Generating UI for Pervasive Devices Using Pattern-Based Approach

Deng-Jyi Chen*, Ming-Jyh Tsai*, Chung-Yuan Huang**
* Department of Computer Science
National Chiao Tung University
1001 Ta Hsueh Road, Hsinchu, Taiwan, 333, ROC
** Department of Computer Science and Information Engineering
Chang Gung University
259 Wen-Hwa 1st Road, Tao-Yuan, Taiwan, 333, ROC

Abstract: - It has been shown that the major effort spent on the design and implementation of the system software for pervasive devices (or handset device such as cellular phone) is the user interfaces (UI). If UI can be developed in a short time, it can be a great help to reduce development time for application software system. Therefore, many researchers in software engineering area have been seeking better solutions to aid UI designers to create UI.

In this paper, we propose a pattern-based approach to generate UI for pervasive device. Specifically, a UI design patterns generator is proposed for UI designers to easily and quickly create the UI patterns for cellular phone. Furthermore, the developed UI patterns can be fine-tuned with a visual UI authoring tool to generate the user look and feel of the target cellular phone system under consideration. The generated user look and feel is then as a guider for the program generator to glue the software design framework and associated functions together to produce the target application system code. Finally, In order to demonstrate the feasibility and applicability of the proposed UI design patterns generator, a simulator is designed and implemented for carrying out the software simulation.

The benefit of the proposed approach is that it enables UI designers to generate user look and feel easily and quickly, and produces automatically the target UI program without writing any textual code. Thus the proposed approach is very suitable for the UI designers (nonprogrammers). In addition, the developed UI pattern can be reused by UI designers to generate target user look and feel. Therefore, it can reduce development time.

Key-Words: - UI design patterns generator, Generic UI template, User interface (UI), Visual UI authoring tool, Program generator, Simulator.

1 Introduction

A well-designed User Interface (UI) encourages easy, natural, and engaging interaction between a system and its users. As interfaces have become easier to use over the past 30 years they have also become harder to design and implement [11][12]; accordingly, sophisticated UI development for any application is now considered a complex and time-consuming task. Studies have shown that almost 80 percent of all application code deals with UI-related issues [5]. Software engineers are constantly searching for new solutions to UI system development problems.

In general, UI designers and UI programmers have much iteration process to go in order to meet the target UI requirement specification. To reduce the UI designer’s heavy workload in this much iterative process, we propose a UI design pattern generator for UI designers to easily design and author the UI pattern. These created UI pattern can be reused easily to create the user look and feel. To avoid UI programmer’s tedious workload in this long iterative process, the UI design patterns generator can automatically generate the UI program according to the UI pattern. Based on this innovative approach, the UI designers alone can complete the UI design and implementation without bothering the UI programmers.

There are a few specific tools available for creating the UI for pervasive devices based on the visual authoring approach. The most common tools found in industrial sectors for the UI development of pervasive devices are embedded Visual C++ [9], Rapid [17], and Symbian’s C++ [15][16].

Consequently, the following problems will be faced when using the above mentioned UI developing tools:
1) Writing textual program is still inevitable.
2) UI Programmers have to work with UI designer in order to modify the changes of UI. (UI designer alone cannot complete the UI task)

---

1 This research was supported by the National Science Council of Taiwan, the CAISER of National Chiao-Tung University, and the Bestwise International computing Company of Taiwan.
3) A much iterative process between UI designers and UI programmers cannot be avoided while using the current approaches to design and implement the UI of the pervasive devices.

4) No UI patterns generator supported in current UI developing tools. Thus, a UI designer could not use it to generate various UI pattern for future reuse. We propose a UI design patterns generator to generate the user look and feel of mobile phone. The proposed UI design patterns generator has integrated into the visual Based software construction methodology. The benefit of the proposed approach is that it enables UI designers to generate user look and feel easily and quickly, and generate the target UI program without writing any textual code. Thus the proposed approach is very suitable for UI designers. In addition, the created UI pattern can be reused by UI designer to generate target user look and feel. Therefore, it can reduce UI application system’s development time.

2 The framework of UI design Patterns Generator and Visual-Based Software Construction Model

The Framework of UI Design Patterns Generator and Visual-Based Software Construction Model is shown in Figure 1.

Generic UI Template: It is consisted of UI structure template, UI layout template, and UI style template which will be elaborated in section 4.2.

UI Template constructor: A tool for making new UI Templates. It instantiates the Generic UI template to construct the UI pattern and then stores it into UI patterns database through the UI template manager.

UI Template manager: A database management system for managing UI patterns; it provides an interface for adding or deleting UI patterns as well as for retrieving an existing UI pattern.

UI patterns database: A database for storing UI patterns.

Software design framework: The architecture framework for the target application system. This framework is generated by generic software framework for pervasive devices which is not emphasized in this paper.

Associated function: Associated function developed based on Application Program Interface (API) library function developed by the functional programmers according to the hardware specification.

Simulator: It is used to simulate the functionalities of the produced target application system on the target cellular phone.

