Personalized Mobile Applications for Remote Monitoring

Miguel A. Laguna, ETSI Informática, University of Valladolid, Valladolid, Spain
Javier Finat, ETSI Informática, University of Valladolid, Valladolid, Spain

ABSTRACT

The development of mobile applications is a challenging activity. The main problems are the limits of the mobile devices (in memory size, processing power, battery duration, etc.) and the diversity of target platforms, display sizes, or input modes (keypads or tactile screens). For these reasons, the software product line (SPL) development paradigm can improve the process of designing and implementing mobile systems. The authors’ approach to SPL development uses the package merge relationship of the standard UML to represent the variability in all the SPL design and implementation models. The combination of this technique and conventional CASE and IDE tools (Eclipse or MS Visual Studio) makes the developments of SPLs for mobile applications easier as it removes the need for specialized tools and personnel. This article presents a SPL that makes possible the remote monitoring of dependent people to facilitate their autonomy. The SPL generic architecture uses Bluetooth wireless sensors connected to mobile devices. These devices are remotely connected to a central system, which could be used in hospitals or aged person’s residences. Moderate cost sensors allow health parameters such as heart rate or oxygen saturation level to be controlled. Risk situations can also be detected using a range of predefined values or specific sensors. The diversity of individual situations and the resource limitations favor the use of the SPL paradigm, as only the required features are incorporated in each concrete product.

Keywords: Fall Detection, Mobile Application, Monitoring System, Software Product Line, Wireless Sensor

INTRODUCTION

The raising of life expectancy and the increase in the number of dependent persons in developed countries introduce new challenges for improving their quality of life (in a broad sense, we include here supervised patients, elderly, or handicapped people). Quality of life is associated with autonomy and mobility. The mobility of dependent people can come into conflict with frequent monitoring of health parameters under medical supervision or with risk detection. The increased costs of care and the geographic dispersion favor the deployment of personalized health services based on low cost ubiquitous systems, accessible to more people while reducing the medical costs. Consequently, one of the most interesting applications of mobile devices is the development of systems to increase the personal autonomy. These technologies have generated huge expectations but we should ensure that their costs are reasonable.
Different inexpensive sensors can provide data about heart rate, oxygen saturation, temperature, etc. In addition to supervised patients, other people with varying degrees of dependence can improve their level of autonomy: persons with different physical disabilities, the elderly who live alone (and have a risk of fall that can be detected using a suitable sensor), etc. The Global Positioning System (GPS) is another inexpensive data source, directly accessible from conventional mobile devices that can be combined with health parameters, completing the information about people location required in the detection of risk situations.

However, the big problem is that the personalization of an application to each person requirements and sensor or device characteristics can be unaffordable in terms of costs and development time. To make things worse, multiple platforms for mobile devices are appearing and evolving continuously (and several display sizes, color depth and variations in keypads or tactile screens can share the same platform). For these reasons we are working in a set of projects aimed to apply the product line paradigm of software development to the specific case of customizable mobile systems.

Software product lines (SPL) are based on the idea of separating the common and variable parts of the product line as reusable coarse-grained components (Bosch, 2000). This approach is demanding and requires a great effort by the companies that take it on. We aim to simplify the change from a conventional development process towards the product line paradigm. To achieve this, we use standard UML elements to represent the SPL architecture variations with conventional CASE and IDE tools (Laguna, González-Baixauli, & Marqués, 2007). We have used that approach to build an e-Commerce SPL and with several mobile systems.

This paper reports the practical experience in the development of a product line for remote monitoring systems, using a combination of sensors, (multiplatform) mobile devices and a server that can be consulted in real time from anywhere by medical personnel using a standard Internet browser. The rest of the article is organized as follows: Section 2 briefly presents the techniques and tools used for the SPL design and implementations. Section 3 is devoted to the description of the product line. In Section 4 the related work is analyzed and, finally, Section 5 concludes the paper.

SOFTWARE PRODUCT LINES: TECHNIQUES AND TOOLS

Feature diagrams represent the variability and commonality of software product lines and permit the configuration of each specific application to be selected. Feature diagrams are the more specific SPL techniques, aimed to handle variability and traceability at each abstraction level of the product line. The product line development requires these diagrams to represent the SPL variability and as a mechanism to obtain the configuration of features that represent the selected combination of variants for each specific application.

However, the optional features must be connected with the related variation points of the architectural models that implement the SPL. This link allows the automatic instantiation of the SPL generic architecture into each specific application. This is derived from the complete architecture selecting or not each optional feature. The selected feature sub-model, through traceability relationships, guides the composition of the pre-existing code packages. The precondition for the success of this process is the existence of traceability links from features to design and from design to implementation.

Our solution to this problem (Laguna, González-Baixauli, & Marqués, 2007) consist of expressing the SPL variability of UML models using the package merge relationship, defined in the UML infrastructure meta-model. This mechanism permits a clear traceability between feature and UML models to be established.

The technique consists of associating a package to each optional feature, so that all the necessary elements that model the feature remain located in a package (maintaining the
A Balanced Perspective to Perioperative Process Management Aligned to Hospital Strategy
www.igi-global.com/article/a-balanced-perspective-to-perioperative-process-management-aligned-to-hospital-strategy/124116?camid=4v1a

The Architecture and Early Findings of a Working SMS-Based System for Individuals with Mild to Moderate Depression
www.igi-global.com/chapter/architecture-early-findings-working-smss/75618?camid=4v1a