Interaction Markup Language - An Open Interface for Device Independent Interaction with E-Commerce Applications

Karl M. Göschka, Robert Smeikal
Vienna University of Technology
Gusshausstrasse 27-29/384
A-1040 Vienna, Austria
goeschka@acm.org

Abstract

Modern E-Commerce Applications tend to have a lot of different user interfaces, most important today are Java and HTML but also emerging technologies like Agents or WAP cellular phones or even three dimensional environments (VRML). The difficulty is to make all those interfaces work together with the same application in an almost similar and transparent way and to keep them synchronized with each other during the lifetime of the application. Our approach is to describe the interactions of an UI rather than the elements or components with an XML description called IML - Interaction Markup Language. Unlike UIML (User Interface Markup Language) those interactions are then assigned to technology specific interactions first, which in turn are then finally assigned to a particular implementation. An IML description is also stable against future developments: To enable the application for a new device, only the IML renderer has to be implemented.

1 Contribution

The contribution of this paper is to analyze a major problem today. This problem has been detected through several E-commerce applications the authors have built throughout the last three years.

The explosion of different ways to create user interfaces (UIs) for Web and network applications as well as the rapid development of the World Wide Web (WWW) itself has created a difficult situation for developers of information systems [5, 6]. Fueling this trend is an increasing variety of appliances that can be used for Internet access.

For enterprise-wide information systems it is not enough to worry about the components, that provide the implementation of the business logic, it is increasingly important to provide means for scalable and flexible presentation logic and open accessibility to the business logic, that can easily be enhanced during the lifetime of the system.

This paper examines the possibilities of appliance-independent user interface descriptions that can be deployed on any device to reduce the development time and cost as well as to increase the maintainability and accessibility of the supported services. Figure 1 shows the desirable architecture. To make it clear: We do not have a solution yet, but we truly admire the problem and want to share and discuss it with others.

The next section outlines the motivation in more detail, the following sections then propose the requirements and implementation environment for the new architecture. Then some existing client-side technologies are summarized as target technologies for our approach. The paper concludes with the description of ongoing and future work.

2 Motivation and Requirements for an Appliance Independent UI Description

Recently we can see many new ways to create user interfaces for distributed applications, especially for the Web. We can roughly divide them into markup languages and GUI (graphical user interface) frameworks with APIs (application programming interfaces). On top of that we encounter a large variety of appliances for Internet access, most notably today: PC, notebook, palm, cellular phone, multimedia phone and the classical circuit switched voice phone. Other appliances include two-way pagers, electronic messaging appliances, and systems for Web access via television (e.g. WebTV). Also emerging today are hybrid appliances: cellular phones with keyboards and smart phones which run palm PC operating systems.