

Model-based Generation of Interactive Digital TV Applications

Fabio Paternò, Sandro Sansone
ISTI-CNR
Via G.Moruzzi 1
56124 Pisa
+390503153066

{fabio.paterno, sandro.sansone}@isti.cnr.it

ABSTRACT

In this paper, we present a solution for the model-based generation of interactive software applications for the digital TV platform, which is soon likely to become one of the most widespread interactive platforms. We show how this solution can be integrated in an authoring environment for model-based design of interactive multi-platform applications and used at run-time for supporting migratory interfaces, which allow users to freely move about, change device and continue task performance from the point they left off in the previous device.

1. INTRODUCTION

Model-based approaches to interactive applications have often been criticised because they require time and effort to develop the models and then the control on the resulting user interface is sometimes limited. However, the increasing availability of various types of interactive devices (desktop, PDAs, cell phones, wide screens, ...) has brought renewed interest in such model-based approaches: the idea is to have device-independent languages able to highlight the main design concepts and then some transformations that generate the implementation adapted to the current target platform. This type of approach has been used to support access through various platforms: desktop, mobile, combined used of graphical and vocal interface, but has not been applied to an emerging platform such as the digital TV. In this paper, we present a solution that is able to generate user interfaces for the digital TV platform. To this end, we had to capture the specific aspects of interactions with such platform, consider the design criteria to obtain usable results and incorporate them into a transformation able to transform logical user interface descriptions into digital TV application implementations.

In the paper, we first provide some background information regarding the model-based approach framework that we apply. Next, we discuss the specific aspects of digital TV and user interface design criteria for this platform. Then, we describe our transformation from logical description to implementation. Lastly, we illustrate how it is exploited at both design and run-time.

2. BACKGROUND

In the research community in model-based design of user interfaces there is a general consensus regarding the useful logical descriptions [2][5]:

- The task and object level, which reflects the user view of the interactive system in terms of logical activities and objects that should be manipulated to accomplish them:

- The abstract user interface, which provides a modality-independent description of the user interface;
- The concrete user interface, which provides a modality-dependent but implementation language-independent description of the user interface;
- The final implementation, in an implementation language for user interfaces.

The advantage of this type of approach is that it allows designers to focus on logical aspects and take into account the user view right from the earliest stages of the design process. In the case of interfaces that can be accessed through different types of devices the approach has additional advantages. First of all, the task and the abstract level can be described through the same languages for whatever platform we aim to address. Then, in our approach we have a concrete interface language for each target platform. By platform we mean a set of interaction resources that share similar capabilities (for example the graphical desktop, the vocal one, the cellphone, the graphical and vocal desktop). Thus, a given platform identifies the type of interaction environment available for the user, and this clearly depends on the modalities supported by the platform itself.

At the abstract level we introduce a number of basic elements able to support different activities in a platform-independent manner, and we also describe how to compose such basic elements through some composition operators. In particular, the composition operators have been defined taking into account the type of communication effects that designers aim to achieve when they create a presentation (grouping, relation, ordering, hierarchy). The concrete level is a refinement of the abstract interface: depending on the type of platform considered there are different ways to render a specific object of the user interface and a specific operator of the abstract user interface. For example, in a graphical desktop system a navigator can be implemented either through a textlink, or an imagelink or a simple button. The same holds for the composition operators: indeed, the desktop environment allows using tables, so the grouping operator can be implemented by a number of techniques including both unordered lists by row and unordered list by column (apart from classical grouping techniques such as fieldsets, bullets, and colours). However, in a mobile phone platform the limited capabilities do not allow implementing the grouping operator by using an unordered list of elements by row, thus this technique is not available on this platform. In a vocal device, on the other hand, grouping can be achieved by inserting specific sounds or pauses or using a specific volume or keywords.

3. USER INTERFACES for DIGITAL TV

As already stated, herein we focus on how to support devices, such as the digital TV, which are not traditionally connected with the office environment. In the case of the digital TV, the objective is to understand the type of issues that inclusion of such a platform would raise in comparison with traditional desktop systems. Indeed, although this platform is similar in many ways to a graphical desktop, we must nevertheless take into account that users generally have no mouse or keyboard for interacting, but just a TV remote control. Therefore, as for comparing the concrete user interface for digital TV with that for the graphical desktop, we noticed some interesting differences. One of these was deciding to use a specific kind of font –Tiresias– due to its high suitability for display on TV screens. In addition, in order to guarantee the best readability of the text, quite high font dimensions were selected (range between 20-36 points), avoiding the smaller ones, which do not guarantee sufficient readability.

