A Transformation-based Environment for Designing Multi-Device Interactive Applications

Silvia Berti, Giulio Mori, Fabio Paternò, Carmen Santoro
I.S.T.I.-C.N.R.
Via G.Moruzzi 1
56124, Pisa - Italy
+39 050 3153066
{s.berti, g.mori, f.paterno, c.santoro}@isti.cnr.it

ABSTRACT
The ever-increasing availability of new types of devices raises a number of issues for user interface designers and interactive software developers. We have designed and developed a tool (TERESA), which can be helpful when designing applications accessible through various device types.

Categories & Subject Descriptors: H.5INFORMATION INTERFACES AND PRESENTATION – I2.2 Automatic Programming: Program Transformation.

General Terms: Design, Experimentation, Human Factors.

Keywords: Model-based Design, User Interface Transformations, Heterogeneous Devices

INTRODUCTION
Designing applications that exploit new multi-platform technology is often a difficult problem. For software developers this introduces the problem of constructing multiple versions of single applications and endowing these versions with the ability to dynamically respond to changes in context. In current practise the design of multi-platform applications is often obtained through the development of several versions of the same applications, one for each platform considered. Then, such versions can at most exchange data. This solution with no tool support to address multi-platform issues is rather limited because it implies high implementation and maintenance costs. The opposite solution, completely automatic, is to use transcoding where an application written in a language for a platform is automatically transformed into an application in a language for another platform. However, this type of solution often provides bad results in terms of usability because it assumes that the same tasks are supported by each platform and tends to support them in the same manner without taking into account the specific features of the platform at hand.

Our approach extends previous work in the model-based design area [2] in order to support development of nomadic applications. In particular, we have designed and developed the TERESA (Transformation Environment for inteRactivE Systems representAtions) tool providing general solutions that can be tailored to specific cases. This tool supports transformations in a top-down manner, providing the possibility of obtaining more concrete descriptions starting from abstract representations. It differs from other approaches such as UIML [1], which mainly consider low-level models.

THE METHOD
Our method for model-based design is composed of a number of steps that allows designers to start with an overall envisioned task model of a nomadic application and then derive user interfaces for multiple devices:

- *High-level task modelling of a multi-context application*. In this phase designers develop a single model that addresses the possible contexts of use and the roles involved and also all the objects that have to be manipulated to perform tasks and the relations among them. Such models are specified using ConcurTaskTrees [3], which allows designers to indicate the platforms suitable for each task.
- *Developing the system task model for the different platforms considered*. Here designers have to filter the task model according to the target platform and, if necessary, further refine the task model, depending on the specific device considered, thus, obtaining the system task model for the platform considered.
- *From system task model to abstract user interface*. Here the goal is to obtain an abstract description of the user interface composed of a set of abstract presentations that are identified through an analysis of the task relationships and structured by means of interactors composed of various operators.
- *User interface generation*. In this phase we have the generation of the user interface. This phase is completely platform-dependent and has to consider the specific properties of the target device. In order to support generation in new user interface languages only this transformation has to be modified.

IUI'04, Jan. 13–16, 2004, Madeira, Funchal, Portugal.
ACM 1-58113-815-6/04/0001.

THE TOOL

TERESA 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. So, designers 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*, we decided that Web applications should be our first target. However, the approach can be easily extended to other environments.

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. 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.

Once the elements of the abstract user interface have been identified, every interactor has to be mapped into interaction techniques supported by the particular device configuration considered (operating system, toolkit, etc.), and also the abstract operators have to be appropriately implemented by highlighting their logical meaning: a typical example is the set of techniques for conveying grouping relationships in visual interfaces by using presentation patterns like proximity, similarity and continuity.

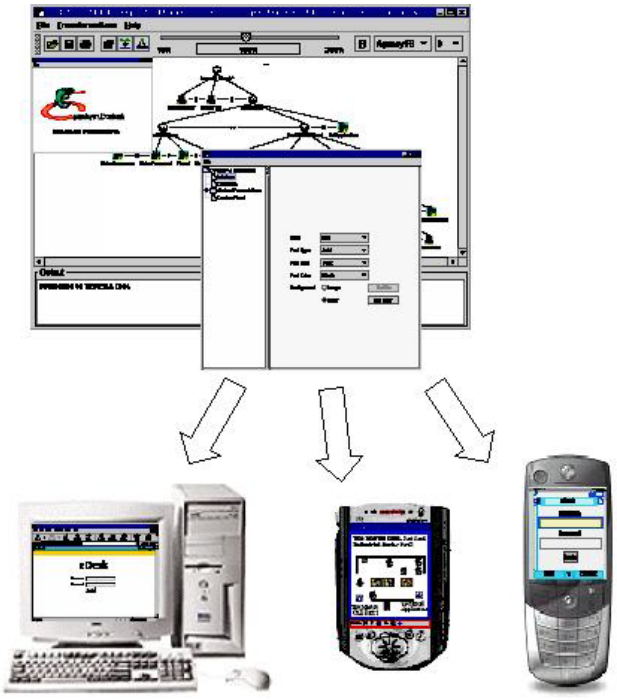

Figure 1: The TERESA Approach.

CONCLUSIONS

The environment presented supports many features to support design and development of user interfaces for multi-platform applications. This is obtained through a number of transformations that can be performed either automatically or through interactions with the designer. The tool helps designers clarify design issues and support analysis and evaluation of design options. It can be freely downloaded at http://giove.cnuce.cnr.it/teresa.html.

This work has been supported by the EU IST CAMELEON project (more information available at http://giove.cnuce.cnr.it/cameleon.html)

REFERENCES

1. Abrams, M., Phanouriou, C., Batongbacal, A., Williams, S., Shuster, J. *UIML: An Appliance-Independent XML User Interface Language*, Proceedings of the 8th WWW conference, 1999.
2. Einsenstein, J., Vanderdonckt, J., Puerta, A. *Applying Model-Based Techniques to the Development of UIs for Mobile Computers*, Proceedings IUI'01: International Conference on Intelligent User Interfaces, ACM Press, 2001.
3. Paternò, F., Model-Based Design and Evaluation of Interactive Application. Springer Verlag, ISBN 1-85233-155-0, 1999.