UI designers use the UI design patterns generator to construct an initial UI pattern, and then use the visual UI authoring tool to modify or fine tune the initial UI pattern to generate the target user look and feel of target cellular phone system. The generated target user look and feel is then as a guider for the visual program generator, the function binding system, to glue the software design framework and associated functions together to produce the target application system code. Finally, one uses the simulator to do software simulation.

The benefit of the UI design patterns generator is that it enables UI designers to create user look and feel easily and quickly, and produces the target UI program without writing any textual program. Thus this generator is very suitable for UI designers.

3 Generating UI patterns Using UI Design Patterns Generator

3.1 UI of Mobile Phone

When we operate a mobile phone, we will see the stand-by screen after power on the mobile phone. Then,
there will be several functional buttons ready for pressing to initiate the desired function. To press a specific functional button, it takes us to the corresponding screen associated with the function. These functional buttons are usually organized into a tree style structure and consists of two basic elements 1) Node that represents a screen or a function and 2) Link that defines the relationship between screen and screen or screen and function.

A screen (or node), which is not a leaf, can be considered as a container (or scene) that may contains many actors (or UI components) including, text actor which provides function of text representation, Icon actor which provides function of drawing representation, input box actor which allows user to input data during execution of a specific function such as pressing telephone number, and list box actor which represents function of multiple data. A link provides a binding among nodes and functions, and control information for a screen to another screen (node to node) navigation.

A screen or node at leaf level will be considered as a function. There are many common functions in most of the current mobile phones including the essential functions (such as communication, phone books, and conversation records) and value-added functions (such as recording, camera, Java Game, infrared transmission). These functions usually are implemented by application programmers.

Thus, the UI of mobile phone can be quite different if the UI navigation structure is different, the layout of actors in a container (or scene) is different, and the style of actors (leave node) is different.

### 3.2 Generic UI Template

After comparing the UI of different mobile phones, we find that there are some similarities of appearance of user interface when these mobile phones are made by the same manufacturer. Nevertheless, even though the mobile phone comes from different manufacturers, the structure of the UI also has some similarity. Therefore, we factored out the common parts from various UI structures, screen layouts, and actor styles to define the generic UI template. These will be defined as structure template, layout template, and style template.

#### 3.2.1 Structure Template

The user look and feel is very dependent on the UI structure for screens to screens navigation. The UI structure, based on topological point of views, can be a ring (or circle list), tree, and ring of tree.

#### 3.2.2 Layout Template

The user look and feel is also sensitive to each actor’s position in a scene. This positioning is called Layout. We could define layout template according to the layout information and then change the actor’s position according to a different layout template.

#### 3.2.3 Style Template

The other factor that affects the user look and feel of a mobile phone is the UI style. The UI style considers the style of actor’s appearance. An actor’s appearance is decided by its attributes in a scene. Even though it is the same actor, different attributes may produce different appearances.

### 3.3 Generic UI Template for Pervasive Devices

In previous subsections, we have defined three UI templates (structure template, layout template, and style template). After combination of these three templates, we obtain a generic UI Template. Furthermore, we use MVC models (model, view, control) [7] to implement a generic UI template for the UI pattern generation of the mobile phone device. The structure template produces model, layout template, and style template provide view. It could generate different user look and feel by different combination of these three templates. Structure template generates UI structure first, layout template offers layout data of actor on the scene, and then style template provides each actor’s style.

### 3.4 UI Patterns Generation Procedures Using UI Design Patterns Generator

How to use the UI design patterns generator to create different UI patterns is summarized below.

**Step 1:** Select a generic UI template

**Step 2:** Generate the complete UI structure from the chosen structure template

**Step 3:** Generate the desired layout based on the information in layout template for each scene (or node in the tree structure)

**Step 4:** Generate the desired style based on the information in style template for each actor (or UI button) on each screen (or scene)

**Step 5:** Repeat step 3 and step 4 till all the scenes and actors chosen from the structure template are defined

The system continuously repeats the actions in step 3 and step 4 till all the user look and feel required layout and style information has been filled into the chosen structure template.

After executing the above steps, we can construct the UI pattern for the handset device under consideration. Next, we can fine tune the created UI pattern by visual UI authoring tool to generate target user look and feel. Then, we use program generator to binding the
4 The Proposed Pattern-Based Approach Assessment

4.1 Purpose
The purpose of this experiment is to explore the efforts spent with the use and without the use of created UI patterns to produce the target user look and feel for pervasive devices. Three application examples contain six, twelve, and eighteen UI screens were used to make such a comparison. On average, there are about 7 UI components for each screen.

4.2 Participants
Study participants were 39 students enrolled in a software engineering class in a Taiwanese graduate course. The students’ majors were computer science (29), electrical engineering (6), information management (2), and other computer-related departments (2). Those students who participated in the experiment all have rich programming experience (at least three years working experience in software programming related area).

4.3 Design
All the students were asked to be familiar with visual UI authoring tool, and use it to generate the target user look and feel for a handset device. Then two weeks later, those students were asked to use the created UI pattern to do the same task.