Usability is a fundamental consideration for interactive services delivered through technologies, such as the digital TV, which can be accessed by people with any background, often with limited computer skills. Since this is a new area there is still a lack of standards, but some guidelines have started to emerge [3] based on a number of user tests. If we analyse the main characteristics of digital TV interfaces, we can find that they involve many aspects, as discussed in the following.

In digital TVs the availability of an application should be clearly indicated and the access should be simple, fast and clear. In particular, it is important to:

- indicate the availability of the application;
- set a key to access the interactive application;
- set a key to exit;
- set a key to support zooming;
- indicate when the application is closing;
- indicate the loading state.

The application structure should be clear and intuitive. In particular, it is important to:

- separate content and commands;
- locate the most important elements in the top-left area;
- locate the audio/video flow in the top-right area;
- dedicate a portion of the screen to the commands;

It is important to consider that there is no mouse to support navigation but TV controls. Thus, designers should:

- structure the content hierarchically;
- assign a consistent and unique function to the keys;
- keep the order and representation of the control keys consistent on the screen;
- allow users to access menus through directional, coloured or numerical keys associated with the functionalities.

It is important that the user receive feedback of the activated process before it terminates. Thus, the interactive applications should:

- promptly communicate through messages what is happening;

- define graphical or textual help functions;

There is no standard so far, even if there are several constraints in the TV platform. Thus, it is important to:

- use appropriate fonts, such as Tiresias, and appropriate text sizes from 20 (for notes) up to 36 (for titles) points;
- reduce flickering by avoiding drawing lines with a size of 1 or 2 pixel;
- avoid strongly contrasting colours in contiguous areas in order to avoid flickering;
- choose appropriate colours taking into account that the DVB-MHP standard supports a limited number of colours (palette with 188 colours).

Some applications require the user to enter some data. The lack of a keyboard requires using the TV control for this purpose. Thus, it is important to:

- limit the number of fields to fill in;
- define mechanisms to enter textual data;
- use the TV control to enter numerical data;
- inform the user when the return channel is used;
- indicate the result of the transactions through appropriate feedback.

The nature of the TV is different from many other communication media providing textual information. It is thus important to pay attention in order to avoid making the reading process tedious. To this end the interactive applications should:

- present the text in such a way that the important information is at the beginning;
- use simple verbal forms;
- use titles, subtitles and abstracts to structure the text;
- divide text on multiple pages or insert a scrolling bar;
- use the colours with semantic purposes.

4. LOGICAL DESCRIPTION for DIGITAL TV USER INTERFACES

In order to support transformations from a platform-independent language to an implementation (in our case for the digital TV), we use an intermediate language, the concrete description, which is a refinement of the abstract interface but platform-dependent.

Table 1 provides a description of how each abstract interface element is implemented in the digital TV platform. Three types of elements are considered: output-only, interaction and the composition elements (indicating how to put the other elements together). Thus, for example, a navigator element (which allows users to move to another point of the application) can be supported through textual or graphical links or buttons.

In our case the generation of the user interface implementation involves the generation of a file in a Java version for digital TVs representing a Xlet, which is an application that is immediately compiled and can be interpreted and executed by the interactive TV decoders. Xlets bear a strong resemblance with common java applets, with the difference that, instead of the Web browser (which executes the applets) there is the MHP layer of the digital receiver (Set-Top-Box) which interprets them.

