An Environment for Designing and Developing Multi-Platform Interactive Applications

Silvia Berti, Giulio Mori, Fabio Paternò, Carmen Santoro

ISTI-CNR, Via G.Moruzzi, 1
56124 Pisa, Italy
{s.berti, g.mori, f.paterno, c.santoro}@isti.cnr.it

Abstract. The increasing availability of new types of interaction platforms raises the need for new methods and tools to support development of nomadic applications. This paper presents a solution based on the use of multiple levels of abstractions and a number of transformations able to obtain concrete interfaces, taking into account the different platforms and preserving usability.

1 Introduction

With the advent of the wireless Internet and the rapidly expanding market of smart devices, designing interactive applications supporting multiple platforms has become a difficult issue. In fact, on the one hand the decreasing cost at which the devices are now offered has enabled an increasing variety of people to become potential users of features and services of novel generations of communication technology as never before. Moreover, the increasing use of such devices in mobile settings has also stimulated a growing interest in exploring the unique opportunities offered by multi-modal applications (graphic and/or vocal), to the aim of providing users with as much flexibility as possible and produce more natural and faster operations. On the other hand, rarely such a high number of flourishing range of opportunities offered have become effective, due to the low quality of the user interfaces provided to the users.

The main problem is that many assumptions that have been held up to now about classical stationary desktop systems are being challenged when moving towards nomadic applications, which are applications that can be accessed through multiple devices from different locations. Consequently, one fundamental issue is how to support software designers and developers in building such applications: in particular, there is a need for novel methods and tools able to support development of interactive software systems able to adapt to different targets while preserving usability.

Model-based approaches [1] could represent a feasible solution for addressing such issues: the basic idea is to identify useful abstractions highlighting the main aspects that should be considered when designing effective interactive applications. In particular, the capability to use task models to describe the logical activities that should be supported by an interactive system in this heterogeneous context is an important starting point to overcome the limitations of the current approaches.

In the paper we first describe the basic concepts of our method. Then, we move on describing the TERESA tool that has been designed and implemented to support this approach. Lastly, some examples of application of the method to a virtual museum have been provided.

2 The Method

Our method [2] for model-based design is composed of a number of steps (see Figure 1) that allows designers to start with an envisioned overall task model of a nomadic application and then derive concrete and effective user interfaces for multiple devices.

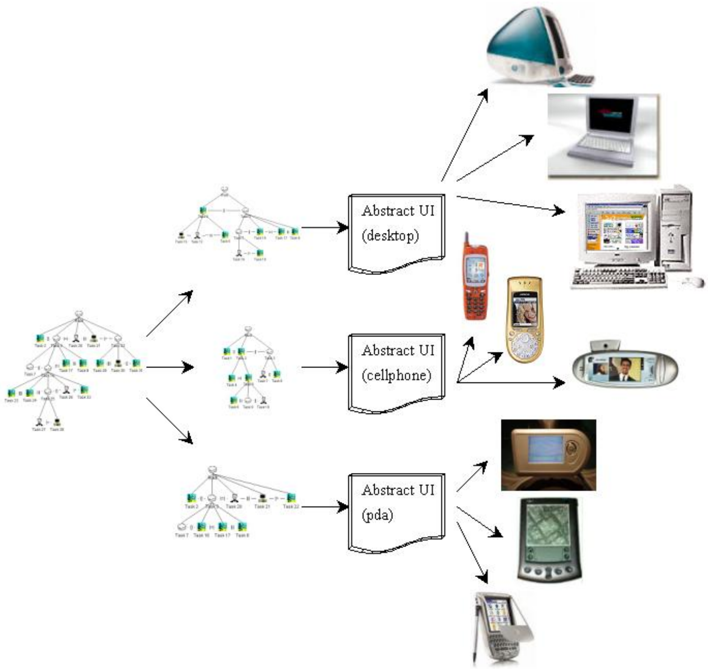

Fig. 1. The transformations supported by our method

The main steps are:

- *High level task modelling*: the output of this phase consists of the description of the logical activities that need to be performed in order to reach the users' goals. This description initially considers an integrated task model where all the activities that have to be supported have been specified. Next, the task model is refined and structured so as to identify the activities that have been supported for each platform considered.
- *Abstract user interface (AUI)*: in this phase the focus shifts to the interaction objects supporting task performance. An abstract user interface is defined in

terms of presentations, identifying the set of user interface elements perceivable at the same time, and each presentation is composed of a number of interactors [3], which are abstract interaction objects identified in terms of their main semantics effects. An XML-based language has been specified in order to describe the organisation of the various interactors within the presentations. The structure of the presentation is defined in terms of elementary interactors characterised in terms of the task they support, and their composition operators. Such operators are classified according to the communication goals to achieve: a) *Grouping*: indicates a set of interface elements logically connected to each other; b) *Relation*: highlights a one-to-many relation among some elements, one element has some effects on a set of elements; c) *Ordering*: some kind of ordering among a set of elements can be highlighted; d) *Hierarchy*: different levels of importance can be defined among a set of elements.

- *User interface generation*. This phase is completely platform-dependent and has to consider the specific properties of the target device. The interactors are mapped into interaction techniques supported by the particular device configuration considered (operating system, toolkit, etc.), and the also the operators defined in the language for abstract user interface are implemented with appropriate presentation techniques.

