UI Software Development for Digital Recorders with the UI Design Tool "NINA"

Authors: *Nobutoshi Todoroki** and *Yoshihiro Yamaguchi***

Introduction

The cost of developing user interface (UI) software has continued to rise as embedded devices become more functional and UI designs become richer. We have been developing the UI design tool "NINA (Navigator for Interface of Application)" to improve the efficiency of UI development for such embedded devices and reduce the cost. This report discusses the application of NINA to the UI development of a digital recorder and its successor models, and evaluates its effectiveness.

1. UI Design Tool "NINA"

"NINA" is a UI design tool for embedded devices; it consists of several functional modules used on PC and a software module for target device (Fig. 1). Although embedded devices have numerous functions and setting items, the screen size is often limited. Therefore, the UI for embedded devices is generally operated by switching the screens for each particular procedure. "NINA" is used to design the UI based on the concept of SCO (State Chart Object), which is useful for modeling UI involving such switching of screens.

1.1 SCO

SCO is a UI component that consists of more than one state and transitions between states. Each state has one scene layout. The scene layout that corresponds to the SCO state at a certain point is displayed as the external appearance of the SCO at that point.

In each scene layout, other SCOs can be laid out as UI components in addition to the basic UI components, such as buttons and labels. This enables the SCOs to be made hierarchical, which allows the designer to design all custom UI components on screen to the UI of device application in the same framework, and then operate them by combining them.

Figure 2 shows an example of the hierarchal use of SCOs. SCO1 has two states – stop and play of music – and when a button is pressed, the device transfers to the other state. In the scene layouts of each state, the UI screens that correspond to "stop" and "play" are created. SCO2 achieves the UI that displays the playback screen when a track to be played is selected. On the playback screen, SCO1 is laid out in the scene layout as a custom UI component. Modifying the design of SCO1 in such designing also automatically changes the playback screen of SCO2.

1.2 Components

An overview of the major components of NINA is given in Fig. 1. The chart editor is a tool for designing the transitions between the UI screens as state diagrams. It also describes the procedures in response to key events during device operation and other events generated in the device, as an event handler. The event handler is written using a Java-based simple script language (Java is a registered trademark of Sun Microsystems, Inc.).

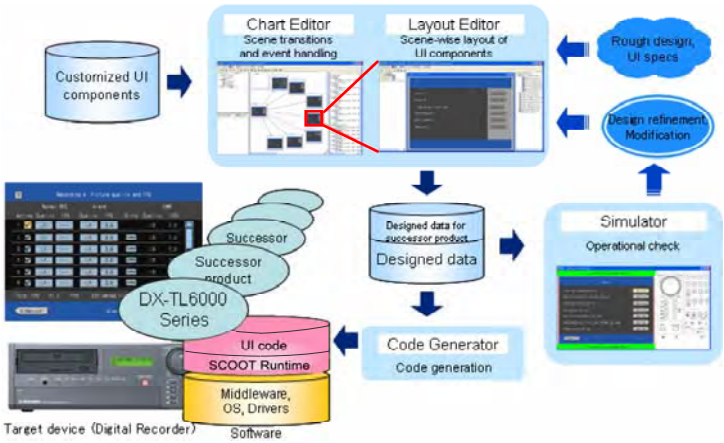

Fig. 1 The structure of NINA and the UI development flow

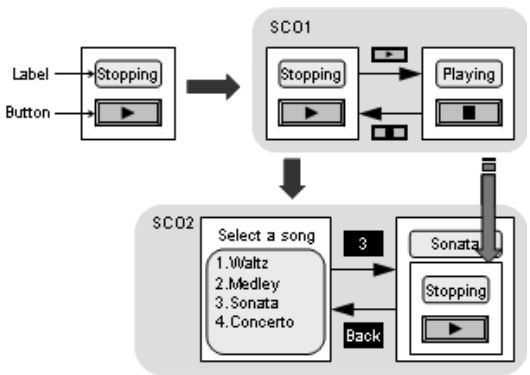

Fig. 2 Hierarchical use of SCOs

*Advanced Technology R&D Center **Kyoto Works

The layout editor creates a scene layout in each state of the SCO. A scene layout combines the SCOs and the basic UI components, including labels, buttons, images, and containers that group the UI components.

