Tool-Supported User-Centred Prototyping of Mobile Applications

Karin Leichtenstern, Augsburg University, Germany

Elisabeth André, Augsburg University, Germany

Matthias Rehm, University of Aalborg, Denmark

ABSTRACT

There is evidence that user-centred development increases the user-friendliness of resulting products and thus the distinguishing features compared to products of competitors. However, the user-centred development requires comprehensive software and usability engineering skills to keep the process both cost-effective and time-effective. This paper covers that problem and provides insights in so-called user-centred prototyping (UCP) tools which support the production of prototypes as well as their evaluation with end-users. In particular, UCP tool called MoPeDT (Pervasive Interface Development Toolkit for Mobile Phones) is introduced. It provides assistance to interface developers of applications where mobile phones are used as interaction devices to a user's everyday pervasive environment. Based on found tool features for UCP tools, a feature study is described between related tools and MoPeDT as well as a comparative user study between this tool and a traditional approach. A further focus of the paper is the tool-supported execution of empiric evaluations.

Keywords: *Computer Science, Human-Centred Design, Human-Computer Interaction, Hybrid Simulation, Mobile Phone Application Development, Pervasive Environments, User-Centred Prototyping Tools, User Evaluation of Mobile Applications*

INTRODUCTION

Recent years have brought the tendency to develop mobile applications in human-centred iterations (ISO norm 13407) in order to increase the user-friendliness (Nielsen, 1994; Rogers, Sharp, & Preece, 2002) of the final product. Usually, this process includes a phase to (1) understand and specify the context of use, (2) specify the user and organisational requirements (3) produce design solutions, and (4) evaluate these design solutions against the requirements. With the continuous involvement of end-users in empiric user evaluations, these phases are iteratively executed until the intended application fulfils all requirements of the end-users (ISO norm 13407). All phases of this human-centred process can be very time-consuming and error-prone and thus they can increase the development costs and time. In particular, the

DOI: 10.4018/jhcr.2011070101

third phase requires experienced programming skills since different kinds of prototypes need to be efficiently and effectively designed and implemented. Also, the fourth phase involves expert knowledge in usability engineering due to the fact that user evaluations need to be properly conducted and analysed in order to efficiently and effectively verify whether the requirements are met.

These are causes why Myers (1995) argues for the development and application of tools or toolkits in order to save time and money. Myers points out two main requirements for tools. Firstly, tools need to improve the result of a development process: the quality of the resulting product. Secondly, tools should also enhance the process itself: the ease of use and efficiency to run through the process.

In this paper we present the approach of all-in-one tools which support both the third and fourth phase of the human-centred design process. We call these tools user-centred prototyping (UCP) tools since they support the design and implementation of prototypes (third phase) as well as the prototypes' evaluation and analysis (fourth phase) with end-users. As an example we introduce our UCP tool called MoPeDT (Pervasive Interface Development Toolkit for Mobile Phones) that supports the UCP of mobile applications where mobile phones are used as interaction device in a user's everyday environment. This concept follows Alan Kay's term *Third Paradigm Computing* (http://www.ubiq.com/hypertext/weiser/UbiHome.html) and the concept of *Pervasive* or *Ubiquitous Computing* (Weiser, 1991) where users can either (1) directly interact with real world objects of the user's everyday life or (2) mediated via an interaction device.

In the following, we illustrate typical steps of the process that we call user-centred prototyping – the third and fourth phase of the human-centred design process. These steps base on our practical experience (Rukzio, Leichtenstern, Callaghan, Schmidt, Holleis, & Chin, 2006). Then we describe existing tool-support for the UCP and typical tool features. The main part of the paper addresses MoPeDT and its validation via a feature and user study. Finally, we cover ideas to tool-supported execute empiric evaluations.

Design Specification of Prototypes

When producing a design solution, the application's appearance and behaviour is typically first designed and later on implemented. For mobile applications, the design is often done by defining a model for the different screens and the application flow which typically resembles a state chart. In this model, each state represents a screen of the mobile application and from each of these screen states user interactions can call other screen states. Consequently, user interactions (e.g. the execution of a keyboard command) are represented by transitions in the model. With respect to the level of detail, the design specification can be modelled with low or high fidelity. Low-fidelity models are often specified with pen and paper since thereby the models can quickly and easily be changed. High-fidelity prototypes are often designed with the support of graphical tools (Rogers, Sharp, & Preece, 2002). Their look and feel is often quite similar to the final product. At early stages of the UCP, the design specification characteristically is less detailed than in later iterations. Figure 1 illustrates a high-fidelity design specification of a mobile application.

Implementation of Prototypes

After having specified the mobile application in the design phase, the prototype can be implemented. In terms of interface design, a prototype represents a partial simulation of a product with respect to its final appearance and behaviour (Houde & Hill, 1997). Prototypes can be classified by their level of detail, range of functions and reusability. Similar to the design specification, the prototypes can be implemented with a low or high level of detail (Nielsen, 1994; Rogers, Sharp, & Preece, 2002). Low-fidelity prototypes normally are prototypes

Related Content