Only output	Digital TV Implementation
Textual	A specific font for the TV is used (Tiresias). The text size is usually larger than in computer output. It usually varies from 20/22 for the notes to 36 for the titles.
Object	Images in their real dimensions or scaled if they are too large. It is possible their zooming or scrolling.
Description	The annotation of the images can be presented as text under the image
Interaction	Digital TV Implementation
Navigator	Operations are associated with the selection of textual links, graphical links, and buttons
Activator	The TV Reset Button resets the fields of a form as well. Button and functions: associate the button selection with a TV function execution
Text Edit	Text fields are supported. The text editing can be performed by selecting the field to edit and entering the data through a virtual keyboard.
Numerical Edit	It is as the text edit but there is no need for the virtual keyboard because it is sufficient to use the numbers of the TV control.
Single Selection	It is implemented with radio button (with up to 6 choices) and a list box (with more than 6 choices)
Multiple Selection	It is implemented with check box (with up to 6 choices) and a list box (with more than 6 choices)
Composition	Digital TV Implementation
Grouping	Groups elements lining them vertically or horizontally, or with the same colour or with bullets
Ordering	Order elements through numbered list
Hierarchy	Most important elements are located in the top area with larger graphical attributes
Relation	It is implemented through a container of interactors and their composition operators, in which some activators trigger functionalities involving all the elements.

Table 1 Logical elements implementation in Digital TV.

In real settings, the file generated with our tool (the Xlet) can be downloaded on the Set-Top-Box (via broadcasting or http). If this is not possible, an emulator can be used to execute the interactive Xlet. For our examples, we used the XletView emulator (<http://xletview.sourceforge.net>), we will show an example of its use in the next sections.

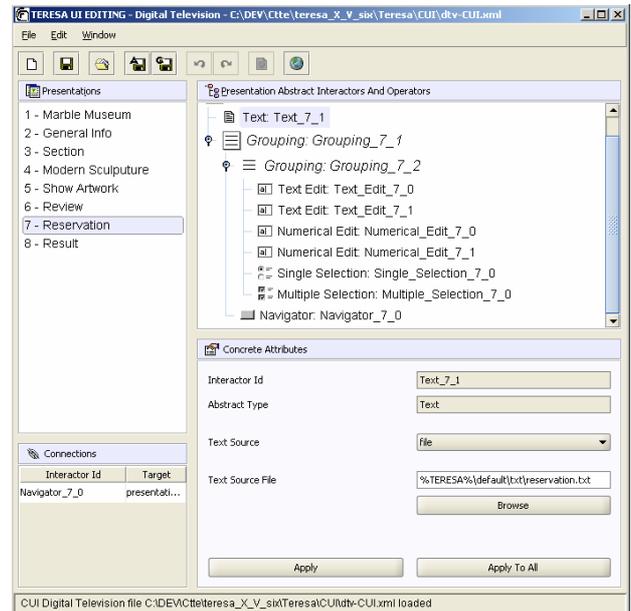


Figure 1. The authoring environment for Digital-TV.

Our logical descriptions and transformations have been included in an authoring environment. Figure 1 shows such an environment in the example of a museum application corresponding to the interface in Figure 2. The interface is divided into four main panels: one (top-left) listing the presentations of the application; the abstract interactors and composition operators of the currently selected presentation are shown in the top-right part; the concrete attributes of the currently selected abstract element are indicated in the bottom-right part (with the possibility of editing them); and lastly the connections supporting the navigation are listed in the bottom-left part.

The next figure shows a part of a museum application in which it is possible to fill in the information requested from the user and proceed with making reservation, and such editing activity is performed by using a virtual keyboard.

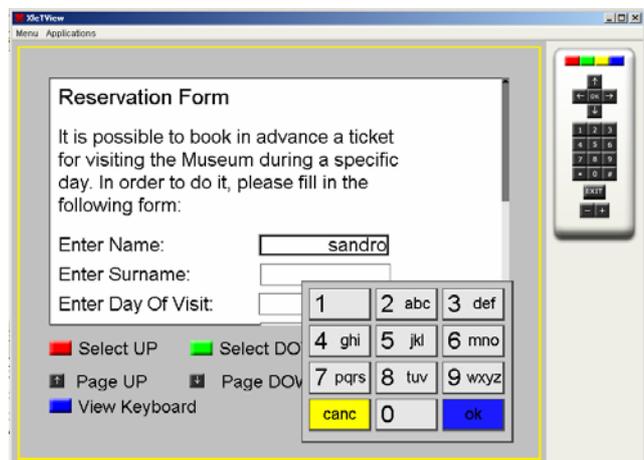


Figure 2. Digital TV interface for Booking a Ticket.