3 The TERESA Tool

TERESA [4] is a transformation-based environment designed and developed by our group at ISTI-CNR. It is intended to provide a complete semi-automatic environment supporting a number of transformations useful for designers to build and analyse their design at different abstraction levels and consequently generate the concrete user interface for a specific type of platform.

A number of main requirements have driven the design and development of TERESA:

- *Mixed initiative*; we want a tool able to support different level of automations ranging from completely automatic solutions to highly interactive solutions where designers can tailor or even radically change the solutions proposed by the tool.
- *Model-based*, the variety of platforms increasingly available can be better handled through some abstractions that allow designers to have a logical view of the activities to support.
- *XML-based*, XML-based languages have been proposed for every type of domain. In the field of interactive systems there have been a few proposals that partially capture the key aspects to be addressed.
- *Top-down*, this approach is an example of forward engineering. Various abstraction levels are considered, and we support cases when designers have to start from scratch. So, they first have to create more logical descriptions, and then move on to more concrete representations until the final system.

- *Different entry-points*, our approach aims to be comprehensive and to support the entire task/platform taxonomy. However, there can be cases where only a part of it needs to be supported.
- *Web-oriented*, the Web is everywhere, and so we decided that in order to be general Web applications should be our first target. However, the approach can be easily extended to other environments (such as Java applications, Microsoft environments, …) because only the last transformation needs to be modified for this purpose.

The TERESA tool offers a number of transformations between different levels of abstractions and provide designers with an easy-to-use integrated environment for generating both XHTML and VoiceXML user interfaces [5]. With the TERESA tool, at each abstraction level the designer is in the position of modifying the representations while the tool keeps maintaining forward and backward the relationships with the other levels thanks to a number of automatic features that have been implemented (e.g. the possibility of links between abstract interaction objects and the corresponding tasks in the task model so that designers can immediately identify their relations). This results in a great advantage for designers in maintaining a unique overall picture of the system, with an increased consistence among the user interfaces generated for the different devices and consequent improved usability for end-users.

4 Generating Multiplatform User Interfaces

In this section we show some examples of user interfaces derived by applying the described approach to a museum application.

<table>
<tr><th>Desktop System</th><th>Cellphone</th><th>VoiceXML-Enabled Phone</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>System:
“The ‘*Boat’* has been achieved through the subtle divisions of the planes enveloping its central part, which is only rough-hewn; the material is white marble.” (*Five second pause*)
“Remember that if you would like to return to the main menu, say ‘*home’* or if you would like to go back to the previous menu, say ‘*back’*.”</td>
</tr>
</table>

Table 1. Presentations of artwork information on different platforms.

More specifically, in Table 1 the presentations refer to a situation in which the user has selected information about a specific artwork of Modern Sculpture section. As you can note, there are some differences concerning the presentation of the artwork selected: on the desktop system the picture of the artwork is shown, together with additional information (title, description, artist, material, ..).; on the cellphone the picture is provided as well but the textual information is more concise; in a VoiceXML-enabled system a vocal description of the artwork is provided. It is worth pointing out that on the different platforms also the navigation mechanisms change: on both desktop system and cellphone some links have been (visually) presented, although they differ in number. In fact, on the desktop system there is enough room for displaying also links to other sections of the museum, whereas on the cellphone the choice is more limited. In the third case (VoiceXML) the navigation is implemented by dialogs that have been vocally provided.

Desktop System	**Cellphone**	**VoiceXML-Enabled System**
		System: (*grouping sound*) The artworks contained in the section are the following: *'Boat'*, *'Totem'*, *'Hole'*. If you would like information on one of these please say its name (*grouping sound*) Remember that if you would like to return to the main menu, say home.

Table 2. Implementation of the grouping operator on different platforms

Another interesting point is represented by the different implementations of the operators of the abstract user interface on the various platforms. In Table 2 the different implementations of the grouping composition operator on the various platforms have been shown: in the first two presentations it is obtained through a list of graphical buttons (1st case) or pull down menu (2nd case) whereas in the vocal interface (3rd case) there are sounds that delimit the grouped elements.

5 Conclusions

In this paper a model-based approach for designing and developing multiplatform applications has been presented, and the main features of the related TERESA tool have been outlined. Some examples of presentations generated have been provided as well, highlighting the capability of the approach to provide usable multimedia interfaces for a broader set of mobile devices, including vocal interaction techniques.

The tool is publicly available at http://giove.cnuce.cnr.it/teresa.html. This work has been supported by the IST V Framework CAMELEON (Context Aware Modelling for Enabling and Leveraging Effective interaction) project. More information is available at http://giove.cnuce.cnr.it/cameleon.html.

References

1. Puerta, A., Eisenstein, J., *Towards a General Computational Framework for Model-based Interface Development Systems*, *Proceedings ACM IUI'99*, pp.171-178.
2. F.Paternò, C.Santoro, "A Unified Method for Designing Interactive Systems Adaptable to Mobile and Stationary Platforms", Interacting with Computers, Vol.15, N.3, pp 347-364, Elsevier, 2003.
3. Paternò, F., Leonardi, A., 1994. A Semantics-based Approach to the Design and Implementation of Interaction Objects, Computer Graphics Forum, Blackwell Publisher, Vol.13, N.3, 195-204.
4. G. Mori, F. Paternò, C. Santoro, "Tool Support for Designing Nomadic Applications", Proceedings ACM IUI'03, Miami, pp.141-148, ACM Press.
5. S. Berti, F. Paternò, Model-based Design of Speech Interfaces, Proceedings DSV-IS 2003, Springer Verlag .