Exploiting UML Extensibility in the Design of Web Applications

Ekaterina Gorshkova
eVelopers Corporation
cathy@evelopers.com

Boris Novikov
Dept. of Computer Science, University of St. Petersburg
Universitetsky prosp.28
St.-Petersburg, 198504 Russia
borisnov@acm.org

Abstract

The application design and development methodologies based on UML are well-known, widely used in practice and proved to be critically important for object-oriented design. However, these methodologies and tools do not provide specific support for modelling issues peculiar to web application.

This paper defines an UML extension capable to refine the design of the client part of web application. Several new diagrams are specified with provide for precise definition of the content of web pages and navigation between them.

Keywords

UML, Web application design, HyperMedia design, Navigation, Browsing semantics

1 Introduction

A design and development of a Web application differ significantly from more traditional stand-alone and client-server ones. It usually has to comply with several kinds of requirements like scalability, security (which are common for client-server and web), and portability of the client software (which is more specific for web applications). They are supposed to provide dynamic generation of the user interface, advanced presentation features, and be easily customizable. These requirements make the design of such an application a challenging task.

Several research efforts addressed hypermedia application design focused on data-intensive web-sites [1, 4, 5, 6, 7].For data model they use entity-relationship model or its extension, for navigation they provide predefined navigation constructs.

Another recent and the closest to us approach is WebML language [2], which provides orthogonal models for designing structure, composition and presentation. However, it uses its own restricted notation and fails to express advanced composition and navigational constructs. We avoid this kind of limitations relying on the power of UML.

Modeling of web applications with UML is described in [3]. This work covers various aspects of the topic. However, this approach uses only class, i.e. static, diagrams to represent both behavioral and structural things, reducing the clarity. Another point is that the stereotypes chosen for this notation are bound to the certain technologies, complicating the creation of conceptual abstract design.

We provide the UML-based methodology for modeling the content of web pages (composition diagram) and navigation between them (navigational diagram). This methodology gives possibility to create an abstract hypertext view regardless of

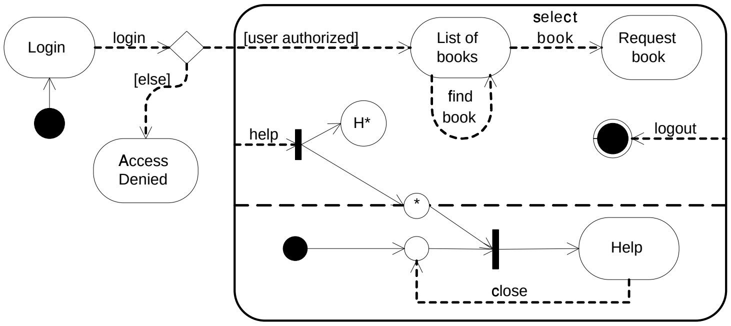

Figure 1: Example of the Navigation diagram.

the technologies used in the client and server side. The final result of such design is a precise definition of a number of pages interconnected with hypertext links. The tool that supports our notation can also provide the prototype of user interface generated from the diagrams.

2 Navigation Diagram

The navigational diagram is implemented as a special case of the UML statechart diagram. It is intended to model navigation issues of the hypertext and helps to visualize the behavior of the user in the system. The whole web application is modeled as state machine attached to the abstraction of the user. When browsing the Web application the user comes to various pages. Each of the pages can be modeled as state and browsing as transition between states. The user leaves the state when some event occurs. This event is initiated by user (when he clicks the hyperlink or presses submit button) or by the system (when he is redirected to some page after a certain amount of time). The user events are shown with dashed line, the system with solid line. Pages are presented in this diagram regardless of their content. Links that exist on many pages are modeled as high-level transitions. An example of the navigation diagram for the simple library application is shown on figure 1.

Sometimes we need possibility to display to the user several documents at once in different browser windows. These windows can be peers or child window can be opened as a pop up from the main one. The general idea is to model multiple windows as concurrent regions of the composite state. This decision enables us to see for each moment what windows do we show to the user. This also helps to define the when the window should be opened or close avoiding lost windows.

This diagram brings the client-side view at glance. It shows the linked collections of pages presented to the user, the alternatives of the browsing and exceptional situations.