4.3.1 without the use of created UI pattern to generate target user look and feel
If there is no suitable UI pattern in the UI patterns database can be used directly, we have to use a visual UI authoring tool to generate the target user look and feel.

4.3.2 Using the created UI pattern to generate a target user look and feel
If there is suitable created UI pattern, we can modify it into the target user look and feel by a visual UI authoring tool. This modification includes (1) Replacing icon and text from one of the screens; (2) Deleting some screens; (3) Creating one screen to the created UI pattern.

4.4 Data and Analysis

4.4.1 Case 1: Development Time of Generate a six-screen user look and feel
Fig. 2 presents the time spent to generate a six-screen user look and feel with the use of created UI pattern and without the use of created UI pattern. Based on the collected data, we found that the average time spent to generate this user look and feel with the use of created UI pattern is 9.92 minutes and the average time spent without the use of created UI pattern is 36.38 minutes. The development time without the use of created UI pattern is about 3.7 times [(36.38 / 9.92)] longer than with the use of created UI pattern. The 27-minute difference represents a time savings of 73%.

4.4.2 Case 2: Development Time of Generate a twelve-screen user look and feel
Fig. 3 presents the time spent to generate a twelve-screen user look and feel with the use of created UI pattern and without the use of created UI pattern. Based on the collected data, we found that the average time spent to generate this user look and feel with the use of created UI pattern is 10.38 minutes and the average time spent without the use of created UI pattern is 54.10 minutes. The development time without the use of created UI pattern is about 5.2 times [(54.10 / 10.38)] longer than with the use of created UI pattern. The 44-minute difference represents a time savings of 81%.
Fig. 3 Time spent in generating a twelve-screen user look and feel

4.4.3 Case 3: Development Time of Generate an eighteen-screen user look and feel

Fig. 4 presents the time spent to generate an eighteen-screen user look and feel with the use of created UI pattern and without the use of created UI pattern. Based on the collected data, we found that the average time spent to generate this user look and feel with the use of created UI pattern is 10.85 minutes and the average time spent without the use of created UI pattern is 80.05 minutes. The development time without the use of created UI pattern is about 7.4 times (80.05 / 10.85) longer than with the use of created UI pattern. The 69-minute difference represents a time savings of 86%.

Table 1 Average time spent in generating six-screen, twelve-screen, and eighteen-screen user look and feel

<table>
<thead>
<tr>
<th>User look and feel</th>
<th>Time spent with the use of created UI pattern (Minutes)</th>
<th>Average time spent without the use of created UI pattern (Minutes)</th>
<th>Time ratio between average time spent without the use of created UI pattern and average time spent with the use of created UI pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>six screens</td>
<td>9.92</td>
<td>36.38</td>
<td>3.7</td>
</tr>
<tr>
<td>twelve screens</td>
<td>10.38</td>
<td>54.10</td>
<td>5.2</td>
</tr>
<tr>
<td>Eighteen screens</td>
<td>10.85</td>
<td>80.05</td>
<td>7.4</td>
</tr>
</tbody>
</table>

From Table 1, one can see the time ratio increments approximately 1.5 times each time as we add six screens.

From Fig. 5 we can note that the average time spent without the use of created UI pattern sharply raises. But using the created UI pattern remains smoothly. From the observation, we can foresee that as the number of UI screens increased then the time ratio between those two approaches will become wider.

5 Conclusions
The well designed UI allows user to operate the product conveniently. So, the design of UI becomes very important which can give customers different impression on the mobile phone and influence their decision to purchase it or not.

UI plays an important role during software development. However, it is also time-consuming for creating a good UI. If UI can be developed in a short time, it can be a great help to reduce development time for application software system. Therefore, many researchers in software engineering area have been...
seeking better solutions to aid UI designers to build UI.

By the proposed approach, UI designers use the UI design patterns generator to create UI patterns and store them for future usage. Furthermore, it could be modified to be a custom-designed user look and feel by the visual UI authoring tool. Based on this innovative approach, the UI designers alone can complete the UI design and implementation without bothering the UI programmers since the UI program will be generated automatically by using the UI design patterns generator. With the assistance of the visual program generator, the UI component is equipped with associated function. We only need to bind the relevant system function with the UI component to generate the target application system code. Finally, we provide a simulator for software simulation.

The proposed generating UI for pervasive devices using pattern-based approach has following characteristics.

1) Ideal for UI designer (nonprogrammer) to produce the user look and feel for the target application software system.

2) The proposed approach has separated the UI system with the application function implementation. These two parts are developed by UI designer and application programmer separately. A system developer just needs to bind these two parts by using visual program generator. In this way, any future changes of the UI will not affect the corresponding application function implementation. Vice versa for the case that the UI is not changed but the associated application functions needed to be changed.

3) Long iterative process between UI designers and UI programmers can be avoided while using the proposed approach to design and implement the user look and feel of the pervasive devices.

4) Rich UI pattern can be supported by the proposed UI design patterns generator to ease a UI designer to reuse to create target user look and feel. Hence, it can reduce time spent while creating user look and feel.

References:
Guide for Engineers and Programmers, Newnes, 2005,