An Implementation of Application Design Framework for a Dressed-up Touchscreen with a Tactile Feedback

Mayu yamanaka*, Makoto Miyauchi†, and Akira Yamawaki§

*Kyusyu Institute of Technology, 1-1 Sensui-cho Tobata-ku Kitakyuusyu-ku Fukuoka-ken, 804-8550, Japan
†Kitakyushu College of Technology, 5-20-1 Shii, Kokuraminami-ku, Kitakyushu 802-0985, Japan

Corresponding Auth: i108126m@gmail.com

Abstract

The products with the touchscreen are increasing while there are elderly or handicapped people who cannot operate touchscreen well. Therefore, the researches of the system supporting operations of a touchscreen by giving fingers a touch feeling become popular. We aim for the realization of the touchscreen that can provide the click feeling and the position of the button by a simple mechanism keeping the flexibility of the touchscreen (4-5). To be spread the welfare systems employing our proposal, providing a generic shared developing environment over applications is required. The developing environment must be a simple mechanism and less dependence to the device. We have proposed the hardware framework with a simple structure[4]. In addition, we have described a concept of the general software framework on this hardware framework[5]. However, since the software framework was a conceptual image, we need further examination to practice implementing the many features that are independent on the specified devices. For example, these features are the user interface and the switching of the application.

In this study, to realize the device-independent software developing environment, we employ GTK(The GIMP toolkit) as the infrastructure of user interface. We also develop the touchscreen system on the Linux to hide the real devices. We develop the touchscreen library combining with the GTK library and the touchscreen library. The touchscreen library is a set of low level functions that cover the details of the real device. In this paper, using their libraries, the study case was implemented the software framework on the prototype. Further, this paper verified that the application actually can be developed using the software framework.

Keywords: touchscreen, welfare, click feeling.

1. Introduction

The products with the touchscreen are increasing while there are elderly or handicapped people who cannot operate touchscreen well. Therefore, the researches of the system supporting operations of a touchscreen by giving fingers a touch feeling become popular. The most famous method is to vibrate the whole touchscreen (1-2). But, generated vibrations are not the same as the sense of the button switch.

In another research, there is the method which expresses the click feeling by charging the peculiar film putting on the touchscreen(3). The various feelings can be expressed by changing charged time. However, it is difficult to recognize the position and boundaries of the icons by the finger.

Therefore, we aim for the realization of the touchscreen that can provide the click feeling and the position of the button by a simple mechanism keeping the flexibility of the touchscreen(4-5). To be spread the welfare systems employing our proposal, providing a generic shared developing environment over applications is required. The developing environment must be a simple mechanism and less dependence to the device. We have proposed the hardware framework with a simple structure[4]. In addition, we have described a concept of the general software framework on this hardware framework[5].

However, since the software framework was a conceptual image, we need further examination to practice implementing the many features that are independent on the specified devices. For example, these features are the user interface and the switching of the application.

In this study, to realize the device-independent software developing environment, we employ GTK(The GIMP toolkit) as the infrastructure of user interface. We...
also develop the touchscreen system on the Linux to hide the real devices. We develop the touchscreen library combining with the GTK library and the touchscreen library. The touchscreen library is a set of low level functions that cover the details of the real device. Using these libraries, we examine how the concept of the software framework is implemented. In order to be switched each application easily, the control part is developed. The control part is specifically separated from each application. The control part works on the touchscreen and switches each application by outside events.

The rest of this paper is organized as follows. Section 2 is the overview of the hardware framework and the software framework. Section 3 shows the touchscreen library covering the details of the touchscreen. In Section 4, the implementing method of an application is explained along with the software framework shown in Section 2. In Section 5, the control part that is dissociated each application is explained. In Section 6, an application is actually developed. It is the ATM application by the implementing method we define. The applicability to practical application and the operation functionality are inspected. Finally, Section 7 concludes this paper.

2. Framework

2.1 Hardware Framework

Fig. 1 shows the hardware framework of the proposed touchscreen. The feature of the proposed hardware framework is that the click feeling and the position of the button are presented by simple mechanisms. The mechanism consists of mechanical switches and a plate that is hollowed out the position of icons.

The plate is put onto the top of the LCD panel. The mechanical switches are put underneath the LCD panel. When a finger pushes the LCD panel, the switches underneath the LCD panel are also pressed. Therefore, this proposed touchscreen give the user's finger the click feeling that is the same sense of the mechanical switch. Since the icon is actually pushed by the pressing of the switch, the user can prevent an incorrect input caused by just touching.

There are many plates. Each plate corresponds to the application. The plates have each identification number (ID). The ID corresponds to the application working on the touchscreen. When the plate is put on the LCD module, the tag reader read the ID and the concerned application is started from the storage device. The plate has the hollowed out parts where the icons are displayed on the LCD module. When this plate is located on the LCD module and the user touches this module, the edge parts of the icons give unevenness to the user's finger. Therefore, the user can recognize the position of an icon.

