

XAML-based 2D/3D UI Prototyping System for Rapid Development of Information Appliances

Taiga Higuchi, Satoshi Kanai

Graduate School of Information Science and Technology, Hokkaido University,
Kita-14, Nishi-9, Kita-ku, Sapporo 060-0814, Japan
Phone 011-706-6449 / Fax 011-706-7120
E-mail : t_higuchi@sdm.ssi.ist.hokudai.ac.jp
kanai@ssi.ist.hokudai.ac.jp

Abstract: Recently, information appliance manufacturers have become very willing to introduce prototyping methods for the efficient development of User Interface (UI) software. To realize this possibility, we propose XAML-based UI software specifications for information appliances and develop 2D/3D UI prototyping systems based on these specification. 2D/3D UI simulation can be executed uniformly in the newly developed system.

Key words: user interface, prototyping, information appliances, XAML.

1- Introduction

The functions of information appliances are currently increasing in their complexity. Manufacturers of products such as digital cameras and mobile phone have begun paying attention to UI prototyping. So far, two different types of UI prototyping systems have been developed: 2D and 3D.

There are already some existing commercial and non-commercial 2D UI prototyping systems; UIML[A1], d.tools[H1], RapidPlus [R1]. In these systems, the static UI structure of the screen images and the dynamic behavior of the screen transition are explicitly modeled. It is therefore relatively easy to use the trial-and-error design process in prototyping. However, the reality of the UI simulation is degraded because the 2D UI prototype substitutes a 2D view for the 3D housing. The patterns of missed operations acquired using the 2D UI prototypes are not necessarily the same as those of the real product.

Alternatively, 3D UI prototyping systems are mostly implemented by remodeling the player tools VRML or Web3D. These systems can thus produce more realistic UI simulations of user interaction. However, in the 3D systems, the static UI of the screen images and the dynamic behavior of the screen transition are not modeled explicitly and in a declarative way. So a large number of texture images displayed on the screen must be built for the 3D UI simulation in the tools.

Therefore, in this paper, we propose a new UI prototyping model that combines a XAML (eXtensible Application Markup Language[X1]) model, which produce a 2D/3D geometry model of the housing and structure of UI screen images, with another declarative model contributing the dynamic UI behavior of the screen images. A 2D/3D UI prototyping system based on the proposed models is also

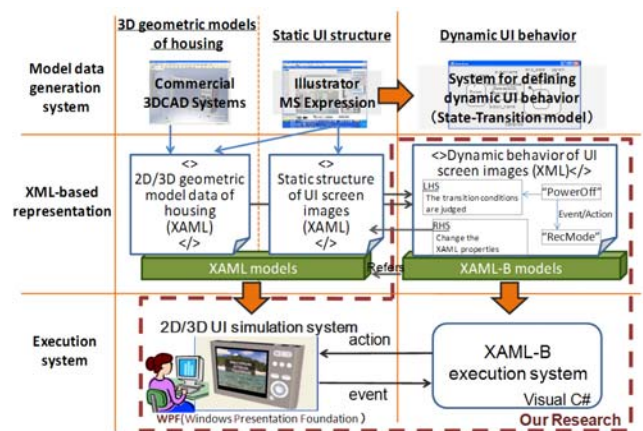


Fig.1 : Proposed process for developing XAML-based UI software

developed.

The contributions of this paper are to make the trial-and-error design process in 2D/3D UI prototyping easily by describing the UI structure and behavior models declaratively and to enable UI designers to easily develop 2D/3D UI prototypes in a standard desktop graphic architecture.

2- Proposed method and necessary conditions

2.1 – XAML and its issues

XAML was proposed by Microsoft as a UI software specification targeted for Windows applications. XAML is an XML based mark-up language which specifies the static structure of a UI running on the WPF (Windows Presentation Foundation). The WPF is the GUI component that describes the static structures of 3D geometric objects and 2D screen images. In the basic development process of UI software using XAML, the static structure of the UI screen images is first described in XAML, and UI dynamic behaviors such as events and actions are then directly coded in Visual C#.

However, it is difficult for most UI designers to build operable UI prototypes using XAML because prototyping requires direct coding in the programming language. The redesign of UIs requires extra time and money. To solve this problem, we first proposed a new model called XAML-B, wherein dynamic UI behavior can be defined in the form of

XML documents. In addition, we then developed an execution system wherein the dynamic behavior described by XAML-B is interpreted and executed. The system enables users to generate both 2D and 3D UI simulations.

3- XAML-B model and its execution system

3.1 – Proposed process for developing UI software and XAML-B model

Fig. 1 shows the proposed development process for XAML-based UI software and the models used in the process. The State-Transition model is effective in the upper process in describing dynamic UI behavior. However, if we specify behavior using a State-Transition model, the description in the lower process tends to cause state explosion. Additionally the XAML specifications do not have a notion of ‘state’ but do have one of ‘event.’ Therefore, in our research, in order to design dynamic UI behavior, the State-Transition model is used in the upper process, and an event-based model is used in the lower process. The proposed event-based model of UI behavior is an ‘XAML-B’ model which is interpreted in the XAML-B execution system.

