2nd Edition, McGraw-Hill, New York, 1979.