DEVELOPMENT OF AN INFORMATION INPUT SYSTEM FOR TELIDON

S. Shlien, W. Sawchuk, C.D. O'Brien, and M. Saaltink
Communications Research Centre, Ottawa

ABSTRACT

Telidon is a public access, interactive, information retrieval system which uses domestic television receivers suitably modified or supplemented. The Telidon system will permit the access of textual or graphic images from vast data banks by conducting systematic searches through numeric keypads. One of the requirements of Telidon is a 'user friendly' system for inputting imformation in the form of 'Telidon pages'. The information preparation system must be simple enough to learn with relatively little training, yet powerful enough to utilize the full capability of Telidon. In addition the cost of the hardware should be affordable by small scale information provider service. Prototypes of the system running on a PDP-11/10 with floppy disks and a Norpak colour display and offering the capabilities of a full graphics/text editor are currently being used for creating Telidon pages. The files created by the system are coded into Picture Description Instructions and are ready for transmission to Telidon terminals without any additional processing.

MISE AU POINT D'UN SYSTÈME D'ENTRÉE DE DONNÉES POUR LE TÉLIDON

RÉSUMÉ

Télidon désigne un système interactif de recherche documentaire qui est mis à la disposition du public par l'intermédiaire d'un téléviseur ordinaire modifié ou augmenté à cette fin. Le public peut ainsi avoir accès à des textes et à des images stockés dans de grandes banques de données, en procédant à une recherche systématique des enregistrements à l'aide d'un clavier numérique. Le Télidon compte parmi ses caractéristiques fondamentales un système d'entrée de données au moyen de pages, afin de faciliter la tâche de l'utilisateur. Ce système d'entrée doit donc être assez simple pour qu'on puisse s'y initier en quelques minutes, mais il doit tout de même comporter les codes nécessaires à l'utilisation totale des capacités du Télidon. De plus, le prix du matériel employé ne doit pas le mettre hors de portée des petits fournisseurs d'information. Les prototypes du système qui sont employés à l'heure actuelle pour fabriquer les pages Télidon se composent d'un PDP-11/10 à disques souples et d'un écran de visualisation couleur Norpak qui offrent autant de possibilités qu'un système complet de traitement de textes et de graphiques. Les fichiers ainsi créés sont ensuite codés à l'aide des instructions de description de l'image en vue de leur transmission aux terminaux Télidon sans autre traitement.

DEVELOPMENT OF AN INFORMATION INPUT SYSTEM FOR TELIDON

Introduction

The Canadian Telidon Videotex system is an information retrieval system to be available to the general public through their own home television receivers. Using a simple numeric keypad, a person would be able to select pages of information such as the news, weather, sports, stock market quotations, etc., and display them on his television. The information would be transmitted as printed text with accompanying graphical illustrations along a telephone or cablevision line, pass through a microprocessor attached to the television receiver and appear in colour on the monitor. Similar systems developed in England and France are just becoming publicly available and efforts have been made to introduce them into other European countries, as well as North America.

The Telidon system codes and transmits the picture using Picture Description Instructions (PDIs). These are very concise geometric commands for drawing lines, arcs, points, shading rectangles, polygons, curves and printing text. The instruction set and its syntax has been designed specifically for efficient communication of the graphical information as cheaply and quickly as possible. As a result, the receiver must have an intelligent microprocessor to translate these instructions into commands which will drive a display. This special hardware would appear as an add on device that would be bought at a department store or specialty shop and would be attached to the television receiver.

The PDI codes, like machine instructions, are not suitable media for people to use to describe a picture and therefore an information input system is required to allow a user to 'draw' and convert it to PDI code. The drawing system should also allow the user to edit an existing page, correct errors or include additional information. Such a system must contain the usual hardware for a graphics editor (display-console, joystick and memory) as well as considerable software for reading and writing a PDI file. With such a system, there is the usual tradeoff between speed, hardware costs, software development costs, ease of operation and generality. Since an input system would be used by many people with little or no computer background, heavy priority was placed in designing a fool-proof system that a person could learn to use by himself in a short time. Furthermore, it was also important that the system demonstrates the full capabilities of the Telidon system.

Several prototypes of the system using the PDP-11 RT-11 operating system have been designed and tested. Presently, they are software based systems, but there are plans for identifying the key modules and transfer them to firmware. This report describes the system from both the user's and programmer's view. Before going into the details, we will first discuss the hardware, PDI grammar and programming language since they are the main factors which guided the design.