3.2 –Description of UI behavior using XAML-B model

The event-based XAML-B model refers to UsiXML[L1], which is an XML-compliant user interface description language with various interaction modalities. As shown in Fig. 2, the event-based model consists mainly of “Behavior,” “Lhs,” “Action,” and “Event.” An Action is a process triggered by an Event that causes screen transition. An Lhs expresses the conditions under which each Action is executable. An Action may be a “Methodcall” or “Rhs”. An Rhs describes a UI behavior, such as a XAML model screen transition, when the conditions are fulfilled. A behavior aggregates these notions to express a particular interaction transition. A pair consisting of an Lhs and an Rhs describes a transition rule.

3.3 –Event-based model execution mechanism

The 2D/3D UI simulation system implemented by WPF collaborates with the XAML-B execution system using Visual C#. An example of the event-based model is shown in Fig. 3. When an event from the user is sent to the execution system from the WPF, the system compares the attribute values of the Event tag with those of each Lhs tag described in a set of Behavior tag. These tags are connected to XAML-B model as show in Fig 2. When those values match, the attribute values in XAML are rewritten to those specified in the Rhs tag of the Behavior tag.

We implemented a XAML-B execution system using Visual C#. A 2D/3D UI prototype of a digital camera (Fuji-Finepix Z1) on the market was modeled using this system, as shown in Fig.4. In this example, a 2D geometric model of the camera housing expressed by XAML was built using Illustrator, while a 3D model was generated using SolidWorks. The static structures of UI screen images, including menu lists and icons, was built in XAML using Illustrator. Changes in the UI screen caused by user interaction were described by the proposed XAML-B using a text editor. As shown in Fig.4, we confirmed that the 2D UI screen images from the 2D/3D UI prototypes were identical and that both simulations worked well.

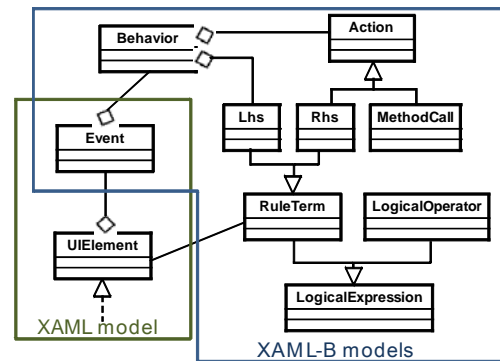


Fig. 2 : XAML and XAML-B models

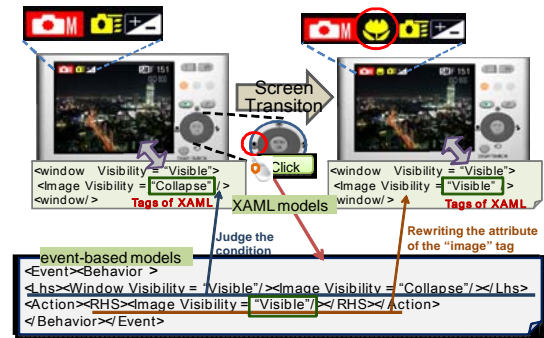


Fig. 3 : Execution of Event-based model

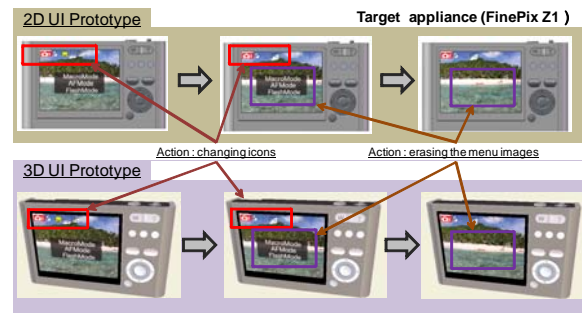


Fig. 4 :a example of 2D/3D UI prototype

4- Conclusions

We proposed XAML-based UI software specifications for the information appliances and developed a 2D/3D UI prototyping system based on these specifications. We were able to successfully execute uniform 2D/3D UI simulation using this system. Our future research tasks will include developing the functions of interactive XAML-B modelling and usability assessments.

References

[A1] Ali,T.F., et al.:”Building Multi-Platform User Interfaces with UIML” In: Multiple User Interfaces: Engineering and Application Framework. John Wiley and Sons, New York ,2003
 [H1]B.Hartmann et al.: ””Reflective physical prototyping through integrated design, test, and analysis”, UIST2006, pp.299-308, 2006
 [R1]RapidPlus, <http://www.skymobilemedia.com/main.php?func=devtools>
 [L1]Limbourg, Q., et al.: UsiXML: a Language Supporting Multi-Path Development of User Interfaces”, EHCI-DSVIS’2004, Lecture Notes in Computer Science, Vol. 3425, pp. 200-220 , 2005
 [X1]XAML,<http://msdn2.microsoft.com/en-us/library/ms747122.aspx>