Joachim Wolfgang Kaltz

An Engineering Method for Adaptive, Context-aware Web Applications

Herbert Utz Verlag · München

Informatik

Band 86

Zugl.: Diss., Duisburg-Essen, Univ., 2006

Bibliografische Information Der Deutschen Bibliothek:
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Dieses Werk ist urheberrechtlich geschützt.
Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Entnahme von Abbildungen, der Wiedergabe auf fotomechanischem oder ähnlichem Wege und der Speicherung in Datenverarbeitungsanlagen bleiben – auch bei nur auszugsweiser Verwendung – vorbehalten.

Copyright © Herbert Utz Verlag GmbH · 2006

ISBN-10 3-8316-0647-1
ISBN-13 978-3-8316-0647-4

Printed in Germany

Herbert Utz Verlag GmbH, München
089-277791-00 · www.utz.de

Contents

Chapter 1

Introduction

1.1 Theme Explanation

In software, networking can be used to provide the user with up-to-date information and access to services offered by a remote provider. In novel applications, networking is increasingly handled via Web mechanisms (Rheingold, 2002); corresponding Web-based systems support in particular applications accessed in a Web browser on a personal computer, and software for mobile systems that operates over the Web.

Web-based systems are increasingly complex and used for a wide range of purposes, possibly in changing circumstances, for example in various locations and with different devices. This has raised new issues related to applying the technology, in particular regarding usability of such systems. The amount of information provided to users of Web-based systems is increasing rapidly; this "content explosion" (Monaco et al., 2001) can result in confusion as to which information is relevant. Within many current Web applications, too much cognitive effort is required by the user to "recognize important relationships and recall key information" (Griswold et al., 2003). Furthermore, though Web-based systems have traditionally focused on information delivery, they are now also used as a platform for providing additional types of services, which may require complex interactions with the user (that is, interactions beyond mere activation of a link in a Web page). This raises the question of which services to provide in which situation, and how to integrate as seamlessly as possible services within the offering of a Web-based system, to reduce the amount of effort a user must expend to find and use a service.

Web resources are increasingly plentiful, but a user's cognitive resources are not unlimited. This is likewise true of the physical resources a user has access to in a specific situation. No matter how much bandwidth and computing power increase, there will always be scenarios in which these resources are insufficient, in particular in mobile computing scenarios (Forman and Zahorjan, 1994). Web engineers must address these problems of cognitive overload and physical constraints, as they restrain usability of Web-based systems and may even cause users to entirely reject a system.

Adaptation has been proposed as an approach to address these problems. In such an approach, a system is conceived in a manner such that it is capable of adapting its operation according to

the present situation of usage. Corresponding Web applications can assist the user in navigation, content and service selection according to known preferences and current circumstances: an application should present the user "the right thing at the right time in the right way" (Kappel et al., 2003). In addition, for wider acceptance in everyday usage, an application should "require human attention for only critical aspects of task execution that require their input" (Thayer and Steenkiste, 2003). Thus, a requirement for software can be that software should adapt its offering according to the situation; meaning that what is being offered, and how, should depend on the user and the *context* in which the user is using the system (Morse et al., 2000).

1.1.1 Adaptivity and Context

Varying terminology is used in computer science to describe the goals of adapting software to user needs, and of providing the conditions necessary for this adaptation. One needs to distinguish, on the one hand, the activity where a system engineer, or an individual user, explicitly and manually adapts an application to needs and preferences, and, on the other hand, automatic adaptation of an application's behavior, performed by a system.

Software supporting user-initiated individualization can be called adaptable, whereas software supporting system-initiated individualization can be called adaptive (Oppermann, 1994). A distinction in adaptability can be made according to the person performing the individualization. When the end-user tailors an application according to preferences, this activity can be called personalization. When a manufacturer configures or modifies software according to customer requirements, this can be called customization. When a customization is aimed at a large number and variety of customers, the term mass customization (Gilmore and Pine, 2000) can be used.

In this dissertation, we call a system adaptive when it is capable of automatically adapting an application's behavior, during application usage, according to the context in which the application is being used, that is, the "context of the interaction" (Dix et al., 2003). Additional terminology is sometimes used in research. The field of adaptive hypermedia uses the term personalization to describe the activity of systems which analyze the user's past behavior in order to decide what information to provide next, for example in tutoring systems (Brusilovsky et al., 1998). In mobile computing, the term customization is used to describe the goal of software adaptation to a user's location of use and device constraints (Badrinath et al., 2000); ubiquitous computing likewise uses this term in this sense (Kappel et al., 2003). Using the term customization for automatic adaptation can be misleading, as customization can also be used to describe an engineering activity, as discussed above.