The prototype hardware consisted of a PDP-11/10 system with 32K core memory, floppy disk, television like monitor, a display processor, a NORPAK RGP500 with a 320 by 240 raster memory, hardware cursor controlled by a joystick and a typewriter keyboard with 16 function buttons. The keyboard is used for entering in text and the 16 function buttons can be used for specifying colours and drawing commands. The display processor stores the image in both raster and geometric forms. Any object drawn on the screen can be tagged, allowing the user to alter it using the joystick.

Along with the RT-11 operating system, the PDP-11 Assembler, Fortran and the IGPL languages were also utilized. The latter is a graphics language developed by the Communications Research Centre in conjunction with Norpak Ltd, suitable for controlling the display hardware.

Besides the IGPL language and the display hardware, the other major influence on the design of the information input system was the PDI definitions. In order to utilize the full power and flexibility of the PDI commands, there is a button dynamically assigned for almost every PDI opcode. The user draws a picture on his screen by recording the associated PDI instructions in a buffer area, and executing these instructions with another software module. This interpretative approach was expedient for software development since it helped ensure the integrity of the PDI codes. Furthermore, future upgrades will be able to utilize firmware Telidon home terminals for executing the PDI commands.

A description of the PDI grammar is given in the following table:

THE SIMPLIFIED PDI GRAMMAR

```
<PDI file>        :: = <Text>
                  :: = <Text> <Shift in> <Graphics>
                  :: = <Text> <Shift> <Graphics> <Shift out>
                       <PDI File>
<Text>            :: = <ASCII character>
                  :: = <Text> <ASCII character>
<Shift in>        :: =  ASCII character code 15
<Shift out>       :: =  ASCII character code 14
<Graphics>        :: = <Item>
                  :: = <Graphics> <Item>
<Item>            :: = <Draw command>
                  :: = <Control prefix> <Draw command>
<Draw command>    :: = <opcode> <opcode data>
<Control prefix>  :: = <control>
                  :: = <control prefix> <control>
                  :: = <>
<Control>         :: = <Colour value> <value>
                  :: = <control status> <control data>
```

```
<Opcode>           :: = <point> <type>
                   :: = <line> <type>
                   :: = <rectangle> <type>
                   :: = <arc> <type>
                   :: = <polygon> <type>
                   :: = <bit> <type>
<Control status> :: = <control code> <clear> <type>
                   :: = <control code> <draw control> <type>
                   :: = <control code> <tone control> <type>
                   :: = <control code> <line texture> <type>
                   :: = <control code> <fill control> <type> <fill data>
                   :: = <control code> <text control> <type> <text data>
<Value>            :: = <black>
                   :: = <blue> or <grey 1>
                   :: = <red> or <grey 2>
                   :: = <purple> or <grey 3>
                   :: = <green> or <grey 4>
                   :: = <cyan> or <grey 5>
                   :: = <yellow> or <grey 6>
                   :: = <white>
```

The terminals are ASCII character codes selected in accordance with the ISO 646 standard character set.

<type> here refers to an atom which has different meanings in different contexts. For opcodes, type indicates whether the coordinates are relative or absolute, whether the object is shaded or the direction of drawing the arc. For control status, type either passes parameters or subdivides the fill control or text control into different sub control commands.

<tone> determines whether the colour and value are interpreted as a colour or grey tone. On our current interpretative implementation grey-tone PDIs may be created but not displayed.

<draw control> principally controls the flash mode of the item. If flash is on, the item will blink on and off.

Knowledge of the details of the PDI coding scheme is not important for operation of the software except for a few specific points. There are two types of PDI commands and the order in which they occur in the file will affect the appearance of the displayed picture. The first type is drawing commands and include operations to draw points, lines rectangles, arcs, polygons and text. The second type is status commands which affect the colour and mode of display of all the following drawing commands. These status commands remain in effect for the duration of the file or until they are overridden by other similar control commands. The order of displaying the items is also important in the sense that if two objects overlap each other, the object which was last displayed will appear on top.

Operation

The information input system is an editor which allows the user to create, modify or append Telidon pages. It is more general than a text editor since the program manipulates both text and graphics commands and allows the user to see the picture exactly as it would be displayed by Telidon. The Telidon pages are stored as separate files which are read or written on disk under user's control. If the user wishes he is able to merge several files into one Telidon page. All the files are coded directly into PDIs avoiding the problem of having to back up any intermediates files which would be needed whenever the user wants to change an existing PDI file. The program also contains two translators; one to convert the user's command into PDI instructions and the other to convert the PDI codes into commands which drive a hardware display.

