

文章编号: 0529-6579 (2000) S2-0185-06

浅析 Web 应用程序对象建模方法*

潘志明

(中山大学计算机科学系, 广州 510275)

摘 要: 根据 Web 应用程序的特点, 从建模方法的四个组成部分比较系统地讨论了 Web 应用程序的对象建模方法. 建模过程很重要, 它是描述如何获得模型的指导. 结论都反映在建模过程中.

关键词: Web 应用程序; 面向对象; 建模方法; UML

中图分类号: TP311.52 **文献标识码:** A

本文讨论的 Web 应用程序 (以下简称为 Web 应用), 是指以浏览器 (Browser) 作为用户界面的 Web 应用. 方法是指导人们进行软件开发的一个框架. UML 的发起者之一 James Rumbaugh 认为建模方法应该包含以下四个组成部分^[1]: ① 一组建模的概念; ② 建模概念的符号表示及其视图; ③ 建模的过程; ④ 建模过程中的要点和主要原则.

由于建模方法本身并不描述应用领域的特点, 所以本文先分析 Web 应用程序的特点, 然后按照以上这四个部分的思路去探讨 Web 应用的对象建模方法.

1 Web 应用程序

Web 应用与网站并不是同一概念. Web 应用是利用网站作为其前端的应用程序. Web 应用有如下特点: ① Web 应用是通过 Internet 对外开放的, 即它的潜在用户数量可以是很大; ② Internet 的潮流变得很快, 过时的 Web 应用很快就被人们淡忘; ③ 从来没有别的软件应用程序会象 Web 应用那样地注重用户界面; 也从来没有象 Web 应用那样高风险的应用程序. 第一点要求 Web 应用必须具有高性能 (performance) 的, 以及较好伸缩性 (scalability) 和可靠性 (reliability) 的体系结构. 第二点则要求具有较好的应变能力. 因此, 建模是 Web 应用开发过程中的重要而且是必要的一个环节. 第三点强调了用户界面创造性设计的重要性, 同时也强调了演示层 (Presentation Layer) 建模在 Web 应用建模中的重要性^[2]. 通常, Web 应用是三层的 Browser/Server (简称 B/S) 应用, 分为演示层 (Presentation Layer)、业务层 (Business Layer) 和数据层 (Data Layer). 这里的“层”指的是一个能够响应请求提供服务的软件包. B/S 应用程序和 Client/Server (简称 C/S) 应用程序的区别

* 基金项目: 广州市科委资助项目 (98-z-093-01)

收稿日期: 2000-10-15; 作者简介: 潘志明 (1974~), 男, 硕士.

主要体现在演示层上,所以 Web 应用建模方法的研究重点是其演示层。Web 应用演示层的实现技术共同点是,把演示的内容和程序逻辑分离。本文把演示的内容称为 HTML 模版,把 Web 服务器上的程序逻辑称为服务器端处理,把 HTML 模版经服务器端处理加工后的产物称为 HTML 页。现归纳演示层特点如下:① 演示层分为 HTML 模版和服务器端处理两部分;② 无论采用何种服务器端技术,最终由 Web 服务器传送给浏览器的都是 HTML 页。③ 服务器端处理可以由 ASP^[3]、WebClass^[4]或 JSP^[5]等实现,主要有以下 3 个功能:① 接收用户输入;② 替换(插入)功能,即根据服务器的业务逻辑状态对 HTML 模版中的某些部分进行动态地替换;③ 改变服务器的业务逻辑状态。

2 面向对象的建模概念

任何模型都是在—组建模概念的基础上建成的,Web 应用也不例外。面向对象技术最核心的建模概念是类和用例。利用面向对象技术,可以更直接地描述现实世界,构造出模块化的、可重用的、可维护性好的软件,并能控制软件的复杂性和降低开发维护费用。面向对象的开发模型包括:需求模型、分析模型、设计模型、实现模型和测试模型。

Web 应用(特别是演示层)建模首先要处理好以下几个关于建模概念的问题:

2.1 如何得到类

客户关心的不是应用有什么类,而是关心应用系统能够提供的功能。所以,结构化的设计方法主要考虑的是功能及其分解。采用面向对象的方法更有利于处理需求的变化,它利用用例把功能映射为类。具体地说,可以参考应用领域的概念模型,通过对用例的事务序列的分析而得到类。另外,还可以使用 CRC(class responsibility collaborator)卡帮助确定类。

2.2 如何细化类

从最初确定类到用代码实现类,有一个从粗到精逐步细化的过程。通常主要有 3 个层次的视角去看类:

(1) 概念层(分析模型):这里描述的是应用域中的概念。这些概念与实现它们的类有关(类是这些概念的抽象),但通常并没有直接的映射关系。

(2) 规范层(设计模型):这里描述的是接口。这个接口可能因为不同的实现环境、不同的运行特性或者不同的买主具有很多种不同的实现。

(3) 实现层(实现模型):只有在这种观点中才真正有严格意义上类的概念,它揭示了软件的构成情况。

我们把分析模型^[6]中的类称为分析类,分析类与具体实现无关。相应地称设计模型中的类为设计类。设计类是与具体的实现语言相关的,是具有语言特定细节的类。可以由分析类导出设计类,但考虑到实现语言的特定细节,分析类和设计类不一定是一一对应的。而实现模型中,用具体的编程语言的源代码实现的类,我们称之为实现类。由于设计类已经考虑到具体实现语言的特定细节,所以设计类和实现类往往是一一对应的。利用代码生成工具,可以把设计类转换成实现类的类框架结构。利用逆向转出工程代码工具,可以把实现类转出为设计类。总之,从分析模型、设计模型、到实现模型,类被逐步地细化。而分析类则是通过对用例的分析,并参考应用领域的概念模型而得到的。

2.3 如何处理界面

如何把界面映射为类对 Web 应用演示层建模非常重要。根据上节的结论, 界面类同样有从分析模型、设计模型、到实现模型, 逐步地细化的过程。在分析模型, 界面被抽象为边界类。分析用例时, 边界类的主要任务是将执行者对系统的行为翻译为系统内部事件, 并且当这些系统内部事件执行后, 将执行者感兴趣的信息反馈给执行者。在设计模型, 由于设计类是语言相关的, 所以采用不同的 Web 服务器端处理技术, 会得出不同的演示层设计类。本文认为, 如果能根据 Web 应用演示层的特点, 提取出各个 Web 服务器端处理技术的共性, 根据这个共性则能够得到通用性更好的演示层设计类。因为这种设计类反映了共性, 所以它可以用各种 Web 服务器端处理技术去实现。

3 UML 及其对 Web 应用演示层的扩展

为了构建、检查和操纵模型, 人们需要一些记号去表示建模概念。建模语言是建模概念及其符号表示的综合体。建模概念是建模语言中的元模型, 其符号表示则是建模语言的表示法。

3.1 标准建模语言 UML

从 1989 ~ 1994 年, 建模语言的数量从不到 10 种增加到了 50 多种, 一度爆发了“方法大战”。多种建模语言的并存, 极大地妨碍了用户间的交流。标准建模语言 UML^[7]的出现结束了这种混战的局面。UML 的定义包括 UML 语义和 UML 表示法两个部分。UML 语义通过其元模型来严格定义。同时, UML 语义还支持对元模型的扩展定义。

UML 表示法定义了 UML 的表示符号, 为建模者和建模开发工具的支持者提供了标准的图形符号和正文语法, 使用这些图形符号和正文语法为系统建模就可以创建标准的系统模型。为了能支持从不同角度考察系统, UML 定义了以下 5 类模型图: 用例图、静态图、行为图、交互图和实现图。UML 是一种先进实用的建模工具。

3.2 UML 对 Web 应用演示层的扩展

UML 通过对元模型的扩展定义来支持各种具体的实现语言。换言之, 利用 UML 的扩展机制, 可以设计出具有特定语言细节的设计类。

UML 的扩展机制^[7,8]定义了构造型 (stereotype)、标记值 (taggedvalue) 和约束 (constraint) 3 个具体的结构 (construct)。其中, 构造型是 UML 最重要的内建扩展机制。从作用上讲, 构造型是建模时引入的一类新的元模型元素。它是现存的元模型元素的一个子类。构造型可以引入新的标记值和约束, 以及新的图形表示。任何模型元素可以被最多一种构造型标记。限于篇幅, 本文仅介绍构造性技术。

前文分析 Web 应用演示层的特点时, 提出了 3 个重要的概念: HTML 模板、服务器端处理和 HTML 页。HTML 模板是开发过程的中间产物, 无需建模。HTML 页是 Web 服务器对浏览器的响应, 而服务器端处理是对各种 Web 服务器端处理技术共性的抽象。

可以为服务器端处理设计《服务器端处理》类 (读作“服务器端处理类构造型”; 其中书名号表示构造型, 下同); 为 HTML 页设计《客户端页》类、《表单》类、《帧集》类和《目标帧》类。《客户端页》类与《表单》类、《帧集》类构成整体和局部的关系。《表单》类的属性是输入域 (Input Field), 《帧集》类的属性是帧, 它们都没有方法。《目标帧》类既没有属性也没有方法, 它的引入是为了可视地表现帧机制 (见图 1)。另外, 把

客户端脚本文件抽象为《客户端脚本》类. 上面主要讨论了类构造型, 下面探讨它们之间的关联构造型. 《客户端页》类之间、从《客户端页》类到《服务器端处理》类存在《链接》和《目标帧链接》关联. 《服务器端处理》类之间、从《服务器端处理》类到《客户端页》类存在《重定向》关联. 从《服务器端处理》类到《客户端页》类存在《加工》关联. 从《表单》类到《服务器端处理》类存在《提交》关联.