To possess the ability to fulfill an individual user's needs, software must take into account several factors, such as the user's profile, current task or goal, and possibly additional factors such as location, time, or device used. The combination of all relevant factors in software usage can be termed the context, and thus a Web software that takes them into account is a context-aware application. The term context-sensitive is sometimes used interchangeably, see, for example, (Clerckx et al., 2005). In turn, the term context-adaptation is sometimes used to refer to the activity of systems basing their adaptation on context (Mani et al., 2004); this term can however be misleading, as it might also refer to the activity where a system adapts its knowledge about context itself, as opposed to adapting the application according to this context. To use informa-

tion about context, a system must determine this information during its operation: this activity is called context sensing, context reasoning, or context inference.

In summary, using the notion of context in software (that is, "context-aware computing") is a paradigm in which information about the situation of system usage is gathered and structured in such a way that this information can be used for (automatic) adaptation of software according to the situation. The motivation for context-awareness (and with it, adaptation) is, ultimately, to increase usability of systems. In an adaptive Web application, knowledge about context might be used at various levels, from the structuring and appearance of the user interface to the information and services provided.

1.1.2 Web Applications

A Web application is software which enables a person, via a user interface, to make use of information and service offerings provided on the World Wide Web (in short, Web). To describe such applications, a definition of the Web is then required; according to the Merriam-Webster Dictionary, the World Wide Web is:

> a part of the Internet designed to allow easier navigation of the network through the use of graphical user interfaces and hypertext links between different addresses – called also Web.

The purpose of a Web application is traditionally to provide users with an access point to an information system, respectively to several information systems integrated for external (Web) access at this point (Rheingold, 2002). Increasingly, these applications go beyond mere presentation of information, and provide many types of services, sometimes requiring complex interaction: in other words, these applications provide operational features in addition to information delivery.

This dissertation focuses on software designed for the Web in a professional environment. In this regard, a Web application belongs to a specific sort of network-centric software systems used between an enterprise and its customers, or within an enterprise. Using Web mechanisms for networking can be compared to other fundamental approaches as outlined in Table 1.1.

A significant particularity of Web mechanisms is that they are standardized: notably, the manner in which resources are addressed (URL[1]), and the way in which client and server communicate (HTTP[2]). Unlike a typical client-server application that requires a custom client and operates in a controlled environment, Web applications can be used by a general purpose Web client, such as a Web browser, by people not necessarily known to the service provider. The fashion in which a Web application is used is thus less predictable than in classic software scenarios (Dumke et al., 2003): though a Web application is often targeted at certain groups of users, the actual type of user and the usage scenarios can not be known precisely. This requirement, along with requirements following from the specification of the Web standards, results in specific constraints for Web applications, notably regarding user interaction with these systems.

[1] Uniform Resource Locators, see Appendix A

[2] Hypertext Transfer Protocol, see Appendix A

	Client type	*Service offering*
host	terminal	A user is directly connected to a host, via a terminal. The user chooses a specific transaction, and then interacts with this transaction.
client-server	custom	The user interacts with a client software that is located in the same network as a server. The client software has some knowledge of the application domain and relies on the server to provide functionality such as data persistence: for example, a client might be connected to a relational database management system (the server), and interact with several tables in the database.
Web	standard (Web browser, WML browser)	A Web server offers information and services at a specific location, identified via a URL, and Web clients communicate via the HTTP protocol with the Web server. The client has no knowledge of how the server implements its functionality.

Table 1.1: Network-centric applications in a professional environment

The goal of a Web application as understood in our work is not to recreate or replace existing information systems. The information systems themselves (referred to as "backend" in this regard) need not be engineered using Web technology, but the user's access to these systems is increasingly occurring via Web technology. To illustrate the difference, consider the following scenario: the Web offering of a health insurance company provides Web access to medical knowledge that can be queried and answered by a medical expert system (the backend). This Web application integrates a possibility for users to obtain knowledge from this medical expert system, but does not replace the expert system.

1.1.3 Web Engineering

Web Engineering is an emerging discipline that advocates (Deshpande and Murugesan, 2001):

> (...) establishment and use of sound scientific, engineering and management principles and disciplined and systematic approaches to the successful development, deployment and maintenance of high quality Web-based systems and applications.

While related to Software Engineering, Web Engineering must take into account, in addition to the technical particularities of Web applications (see Section 1.1.2) specific characteristics of Web-based systems, notably:

- Web-based systems are primarily content-driven (although they may in addition provide operational features). Navigation through this content must be achievable in a user-friendly manner.