The user controls the program by hitting a sequence of buttons and drawing objects using the joystick. When the program is started, the main menu appears at the bottom of the screen as a series of eight annotated blue rectangles (or buttons). To get into the different modes of the program, the user either positions the marker over one of the buttons and strikes the button on the joystick, or else pushes the corresponding function button on top of his console. The main menu is replaced by a new series of buttons appropriate for this mode; the last button 'RTRN' is always reserved for tracing back to the main menu. In some of the drawing modes, the menu disappears and prompt messages appear on top of the screen to tell the user to select the points associated with the desired object. At any time, the user can abort the current mode by hitting function 16 on top of the console, causing the current set of buttons to reappear.

The user has control over the system by about 100 button initiated functions, but by arranging these buttons in a tree structure the hardware is simplified and the user is never overwhelmed with choices. This approach has the disadvantage of requiring the user to work his way through a tree structure in order to perform some specific function but other designs would require more extensive hardware and a more experienced operator. In designing this system it was hoped that it would be completely transparent to the user and that he would be able to learn how to operate it with little outside assistance. The program is not far from this state, provided the user undertands the general philosophy of the PDI commands and the meaning of the status controls.

The primitive alpha-geometric commands of the Telidon system are:

(1) point
(2) line
(3) rectangle
(4) arc
(5) polygon
(6) text.

The first two commands permit the user to draw points and lines anywhere on the screen. The rectangle command causes the system to draw a rectangle of any size (either shaded or outlined) at the coordinates specified by the user. The arc command allows the user to draw either a circle or part of a circle (outlined or shaded) by specifying a start point on the circle, the centre of the circle and an end point. If the start point and the end point are the same, the program assumes the user desires a complete circle. The polygon command allows the user to specify the vertices of a polygon to be outlined or shaded. If the polygon is to be shaded correctly, the user must ensure that it is a true polygon in the geometric sense (lines do not intersect except at vertices), and that there are no more than 200 vertices. The text command allows the user to type in text to be printed out anywhere on the screen. The user can best appreciate the power of these primitives by examining some of the pages created for the Telidon system. For example, a map of Canada can be drawn as a 150 vertex polygon. All of these drawing commands are accessed by hitting the 'DRAW' button which appears on the main menu.

When the user draws an item, he has a choice of one of eight colours, a choice of different textures, a choice of several text sizes and various shading modes. Rather than asking the user to specify these choices each time he attempts to draw a new object, the program assumes the same modes used for the previous item carries over unless otherwise specified by the STAT buttons. The initial default parameters will cause the program to display every object in white, draw solid lines, fill rectangles, arc and polygons, and use the basic character set.

The EDIT button allows the user to correct any mistakes that have been made. The object can be deleted using the ERAS button, its display mode can be changed using the MDFY button, another copy of the object can be drawn using the COPY button, or the order of displaying the objects can be changed with the INST button. The RECL and STOR buttons permit recalling an existing PDI file or storing the current picture in a PDI file.

Implementation

The system was engineered mainly in software using off the shelf computer hardware and operating systems in order to meet the tight deadlines required by Telidon experiments and to ensure that only mature technology was being used in the prototype. Nevertheless, the system was designed to be compatible with future plans to convert some of this software into firmware. The software can be separated into four main modules, each of which alters or displays a portion of the PDI buffer. The main module is called the 'human interface' which displays the different menus of buttons and transfers control to the other support modules. The supporting modules translate the commands received from the main module into PDI's, edit the PDI instructions in the buffer, or convert the PDI instructions into opcodes which would drive a hardware display device. All of the modules read and update the control status vector which stores the current display status of an item (colour, position, texture, blink mode, fill mode and character size).

Since the current hardware does little except draw points, lines, characters and areas, considerable software had to be written to interpret PDI commands for drawing circles and shading them with complicated textures. The editing of existing PDI files required special care since inserting a colour command, in front of a drawing command changes the colour of the item as well as all the following items occurring before the next colour command. Conventions had to be established for sequencing taggable items and finding a particular item in the PDI file. Later versions of the system may employ Picture Manipulation Instructions (PMIs) to perform some of these operations in hardware.

Conclusion

The information preparation system is a key element in an overall public access information system. The Telidon concept will soon be tested in field trials across Canada using many communication means such as telephone lines, cable television channels, broadcast RF signals and even optic fibres. All these experiments will require the generation of extensive data bases encoded in PDI form. The information preparation system will be a useful tool for information providers to create these data bases.