

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 building interactive software systems. Some work has been dedicated to obtaining natural programming (Pane and Myers, 1996), which aims to support programming through languages understandable by people without any specific programming skills. On the one hand, natural development implies that people should be able to work through familiar and immediately understandable representations that allow them to easily express relevant concepts, and thereby create or modify applications. 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 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 to avoid dealing with low-level details. Despite such potential benefits, their adoption has mainly been limited to professional designers, but new solutions are recently emerging that are able to extend such approaches in order to achieve natural development by enabling 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, providing semi-automatic support for a number of transformations useful to build and analyse the user interface design at different abstraction levels and generate a suitable implementation for various platforms (desktop, PDA, mobile phones, vocal, ..). This tool has been extended to smooth the transition from a model-based approach to natural development according three criteria: integrating informal and structured specifications; providing effective representations; and supporting different entry points/abstraction levels.

Integrating informal and structured specifications: End-user development can benefit from using multiple representations with various levels of formality. In fact, at the beginning of the design process many things are obscure and unclear, so it is hard to develop precise specifications from scratch, especially because a clear understanding of the user requirements is a non-trivial activity. The main issue of end-user development is how to exploit personal intuition, familiar metaphors and concepts to obtain/modify a software artefact. Examples of informal input for more structured representations are *textual scenarios* (Tam and others, 1998) and *sketches on board* (Landay and Myers, 2001). For example, non-programmer users feel comfortable with sketch-based systems that allow them to concentrate on concepts by exploiting natural interactions, instead of being distracted by cumbersome low-level details required by rigid symbolisms. Such systems are generally able to recognise graphical elements and convert them into formats that can be edited and analysed by other software tools. With our approach, it is possible to interpret informal sketches, and translate them into appropriate descriptions that can be used in TERESA.

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 easy to develop, analyse and modify, especially when large case studies are considered. When the visual model increases in complexity, designers have to interact with many graphical interconnected symbols that make difficult understanding and analysing the specification: therefore, effective representations are needed. For example combining focus and context views of the model and applying semantic zooming techniques.

Supporting different entry points/abstraction levels: The starting point of the development activity can often vary. Sometimes people start from scratch to develop completely new artefacts; in other cases people need to understand the underlying conceptual design of an existing system, in order to adapt 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 cycles. The round trip engineering process supported by TERESA, a transformation-based environment for the design and development of ubiquitous interfaces that can be accessed through heterogeneous platforms (desktop, PDA, mobile phones, vocal and others, (see a small example in Figure 1) allows maintaining links among elements at different abstraction levels and helps users in understanding such links.

Desktop System	Cellphone	VoiceXML-Enabled Phone
		<p> System:</p> <p>“Welcome to the Marble Museum of Carrara Voice Response System. You can always use “out” to exit the application. If you want to listen to general information, say “Information”; if you want to listen to information about specific artworks, say “artworks”; if you want to book a ticket, say “ticket”.</p>

Figure 1: An example of multi-device interface obtained through TERESA.

The authoring environment allows even people without background in programming to specify the logical activities to support, also indicating the potential contexts of use. Moreover, the environment supports the transformation of such logical descriptions into more concrete ones, providing suggestions on how the resulting interface should be organised to implement usability criteria specific to the devices and modalities considered, although end-user developers can still modify such suggestions in order to meet specific needs. TERESA (publicly available at <http://giove.isti.cnr.it/teresa.html>) offers a mixed-initiative interaction paradigm together with adaptive features and multiple representations at different levels of abstraction to ease the development of software at all levels.

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 IUI'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