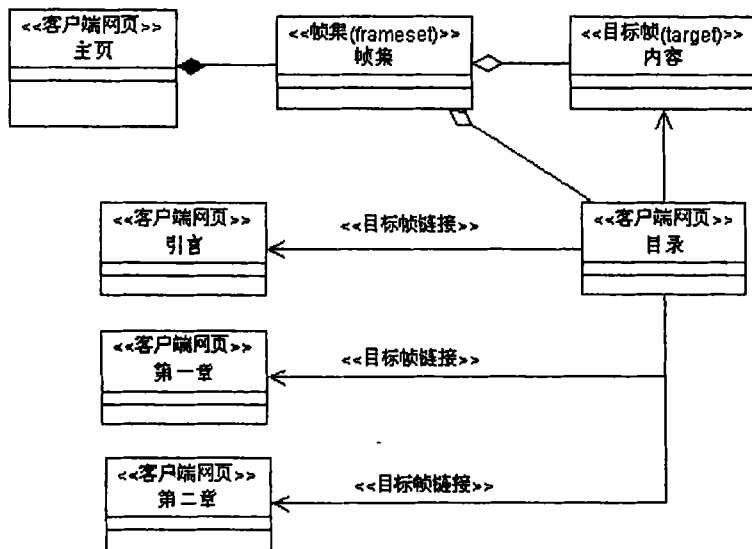


图 1 一个带帧的 HTML 页的设计类类图

通过以上这些类构造型和关联构造型, 可以为 Web 应用演示层画出设计类类图.

4 Web 应用建模开发过程

建模过程是关于如何利用建模概念及其表示符号去建立模型的路线. 不同的建模过程最后有可能得到相同的模型. 建模过程很重要, 它是描述如何获得模型的指导. 所以, 本文的研究结论最后都反映在建模过程中. 但是, 建模过程不能代替建模时需要的创造性, 建模过程只能指出哪里需要人的创造性. 建模过程中的要点和主要原则就是关于这些创造性的行之有效的经验的总结. 本节主要讨论 Web 应用的建模过程及其中的一些要点和主要原则.

4.1 演示层建模

Web 应用程序演示层的建模是一个从分析用例得到分析类、从分析类得到设计类、从设计类得到实现类的过程.

用例提供了一个让 Web 应用开发的参与者(包括客户、分析员、艺术指导、系统建筑师、网页设计员、程序员等)表述它们对 Web 应用解决方案的构思的途径. 分析类的类图从用户界面的角度, 描述 Web 应用的用户是如何被参与者(角色)激活, 并执行相关的一组特定的事务序列. 在这个阶段暂时还不知道(也无需知道)这些网页是什么样子的; 可以称之为“概念网页”. 在类图中, 可以用分析类中的边界类来表示“概念网页”. 由于分析类类图主要把握的是“概念网页”及其相互的连接关系, 可以把分析类类图称为

导航图^[2] (见图 2). 可以对每个执行者画一张导航图.

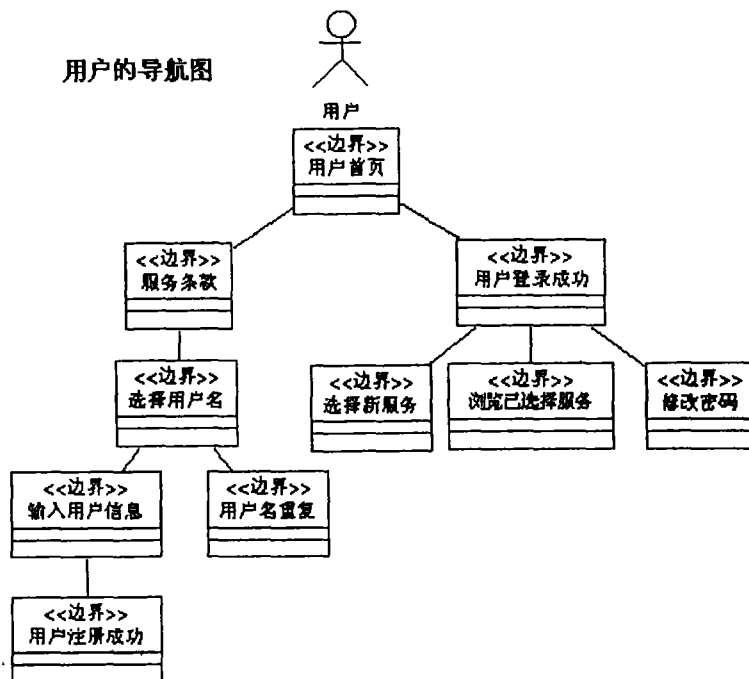


图 2 导航图

一旦用例的导航图基本确定, 就可以着手做设计类. 分析类中的“概念网页”, 对应着设计类中的《客户端页》类和《服务器端处理》类. 具体地说, “概念网页”有可能对应客户端页, 也有可能对应经服务器端处理加工过的客户端页. 那么如何判断导航图里的“概念网页”应该对应哪一个呢? 可以通过服务器端处理的 3 个功能来加以判断: ① 是否需要处理用户的输入? ② 是否需要读取服务器的业务逻辑状态信息? ③ 是否需要修改服务器的业务逻辑状态信息?

