

Natural Development of Ubiquitous Interfaces

Silvia Berti, Fabio Paternò, Carmen Santoro

ISTI-CNR

Via G.Moruzzi, 1 – 56100 Pisa - Italy
{s.berti, f.paterno, c.santoro}@isti.cnr.it

Natural development aims to ease the development process of interactive software systems. Some work has been dedicated to obtain natural programming (Pane and Myers, 1996), which aims to support programming through languages that are understandable by people who have not programming experience. However, there is a need for a more comprehensive view of the development cycle, thus not limited only to programming, but also including requirements, designing, modifying, tailoring, Natural development implies that people should be able to work through familiar and immediately understandable representations that allow them to easily express and manipulate relevant concepts, and thereby create or modify interactive software artefacts. On the other hand, since a software artefact needs to be precisely specified in order to be implemented, there will still be the need for environments supporting transformations from intuitive and familiar representations into more precise, but more difficult to develop, descriptions.

The main motivation for model-based approaches to user interface design has been to support development through the use of meaningful abstractions in order to avoid dealing with low-level details. Despite this potential benefit, their adoption has mainly been limited to professional designers. New solutions are emerging that are able to extend such approaches in order to achieve natural development through environments that enable end-users to develop or modify interactive applications still using conceptual models, but with continuous support that facilitates their development, analysis, and use. TERESA (Mori, Paternò and Santoro, 2003) is an authoring transformation-based environment for developing ubiquitous interfaces. 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 various types of platform (Figure 1). This tool has been extended for smoothly moving from a model-based approach to natural development. For this purpose we have identified three key criteria: integrated support of both familiar/informal and engineered/structured specifications; effective representations supporting analysis and highlighting the information of interest; and possibility of flexible support for development of software artefacts from either scratch or an existing system.

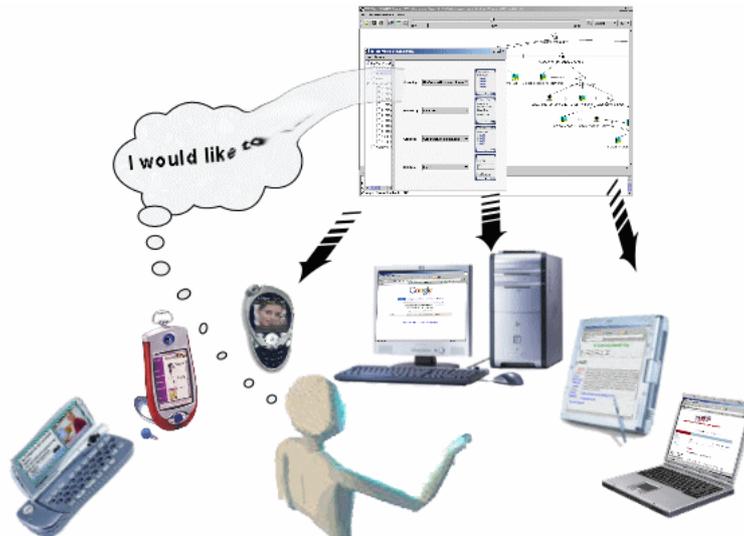


Figure 1: The approach underlying TERESA

Integrating informal and structured representations: End-user development can benefit by the combined use of multiple representations that can have various levels of formality. At the beginning of the design process many things are obscure and unclear. It is hard to develop precise specifications from scratch. In addition, there is the problem of clearly understanding what user requirements are. Thus, it can be useful to use the results of initial discussions to feed the more structured parts of the design process. In general, the main issue of end-user development is how to use personal intuition, familiar metaphors and concepts to obtain or modify a software artefact, whose features need to be precisely defined in order to obtain consistent support for the desired functionality and behaviour. In this process all the available multimedia possibilities should be addressed. Examples of informal input for more structured representations are textual scenarios (Tam and others, 1998) and sketches on board (Landay and Myers, 2001).