2.2 Software Framework

Fig. 2 is the concept of the software framework working on the hardware framework shown in Fig. 1. In this software framework, the applications and the control part of the touchscreen are obviously separated to be able to switch the applications easily.

When the plate is switched, the control part recognizes the ID and loads the targeted application from the storage device. Then, the application is performed.

The application has the structure of the state transition of the screen hierarchy. Each state of this structure corresponds to the screen. According to state transition, an action in response to the transition during the screen hierarchy, the drawing of buttons and background, and the switching of the button are performed.

The concept of software framework can be implemented by the tree structure shown in Fig. 3. Tree-tracking program in Fig. 3 performs the touchscreen application tracking bidirectional tree. This tree is the common concept over applications. Each node of the bidirectional tree corresponds to the state, i.e., one screen. The top node means the screen 0. The screen 0 is the top screen of a touchscreen application, which is displayed first. Nodes have the pointer to the next screen (top, back, and p[i]), the pointer to an execution program of a screen (program), and an input data to an execution program (data[N]). The execution program performs the role of the screen displayed at the screen hierarchy. By replacing the program pointed, an execution program of the screen can be
easily changed. Each node can have any number of the pointers to the branch destination nodes. In order to be able to turn back the state transition, each node has the pointer the screen 0 (top), and the pointer to just before the screen(back).

3. Touchscreen Library

3.1 GTK+

GTK+[6] is a library to create a GUI (Graphical User Interface) application. Originally, GTK was developed to create an image editing application called GIMP. However, at present, GTK is used to create various applications in addition to GIMP and is also used as a tool kit for the GNOME desktop environment.

GTK+ is the multi-platform library working on Windows and Mac OS X as well as Linux. Further, GTK+ is the library written in C language. However, in order to be able to use the programming languages except for C language, various bindings such as C++, C#, Python, Ruby are published.

3.2 Application Program Interface

According to Fig.2, the basic library is provided to develop an application. Its library realizes the touchscreen function. A set of low level functions was developed combining Linux with GTK. Their functions are classify as functions for the control part, for applications, and for the tree-tracking program, and are as follows.

(For the control part)

- Plate_Tag_Receive(An array that stores the ID of 16 numbers) is the function for receiving the ID of 16 numbers from a tag reader.
- ID_Check(An array of the received ID) compares the received ID with the pre-registered plural IDs. Whether the received ID corresponds with the pre-registered IDs is output.

(For the screen of an application)

In the screen of an application, the basic part such as the creation of the window and an arrangement of the button and label uses functions of the GTK.

- Get_Touched_Point(the pointer of the x-coordinate, the pointer of the y-coordinate) gets X-Y coordinates of the touched part on the LCD screen.
- Recv_Data(None) receives the candidate pointers of the transition destination and the input data from the tree-tracking program.
- Send_Data(the input data, the pointer data of the transition destination) sends the input data and the pointer of the transition destination to the tree-tracking program.
- Display_Button(the widget, the data designated by...
4. Application program

The application program consists of the tree-tracking program and screen programs as shown in 2.2.

The tree-tracking program administers the state transition of the screen hierarchy. The screen programs implement an action and the position of the icon regarding each screen. Because the screen programs store the program designated in each node, the screen programs change when every node is transitioned.

Between the tree-tracking program and the screen program, the pointers and an input data are sent and received. Their pointers determine the next transition destination. The input data is used to perform the screen action and determine the state transition. These pointers and the input data are integer arrays. In their data passing, inter-process pipe is used. Because the FIFO file mediates between two programs, the data passing becomes possible.

Next, an internal operation of the tree-tracking program is explained. In tree-tracking program, first, the node designated by the current node pointer is read. Then, the screen program stored in its node is started. The tree-tracking program writes the input data and candidate pointers of the transition destination to the FIFO file. Thus, the input data and their pointers can be sent to the screen program. The tree-tracking program waits the screen program to be finished. When the screen program is finished, the node pointer of the next transition destination and the input data to send to its transition destination are written to the FIFO file. Then, the tree-tracking program reads out its pointer and its data from the FIFO file. When reading is complete, the input data of the transition destination node is renewed for the received input data. Further, the node pointer is overwritten to the node pointer of the transition destination. Then, the tree-tracking program transitions to the next node.

Finally, an internal operation of the screen programs is explained. When a screen program is started, the input data and pointer data of the transition destination candidate are read out from the FIFO file. Using input data reading out, the main program is run, and the next transition destination is determined. Then, the screen program selects the determined transition destination from pointers of the candidate transition destination. In addition, input data sending its destination is created, and its data and the pointer of the transition destination are written to the FIFO file.

5. Control Part

In the control part, regardless of applications, actions required to touchscreen are stored. The control part performs the following actions.

1. Read the identification number by the tag reader.
2. Compare the registered ID and the read ID.
3. Send screen information to the LCD panel corresponding to the state transition.

