Designing Web Applications

Gustavo Rossi
UNLP, La Plata, Argentina

Daniel Schwabe
Departamento de Informática, PUC-Rio, RJ, Brazil

INTRODUCTION

The World Wide Web (WWW) has become the most widely used platform for application development and information delivery. Web applications have evolved from static, read-only Web sites to current, collaborative, mobile, and pervasive information systems. Most companies are automating their core work flows using Web technologies; new businesses supported by the provision of complex Web services appear every day.

These application fields impose new modeling, design, and implementation requirements. Applications must have good performance, but they must also be usable and often adaptable to the individual user, his or her location, preferred interface device, and so forth. The time frame of the development life cycle must be short, which means we must improve design and implementation reuse. Applications evolve constantly and in unforeseen ways, so modularity is a must.

Web applications are different from "conventional" applications mainly because they are based on the hypermedia metaphor and because they allow users to access information by navigating through multimedia nodes that are connected by links. More complex structures, such as hierarchical indexes and landmarks, are often necessary to help the user find his or her way through the ocean of information. Successful Web applications provide good navigation topologies, helping the user to complete tasks without experiencing the "lost in hyperspace" syndrome.

In this context, conventional software engineering approaches fail to fulfill the needs of this application domain, if only because they neglect the navigational dimension of Web applications. They simply consider them just as a particular case of interactive applications; for example, they do not provide meaningful abstractions to model the unique features of this kind of software. In particular, treating a Web application just as any other transactional software, in which queries over data are posed using a browser as an interface, ignores the richness of the hypermedia paradigm for describing information systems. The same is true when considering a Web application just as a kind interactive application, in which the standard MVC (model-view-controller) paradigm (Knight & Dai, 2002) can be applied. While database- and programming-centric approaches can be used in the implementation step, conceptual modeling of Web applications requires a more careful approach.

Fortunately, the Web engineering community has come forth with a brand new set of methods and data models to address these issues. In the following section we survey the most interesting aspects of those methods.

BACKGROUND

The leading methodologies in this field, such as Web Modeling Language or WebML (Ceri, Fraternali, & Bongio, 2000; WebML, 2003), Object-Oriented Hypermedia Design Method, also called OOHDM (Schwabe & Rossi, 1998), UML-based Web Engineering or UWE (Hennicker & Koch, 2000), Object-Oriented Hypermedia or OO-H (Cachero, Gómez, & Pastor, 2000), or W2000 (Baresi, Garzotto, & Paolini, 2001), consider the Web-application-development life cycle as an iterative, incremental process, in which different models are specified before the implementation.

The design space is thus partitioned in basically three data models: conceptual, hypertext, and presentation models. Though each method provides its own primitives for describing each model, there is a general consensus on considering these three aspects separately.

The application or conceptual model describes the underlying application domain using well known design primitives: WebML uses data-modeling languages (such as the entity relationships), while OOHDM, UWE, and OO-H use object-oriented concepts in the syntax of UML (unified modeling language).

The conceptual model can be viewed as the data model in traditional applications. The navigation model, meanwhile, is built as a hypertext view over the conceptual model. There may be different views for different user profiles, providing distinct navigation paths or even navigation objects. Generally speaking, a navigation or hypertext schema describes the items that the user will navigate and the links he or she will follow to traverse

those items. The navigation schema is described using nodes and links, which abstract the usual components of a Web site.

To model these aspects, almost all methods use their own proprietary notations; UWE, however, has chosen to use a lightweight UML extension by defining stereotypes for each typical hypertext component such as navigation object, link, and index. In this way, it can benefit from all existing concepts and tools developed in the UML world.

In OOHDM, the high-level navigation structure of the application is described by a navigational contexts schema. A navigational context is a set of nodes sharing some properties (products of a brand, results of a query, etc.), and the navigational context schema describes all meaningful contexts together with indexes for accessing the nodes in a context. Contexts allow specifying the navigation alternatives for all its members at the same time, without having to detail it for each individual item. For example, it is possible to state, "From any given product, one can navigate to any of its related products," instead of having to say, "From product CD-ROM, it is possible to navigate to product CD Blanks," from, "From product Shaver, it is possible to navigate to product Shaving Blades," and so on.

All modeling methods provide a design armoury for describing abstract aspects of the user interface in terms of interface objects. These objects show how pages will look like though avoiding delving into the implementation aspects.

Once the different Web application models have been completely specified, a running Web application can be built. Different methods provide different strategies for mapping models into running applications. In general, implementing the conceptual model is straightforward using either conventional databases or object-oriented tools. The navigational views may be materialized or not, but they can be easily described using current server page technologies; interfaces can be built by using HTML (hypertext markup language) or XML (extensible markup language) and Extensible Stylesheet Language (XSL) tools.

Some methods go further. For example, WebML provides a set of tools for automatically generating the running application by performing a set of transformations on the previously described models (WebRatio, 2003).

The main contribution of these methods to the field of Web engineering has been threefold. First, they posited the idea that modeling Web applications involve more than defining conceptual models, that is, the introduction of the navigational modeling activity is essential to understand the dynamics of these applications (Rossi & Schwabe, 1999).

Second, their notations are based on well-known modeling practices (object orientation and entity relationship), which improve abstraction and reuse capabilities, (see, for example, Koch & Kraus, 2002), and provide a good framework for extending the models. In this sense, these methods have been recently used to specify customized applications (Abrahão, Fons, González, & Pastor, 2002; Matera, Ceri, Dolog, & Nejdl, in press; Rossi, Schwabe, & Guimaraes, 2001) and Web-based business processes (Brambilla, 2003; Schmid, & Rossi, 2004).

Third, they have introduced abstract interface specification (or presentation) as an important activity in the engineering process, factoring out interface technology from the design itself, a crucial aspect in times of continuous, fast technological and standards evolution.

The Web engineering community has also discovered a large set of Web design patterns that record and convey design experience (Garzotto, Paolini, Bolchini, & Valenti, 1999; Lyardet, Rossi, & Schwabe, 2000; Molina, Meliá, & Pastor, 2002; Rossi, Schwabe, & Garrido, 1997). These patterns allow the designer to reuse smart solutions that we (and other people) have found in successful applications while solving recurrent problems. These patterns have generated the architecture of Web frameworks, semicomplete designs in a particular application domain that can be refined and customized to obtain running applications (Fayad, Schmidt, & Johnson, 1999; Schwabe, Emeraldo, & Rossi, 2001). More comprehensive work toward obtaining a complete set of design guidelines from a system of patterns can be read in Van Duyne, Landay, and Hong (2003).

FUTURE TRENDS

Related Work

Since their inception, modeling and design methodologies have evolved quickly, and now we can say that the Web engineering field is entering adolescence. There are many other methods deserving mention, such as Web Site Design Method or WSDM (De Troyer & Leune, 1998), focusing on audience-driven modeling and design, and Web Unified Modeling Language, also called WUML (Kappel, 2001), devised specifically for ubiquitous Web applications.

Coming from a complementary perspective, there are proposals aimed at integrating conventional data models and database management systems with the Web platform. One of the first such approaches is Strudel (Strudel, 2003), which allows building Web sites as queries to a database through a specially designed language. An-