Providing Effective Representations: visual modelling techniques have been more and more adopted in the software design process. However, we are still far from visual representations that are easy to develop, analyse and modify, especially when large case studies are considered. As soon as the visual model increases in complexity, designers have to interact with many graphical symbols connected in various ways and have difficulties analysing the specification and understanding the relations among the various parts, thus the need for effective representations useful to easily analyse models, for example combining focus and context views on the model.

Supporting different entry points/abstraction levels: The starting point of development activity can often vary. In some cases people start from scratch and have to develop something completely new; in other cases people start with an existing system (often developed by somebody else) and need to understand the underlying conceptual design in order to modify it or to extend it to new contexts of use. Thus, a general development environment should be able to support a mix of forward (from conceptual to concrete) and reverse (from concrete to conceptual) engineering processes. This calls for environments that can support various transformations able to move among various levels (code, specification, conceptual description) in both a top-down and bottom-up manner and to adapt to the foreseen interaction platforms without duplication of the development process.

Examples of such criteria have been taken into account in the new version of TERESA, a transformation-based environment supporting the design and development of ubiquitous interfaces that can be accessed through heterogeneous platforms (desktop, PDA, mobile phones, ...) and from different locations and are able to dynamically adapt to the new context of use. An example of user interfaces obtained through the tool is in Figure 2.

This authoring environment allows even people without background in programming to specify the activities that would like to be supported indicating the potential contexts of use. Then the environment supports the transformation of such logical descriptions in more concrete ones providing suggestions on how the resulting interface should be organised in order to provide usability criteria specific to the devices and modalities considered. End-user developers can still modify the suggestions in case they prefer to tailor the interface to specific needs. The mixed-initiative interaction paradigm provided to TERESA's users, together with the adaptive features and multiple representations at different levels of abstraction, make TERESA an interesting example of how it is possible to embed end user development features in model-based interactive systems, so as to enable even non-professional users to develop and modify their own interactive applications.

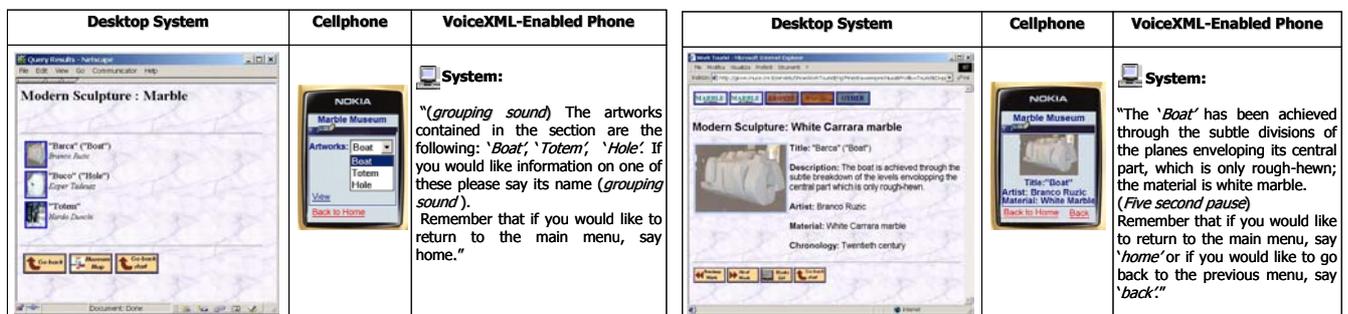


Figure 2: An example of multi-device interface obtained through TERESA.

References

- Landay J. and Myers B., "Sketching Interfaces: Toward More Human Interface Design." In IEEE Computer, 34(3), March 2001, pp. 56-64
- Mori, G., Paternò, F., Santoro, C., (2003) Tool Support for Designing Nomadic Applications, Proceedings Intelligent User Interfaces '03, pp.141-148, ACM Press, 2003.
- Pane J. and Myers B. (1996), "Usability Issues in the Design of Novice Programming Systems" TR# CMU-CS-96-132. Aug, 1996. <http://www.cs.cmu.edu/~pane/cmu-cs-96-132.html>
- Tam, R.C.-M., Maulsby, D., and Puerta, A., "U-TEL: A Tool for Eliciting User Task Models from Domain Experts", Proceedings IUI'98, ACM Press, 1998