The simulator is a tool which simulates the UI operation designed using the chart editor and layout editor, allowing the designer to check it on a PC used for development.

The code generator uses the UI design data to generate C++ source code for target devices: The generated codes do not include any platform-dependent process and do not need to be modified by the programmer.

The SCOOT (SCO Oriented Technology) runtime is a software module written in C++. It runs the source codes generated by the code generator on a target device.

2. UI Development of Digital Recorder

We used NINA for the UI development of our DX-TL6000 digital recorder for Japan. Based on the UI, we have developed UIs for six successor models such as DX-TL308E, including models for Europe. An overview of the digital recorder and the UI development of these models using NINA is given below.

2.1 Overview of digital recorder

The digital recorder is a business-use surveillance device which is mainly embedded in the security surveillance system of retail stores and banks. It digitally records surveillance images for a long time.

The latest DX-TL6000 models use a USB mouse and have a simple menu so the user can operate the recorder like a PC. The features of this UI are as follows:

(1) The user can operate the UI using both the front buttons on the front panel and the USB mouse. The front buttons include the six operation buttons: up, down, left, right, enter and back. These buttons move the focus that indicates the UI component that the user is currently manipulating or scroll it, thus selecting the item on the screen and operating "enter/back" to move between the screens or set values. The user can also use the USB mouse to move the cursor and click it just like using a PC.

(2) A digital recorder has particular operations – Focus management in the vertical and horizontal directions, defined in a complicated manner, and nested focus management. The following focus management is performed on the screen shown in Fig. 3:

- Pressing the enter key when a block-shaped UI component is being focused will move the focus to the UI component within the block. In the block, the focus will move by operating the left and right keys.
- Pressing the enter key when a UI component in a block is being focused will move the focus to the entire block.
- Clicking the mouse on each UI component will directly move the focus to the component.

2.2 UI development by using NINA

To achieve the UI described above, the following UI components have been developed:

(1) Extending basic UI components

The basic UI components have been extended so that they can be operated by both pressing the front buttons and clicking the mouse.

(2) Designing custom UI components for digital recorders

Complicated focus management can also be achieved by describing it using script language. How-

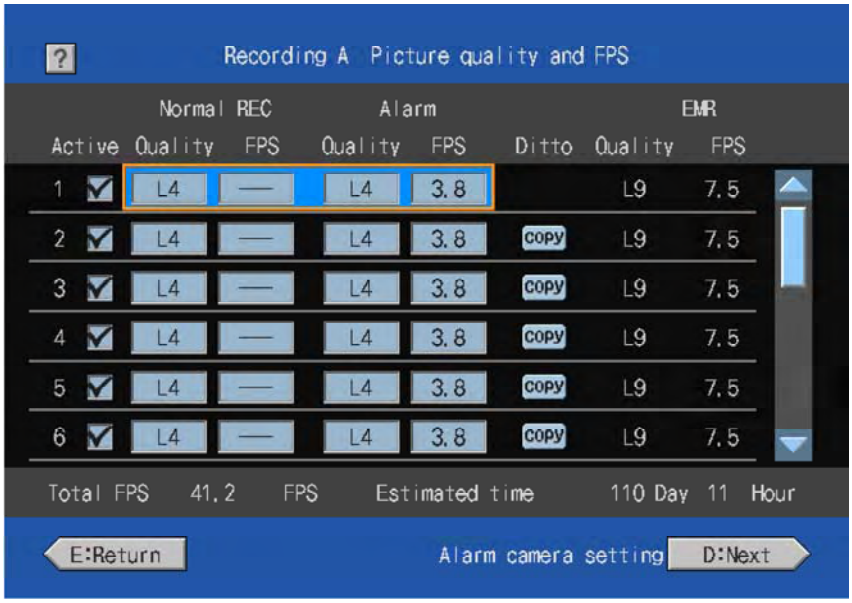

Fig. 3 A setting menu

ever, there are many similar screens, so many clones of code are generated, which increases the quantity of coding and makes it difficult to cover all changes when the specifications are to be changed. To prevent this, we offer custom UI components that include each focus management as the UI design using the framework of SCO. This component can be laid out in the same way as a basic UI component on various screens. Even when the focus management specifications are changed, simply revising this UI component will be reflected in all screens that are using the UI component.

