Dynamically Adaptable User Interface Generation for Heterogeneous Computing Devices

Mario Bisignano, Giuseppe Di Modica, and Orazio Tomarchio

Dipartimento di Ingegneria Informatica e delle Telecomunicazioni,
Università di Catania, Viale A. Doria 6, 95125 Catania, Italy
{Mario.Bisignano, Giuseppe.DiModica,
Orazio.Tomarchio}@diit.unict.it

Abstract. The increasing number of personal computing devices today available for accessing online services and information is making more difficult and time-consuming to develop and maintain several versions of user interfaces for a single application. Moreover, users want to access services they have subscribed, no matter the device they are using, always maintaining their preferences. These issues demand for new software development models, able to easily adapt the application to the client's execution context, while keeping the application logic separated from its presentation. In this work we present a framework that allows to specify the user's interaction with the application, in an independent manner with respect to the specific execution's context, by using an XML-based language. Starting from such a specification, the system will subsequently "render" the actual user's application interface on a specific execution environment, adapting it to the end user's device characteristics.

1 Introduction

Many different personal computing devices are available today to users: they range from full powered notebooks, to portable devices such as PDAs, and to smartphones. All of these devices allow users to access and execute different kind of applications and services, even when they are outside of their office. Moreover, the heterogeneity of the available wireless access technologies gives rise to new network scenarios, characterized by frequent and dynamic topology changes and by joining/leaving of network nodes. Due to the continuous mobility of users' devices and, as a consequence, to network dynamics, applications cannot rely on reliable and stable users' execution contexts. Context-aware mechanisms are needed in order to build adaptive applications, location-based services, able to dynamically react to changes in the surrounding environment[13,3]. Current approaches, both as traditional models for distributed computing and related middleware [4], do not fully satisfy all the requirements imposed by these environments.

One of the main requirements to be able to build ubiquitous and pervasive computing scenarios is the possibility for the user to access the same service from heterogeneous terminals, through different network technologies using different access techniques [12]. In order to accomplish this, while avoiding at the same time the burden of reimplementing from scratch the same application for a different device and/or for

L.T. Yang et al. (Eds.): HPCC 2005, LNCS 3726, pp. 1000–1010, 2005.

a different network technology, new programming paradigms along with the related middleware are needed, providing the adequate level of transparency to the application developer. In particular, from the developer's perspective, developing a new user interface and new content types each time a new device penetrates the market is not a feasible solution.

In this paper we present the architecture of a framework whose goal is to make the presentation layer of the application, i.e. the user's interaction level, adaptive and (as much as possible) independent from the specific execution context. The actual user interface will be dynamically generated at runtime according to context information. The presentation level (user interface), together with the user's interaction model and the associated task model is described at an high and abstract level, in a device independent way. Moreover, the system is supported by a set of adaptation components (and/or renderer), each one specific for the current execution context at the client side (user terminal characteristics, current network features, user preferences, etc). A given render is in charge of adapting the application's user interface for a specific execution environment, according to the actual end user device's features, which will be represented using the CC/PP standard for profile information representation. Depending on the user's needs (user preferences) and the application characteristics, this step can be done either off-line [1], thus distributing the result to the client later on, or dynamically on-line [16]. As further described in this paper, from an architectural point of view, the system has been structured in such a way to promote the dynamic insertion of new adaptation modules (even at runtime), specific for some functionality not foreseen in advance. A prototype of the framework has already been implemented, together with two "renderers": one for standard PCs, equipped with a complete J2SE environment (Java Standard Edition) and the other one for mobile phones equipped with the Java Micro Edition (J2ME) environment.

The rest of the paper is organized in the following way. Section 2 presents a review of related work in this area, trying to outline merits and limits of existing approaches. Then, in Section 3, the system architecture is presented, together with the description of its components and their behavior. Finally we draw the conclusions in Section 4.

2 Related Work

The development of services that all kind of end user's devices should be able to access, despite their different features, had given rise to several different approaches to the problem, both in academic and in commercial environment. Nevertheless, today many of existing approaches often address Web content fruition [14,6,7], having as target devices well defined categories of terminals (typically PDAs like Pocket-PC and cellular phones WAP-enabled). All of these approaches can be generally classified into some categories: scaling, manual authoring, transducing [14,6] and transforming [7]. Among these approaches, the one based on the description of the user interface by means of an XML-based vocabulary seems the most promising. Some of the system/languages aiming at this purpose, adopting an "intent oriented" scheme are: UIML [1], AUIML [2], XForms [17], Dygimes [9], Teresa [10]. In these systems only the interaction of the application with the user is described, but not its graphic components, whose