5. MIGRATION

The transformations that we have discussed aiming at obtaining implementations adapted to the features of the target platform are also useful at run-time. An interesting application is in migratory interfaces: interface able to follow the user across different devices allowing task continuity and user interface adaptation. This type of interface is very useful for forthcoming ubiquitous environments in which users want to freely move and change interaction device with the possibility of continuing the task from the point they left off in the source device. In order to support migration involving Digital TV, we have extended a previously developed infrastructure for migratory interfaces [1]. In practise, the environment supports a number of reverse and forward transformations that allow taking an existing PC desktop implementation of an application, building the corresponding concrete and abstract description and using it as a starting point for creating the concrete description and the implementation adapted for the device accessing it. In addition to interface adaptation, the environment also supports task continuity. To this end, when a request of migration to another device is triggered the infrastructure detects the state of the application modified by the user input (elements selected, data entered, ...) and identifies the last element accessed in the source device. Then, a version of the interface for the target device is generated, the state detected in the source device version is associated with the target device version, which is then activated at the point supporting the last basic task performed in the source version.

In order to perform such transformations the logical descriptions play a key role. For example, when deciding how to associate the state of the source version to the target version, our migration infrastructure looks for the abstract elements corresponding to the implementation elements modified in the source version, then identifies what elements in the target version are used for their implementation and then associate the entered values to such elements. Likewise, the point in which the target interface is activated is identified: the environment identifies the last input entered and the corresponding abstract element and then looks for its implementation in the target version. The presentation including it will be activated in the target device because it is assumed that the users want to continue from the point in which they left off in the source device.



Figure 3. The interface for the mobile device

The example application considered in this case is a shopping application. This allows users returning home from work to start to prepare the shopping list through a mobile device (see Figure 3) while they are on the bus or train. When they arrive home they may realise that something is still missing and thus decide to migrate to the digital TV so that they can continue the editing of the shopping list through a larger screen (see Figure 4) and lastly send the request.

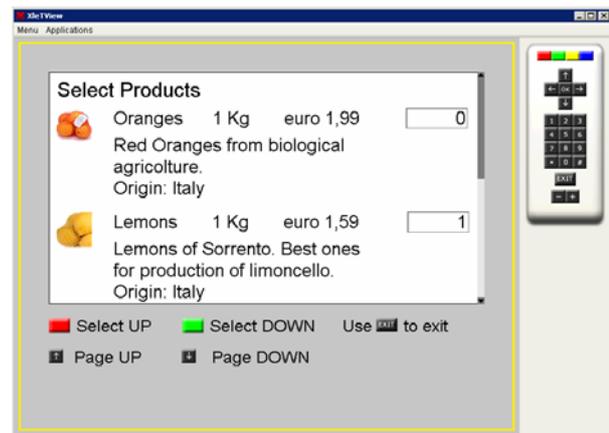


Figure 4. The Digital-TV application interface after migration.

6. CONCLUSIONS

We have presented a solution for the model-based design and generation of interactive applications for digital TVs. It is based on the use of specific models able to highlight the important aspects related to the user interface and provide a description at various abstraction levels. Such models and the associated transformations have been integrated in an authoring environment and a migration infrastructure. We have also shown example applications. The authoring tool is available for external download and use (<http://giove.isti.cnr.it/teresa.html>).

7. REFERENCES

- [1] Bandelloni, R., Mori, G., Paternò, F., Dynamic Generation of Migratory Interfaces, Proceedings Mobile HCI 2005, ACM Press, pp.83-90, Salzburg, September 2005.
- [2] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers. Vol. 15, No. 3, June 2003, pp. 289-308.
- [3] Fondazione Ugo Bordoni – “Raccomandazioni per le interfacce dei servizi interattivi della televisione digitale”, <http://www.fub.it/ambientedigitale/repository/Generale/Raccomandazioni.pdf>
- [4] Mori, G. Paternò, F. Santoro C., Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions, IEEE Transactions on Software Engineering, August 2004, Vol.30, N.8, pp.507-520, IEEE Press.
- [5] Paternò, F., “Model-Based Design and Evaluation of Interactive Application”. Springer Verlag, ISBN 1-85233-155-0, 1999.