A set of these actions show Fig.4. Firstly, the information processing part waits to be sent the ID from a tag reader. When the ID is sent, the information processing part receives the ID. Next, the registered ID and the received ID are compared. The information processing part makes sure whether the received ID matches the registered ID. If the received ID doesn’t match the registered ID, the processing part waits to be sent the ID from the tag reader again. If the received ID matches, the program of the matched application is transitioned. Then, the processing part sends the screen information to the LCD panel depending on the state transition of the application. The LCD panel displays the received screen information.

![Flow diagram of the control part](image-url)
6. Case study

6.1 Prototype System

The structure of the prototype is explained. The block diagram of the prototype shows Fig.5. The LCD module is consisted of a 7-inch wide touchscreen and a 7 inch wide LCD panel. The 7-inch wide touchscreen has a resolution of 800 x 480. AD converter for the touchscreen is consisted of Texas Instruments ADS7843.

This time, the RFID is used as IDs of the plates. ID Innovations ID-12 is used as the tag reader. OMRON B3F5000 is used as the switch.

Raspberry Pi that is a small-board computer is used as the computer. The RFID and switches connect to Raspberry Pi. A 7 inch wide LCD panel mediates a HDMI LCD foundation equipped LCD controller chip RTD2668 made of Realtek company and connects to Raspberry Pi. In the same way, the 7-inch wide touchscreen mediates AD converter, and connects to Raspberry Pi.

![Fig.5 The block diagram of the prototype](image)

6.2 The development of the AMT application

The structure of the ATM system is explained. The created ATM actions are showed in Fig.6. As the ATM system is explained in the chapter 4, the ATM system is performed using the tree-tracking program and the screen program.

As the operation example of the ATM system, Fig.7 shows the flow from the designation of the withdrawn money to the confirmation of its money. When the personal identification number is correctly entered, the state is transitioned to the node of the designation of the withdrawn money. In this screen, the user can designate a sum of the money withdrawn in the ATM. Firstly, when the current node is transitioned to the node of the designation of the withdrawn money, the tree-tracking program starts the screen program that is stored to the current node. Then, in the screen program, the program of the designation of the withdrawn money is started, and waits to be received the input data and the candidate pointer of the transition amounts credited is confirmed, and they are correct, the operation is finished. If the operation is canceled, the screen returns the TOP screen. Further, after amounts credited is confirmed, if the user want to deposit again, the screen can return the bill depositing screen. In the withdrawals, if the personal identification number is input, and its number is correct, the ATM system transitions to the screen that is designated a sum of the money. The user designates a sum of the money, and confirms whether this money is correct. If its money is correct, the action is finished. If the user wants to cancel on the way, and the personal identification number is an error, the current screen can return to the TOP screen. Further, if the sum of the money to want to withdraw is the error, the current screen can return the previous screen to designate again.

![Fig.6 The flow diagram of the ATM application](image)
destination from the tree-tracking program.

After the program of the designation of the withdrawn money was started, the tree-tracking program sends the input data and the candidate pointers of the transition destination received from the previous screen. When the sending is finished, the tree-tracking waits to be received the input data and the pointer of the transition destination received from the screen program. On the other hand, when the screen program receives the data and the candidate pointers received from the tree-tracking program, the program of the designation of the withdrawn money is performed using the received input data. While the program of the designation of withdrawn money is performed, a sum of the withdrawn money and the transition destination are determined. In this time, the state is assumed to transition to the confirmation screen of the withdrawn money. The screen program stores the withdrawn money as the input data. In addition, using the candidate pointers of the transition destination, the program of the designation of the withdrawn money selects the pointer to the confirmation screen of the withdrawn money. The input data and the pointer of the transition destination are sent to the tree-tracking program, the program of a sum of the designation money is finished.

Next, the tree-tracking program receives the input data and the pointer of the transition destination, and renews the input data of the confirmation screen of the withdrawn money that is transitioned to next. Then, the node pointer is overwritten the node pointer the confirmation screen of the withdrawn money. The state moves to its node. These actions are performed each node.

Fig.7 The flow diagram of the designation money

6.3 The verification results

The ATM application showed in 6.2 is performed. The verification results are as follows.

Fig.8 The TOP screen
Fig. 9(a) The screen of the personal identification number

Fig. 9(b) The screen of a sum of the designation money

Fig. 9(c) The screen of the confirmation of a sum of the money

Fig. 9(d) The screen of the finish

Fig. 9 The operation verification of the withdrawals

Fig. 10(a) The screen to put the bill

Fig. 10(b) The screen to confirm the money

Fig. 10(c) The screen of finish

Fig. 10 The operation verification of the deposits

Fig. 8–Fig. 10 shows the operation examples of the ATM. In this way, the operations of deposits and withdrawals are performed. Consequently, it is possible to create applications in the developed software framework.

7. Conclusion

In this paper, the implement of the software framework in the Dressed-up touchscreen with a tactile feedback was proposed. Through the ATM application, the case study showed that it is possible to develop applications using the proposed framework. In the future, the proposed software framework adapts the multiple applications and the proposed touchscreen is inspected usability to some testees.

Reference