接着, 可以根据需要创建《表单》类和《帧集》类. 这样, 演示层的“蓝图”变得越来越清晰, 在导航图中考虑得不够周全的地方逐步被暴露出来. 一般可以为每一个用例画一张演示层设计类类图. 当类图中的设计类基本确定下来后, 可以添上关联以及一些较为明显的细节, 例如表单中的域.

在画好设计类的类图后, 可以着手让网页设计员根据设计类的类图, 设计 HTML 模板 (实现类), 从而实现从设计类到实现类的转变.

4.2 三层 Web 应用建模

上节基本上解决了如何对演示层进行建模. 下面简要探讨如何对三层的 Web 应用建模. 同样从分析类开始. 在用例驱动的设计方法中, 在用例确定后可选择风险最高的用例, 用顺序图来确定该用例的事件流程. 用例有可能有多个流程, 给每一个流程都画一个顺序图. 可以参考用例的“故事情节”确定用例的各个流程. 分析类的顺序图无需每次都从用户首页开始, 从相关的“概念网页”开始就可以了. 分析类的顺序图画好后, 可以拿去跟用户协商. 用户同意后, 可以着手做设计类. 可以使用顺序图去帮助确定业务层和数据层的设计类. 当设计类基本确定之后, 可以用三层服务模型类图对其三层服务的类及其

相互关系进行建模,并最终生成实现类。

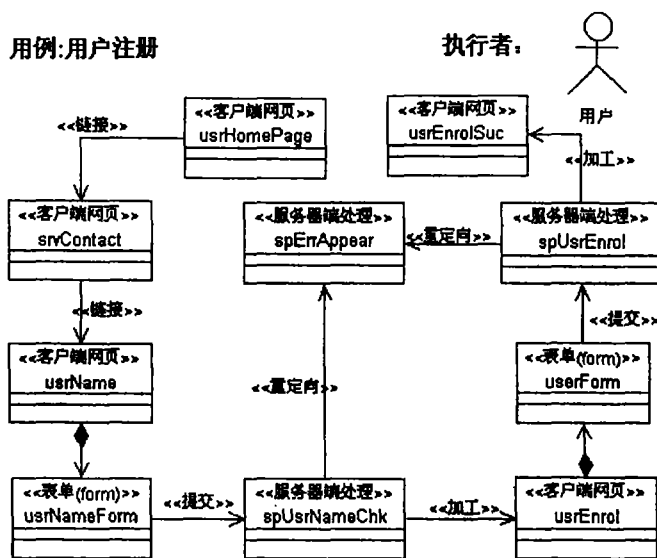


图 3 演示层设计类类图

参考文献:

- [1] JAMES R. What is a method? Rational software whitepaper. <http://www.rational.com>, 1995, Oct.
- [2] STAN W, PER K. Unifying the creative design process and the software engineering process, a rational software & context integration whitepaper. <http://www.rational.com>, 1999, Jul.
- [3] STEVE B, MICHAEL M. Visual InterDev6 开发使用手册. 北京:机械工业出版社, 1999. 92~122.
- [4] WANG J. Develop an E-commerce site using WebClasses. <http://www.devx.com>, 2000.
- [5] HALL M. JavaServer Pages (JSP) 1.0. <http://apl.jhu.edu/~hall/java>, 1999.
- [6] 周之英. 现代软件工程(中)——基本方法篇. 北京:科学出版社, 2000.
- [7] Object Management Group. OMG unified modeling language specification. 1999:1-1~1-11, 2-65~2-73.
- [8] SINAN S A. Extending the unified modeling language (UML). <http://home.earthlink.net/~salhir>, 1999.

On Object-Oriented Modeling Method of Web Applications

PAN Zhi-ming^{*}

Abstract: This paper analyses the characteristic of Web application and then probes into this problem according to the four components of modeling method. Process is important for modeling, for it draws up guidelines for description on how to produce a model. Therefore, the conclusions in this paper demonstrate themselves in the modeling process.

Keywords: Web application; object-oriented; modeling method; UML

^{*} Department of Computer Sciences, Zhongshan University, Guangzhou 510275, China