3 Composition Diagram

The composition diagram is a special case of class diagram. We use it to express structure of the web pages and identify their content: how they are connected together and what data is carried from one page to another. We also show how pages are related to business data that they represent. The composition diagram provides static view of web pages, rather than behavioral features. Only links that are essential for structure are shown in this diagram and neither conditional nor exceptional situations are represented here. This diagram can be used for generating JSP and HTML code.

An example of such diagram is shown in the figure 2.

The main notion of the composition diagram is the page, defined as an autonomous block of screen. Each screen in the navigation diagram is mapped into several pages in the composition diagram. The tool may provide links from pages to screens and vice versa to show their relationship. A page is modeled in composition diagram as a class stereotyped <<page>>. A page may play role of container for other pages. Nested pages are modeled as aggregated classes.

The page has elements like buttons, links and input fields. They are modeled as attributes of the corresponding page. The <<form>> stereotype is a child of <<page>>. It is used to model HTML forms.

After defining pages we need to connect pages to make hypertext. Usually the user jumps to other page when he clicks

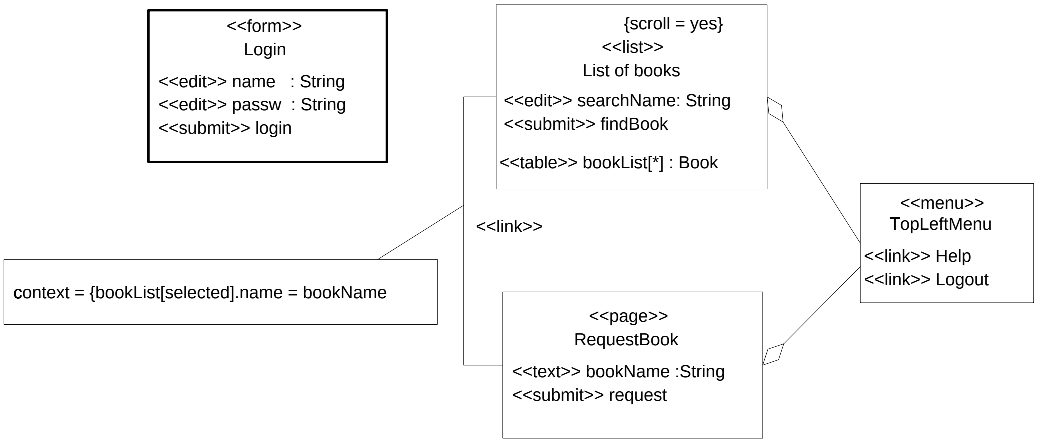

Figure 2: Example of the Composition diagram.

the hyperlink or presses submit button. This behavior is reflected in the navigational diagram. However, we need to show what information is carried from the source page to target one. The navigable association between source and target pages is stereotyped <<`link`>>. Each <<`link`>> has a tag *context* with expression as value. The expression determines what information to show in the target page depending on the state of the source page. The expression can be written in the human language, pseudocode or OCL. If *context* is empty, then pages are connected in a free way and no information is sent from source page to target one.

4 Acknowledgement

The work of the second author was partially supported by Russian Foundation for Basic Research under grant 01-01-00935.

References

[1] P. Atzeni, G. Mecca, and P. Merialdo. Design and Maintenance of Data-Intensive Web Sites. In *Proc. EDBT'98*, Lect. Notes in Comp. Sci. v. 1377, pages 436–450. Springeer-Verlag, March 1998.

[2] Stefano Ceri, Piero Fraternali, and Aldo Bongio. Web Modeling Language (WebML): a modeling language for designing Web sites. *Computer Networks*, 33(1-6):137–157, 2000.

[3] J. Conallen. Modeling web application architectures with UML. *Comm. ACM*, 42(10), 1999.

[4] M. F. Fernandez, D. Florescu, J. Kang, A. Y. Levy, and D. Suciu. Catching the Boat with Strudel: Experiences with a Web-Site Management System. In *Proc. ACM SIGMOD Intnl. Conf.*, pages 414–425, May 1998.

[5] Piero Fraternali and Paolo Paolini. Model-driven development of web applications: the autoweb system. *ACM TOIS*, 18(4):323–382, 2000.

[6] F. Garzotto, P. Paolini, and D. Schwabe. HDM - A model-based approach to hypertext application design. *ACM TOIS*, 11(1):1–26, 1993.

[7] Tomas Isakowitz, Edward Stohr, and P. Balasubramanian. RMM: A methodology for structured hypermedia design. *Comm. ACM*, 38(8):34–44, 1995.