The UI was developed by developing this UI component. As a result, a 580-screen UI was developed in 16 man-months.

2.3 Developing UIs of successor models using NINA

The UI development of six DX-TL6000 successor models is outlined below. The major changes in UI specifications from the DX-TL6000 are as follows:

(1) There are fewer cameras to be controlled and fewer options for various settings.
(2) The video signal used in the models for Europe is different: the screen display area (in the vertical direction) is 1.2 times that in models for Japan.

For (1), the scene layout for fewer options was also added to the UI component that displays the setting items, and each layout is selectively displayed corresponding to the model. For (2), we provided a new basic UI component for which two types of layout parameter can be set and the layout can then be switched for display. Simply replacing the previous basic UI component with this new one allows the display to handle two types of layout using the one screen.

Based on these change strategies, the UI of six successor models was developed. This encouraged reuse of the UI design, and the development was completed by partially revising and adding the definition of UI design, taking only 2.6 man-months.

2.4 Evaluation of application of NINA

With the DX-TL6000, the types of screen increased considerably compared with previous models, and the specifications became much more complicated because the mouse and operation buttons were used concurrently. Nevertheless, the UI was achieved using only half the man-hours of the previous UI. This section describes the evaluation of this achievement, and also considers reusing the UI design in developing future successor models.

2.4.1 Reducing the number of created screens by common use

One common screen that can be used as a base is created, and UI components are then added or deleted as necessary. This allows the designer to express many screens, and this common use of screens substantially reduces the number of created screens in the UI specifications with many similar screens. We analyzed the UI specifications prior to development and tried common use of screens (Table 1). The number of actually created screens was 253, so we successfully reduced the number of screens to 44% of that defined in the UI specifications.

Table 1 UI development cost of DX-TL6000

	Development man-hours	No. of specified screens	No. of created screens
Development scale	16 man-month	580	253

2.4.2 Increasing development efficiency by introducing custom UI components

With the screen in Fig. 3, we compared the man-hours required to describe all focus movement/scroll movement using a script language, and when using a custom UI component. In the latter case, the man-hours for the focus portion were reduced by 65% and the man-hours for designing the UI were reduced by 39% for the entire screen in Fig. 3. Therefore, the man-hours required for the product as a whole will be further decreased.

2.4.3 Increasing the efficiency of development by using a simulator

If a detailed test is conducted in a prior work process using the simulator function of NINA, faults in the post-work process can be reduced. Table 2 shows the number of faults in the UI portion in the simulator test and system test: it is clear that, among the faults that can be detected in the whole work process, 90% were detected in the prior work process.

Table 2 Simulation test and system test results

	No. of test items	No. of detections
Simulator test	6100	487
System test	20000	53

2.4.4 Increasing the efficiency of development of successor models

When developing six successor models, we modified the UI design of the DX-TL6000, using it as the base. Table 3 shows the number of SCOs that could be reused in the development of these models. SCOs accounting for 63% of the entire design could be reused for designing the DX-TL6000; the development man-hours for the remaining 37% were 2.6 man-months.

Table 3 Number of reused SCOs and newly developed SCOs

	Total No.	No. of reuse	No. of new/additional creation
No. of SCO	270	170	100

3. Conclusion

The UI design tool NINA was used to develop the UI of digital recorders. Since the framework of reusing the UI design was used, the UI development efficiency was improved when developing an initial product and successor models. We will continue to develop successor models in the future.