

# MDHDM:基于 MDA 的 Web 应用程序开发方法

段 雷 万建成

(山东大学计算机科学与技术学院 济南250061)

**摘 要** 随着 Web 应用程序的复杂度逐渐增加,人们对系统化开发 Web 应用程序的方法的要求日益迫切。本文提出了一种基于 MDA 的 Web 应用程序开发方法——MDHDM,利用 MDA 的特性弥补了已有开发方法的缺陷。本文中,我们介绍了 MDHDM 的设计步骤并说明了整体结构、相关模型和模型间映射规则。最后通过分析一个开发实例,进一步说明了 MDHDM。

**关键词** Web 应用程序,开发方法,MDA,模型驱动

## MDHDM:A MDA Based Developing Methodology for Web Applications

DUAN Lei<sup>1</sup> WAN Jian-Cheng<sup>2</sup>

(School of Computer Science and Technology,ShanDong University,Jinan 250061)

**Abstract** With Web application being more complex,there are more and more requirements for systemic developing methods for Web applications. In this paper,we present MDHDM,a model-driven method based on MDA for the development of Web applications. With the features of MDA,We try to remedy some deficiencies in current methods. We also introduce the developing steps of MDHDM and the MDHDM's architecture,models and mapping rules. At the end of the paper,we present a case to illustrate our proposal.

**Keywords** Web application,Developing methodology,MDA,Model driven

## 1 引言

近年来,随着 Web 应用程序由以文档为中心向以应用为中心发展,以往简单的基于页面的开发方法不再适用;而由于与普通应用程序的本质区别,又使得传统的软件工程的方法在 Web 应用程序的开发中显得力不从心。

面对这些问题,一些系统化开发 Web 应用程序的方法相继出现,像 OOHDM、RMM 和 UWE 等等。它们将传统的软件工程技术 and Web 应用程序的超媒体特性较好地结合起来,取得了一定的成功。然而这些开发方法也具有一些不可忽视的缺陷<sup>[3]</sup>,主要表现在:

- (1)模型之间缺少清楚和自动的映射。
- (2)各层的模型没有一致的建模形式,无法统一管理。
- (3)内容层、超文本层和展示层三个层次之间的关系不密切,模型间容易出现不统一。
- (4)缺少完善的开发过程支持。

这些缺陷妨碍了上述开发方法在复杂的 Web 应用程序开发中得到广泛应用。而 OMG 提出的 MDA<sup>[1,2]</sup>恰恰可以很好地解决上文提到的前三个问题。MDA(Model Driven Architecture)将软件系统

的模型分离为平台无关模型 PIM 和特定平台模型 PSM,同时又能通过映射规则将它们统一起来,以这样的方式试图去摆脱需求变更所带来的困境。

UML 作为 MDA 的基础技术之一用统一的形式描述面向对象的建模结构,而 MDA 的另一个基础 MOF(Meta Object Facility)<sup>[3]</sup>为模型驱动的元数据管理提供了有力的支持。MOF 可以支持对不同领域的不同建模结构的定义,而这些由遵从 MOF 规范的元模型定义的元数据可以由 MDA 工具统一管理并按照一定的映射规则进行转换。

鉴于 MDA 的上述特点和已有 Web 应用程序开发方法的不足,我们提出了一种基于 MDA 的 Web 应用程序开发方法 MDHDM(Model-Driven Hypermedia Design Method)。MDHDM 具有不同的 PIMs 和 PSMs 以及不同模型间的映射规则来支持 Web 应用程序的开发。本文第2节概述了 MDHDM;第3节通过一个图书馆管理程序的局部设计来说明 MDHDM 的应用;最后给出了结论和将来仍需完善的工作。

## 2 MDHDM:模型驱动的 Web 应用程序开发方法

MDHDM 以 OOHDM<sup>[5]</sup>为基础,针对 Web 应

用程序提出了一种迭代式设计开发过程。它共由三个步骤组成,分别是概念设计、导航设计和展示设计。对应的产品分别是:概念模型、导航模型和展示模型,这些模型构成了 MDHDM 的 PIMs。在设计结束后,利用基于 MDHDM 的工具可以自动化生成 PSMs 和系统的实现代码。下面我们将分别说明 MDHDM 的整体结构,PIMs 和 PSMs。

### 2.1 MDHDM 结构

在 MDHDM 中,我们共定义了三个平台无关模型(PIMs),包括概念模型、导航模型和展示模型;两个平台相关模型(PSMs),分别对应于系统的 EJB 模型和基于 OOHDM-Java<sup>[4]</sup>的 Web 层模型,以及模型之间的映射规则,这些规则可以实现概念模型-导航模型、导航模型-展示模型、PIM-PSM 以及 PSM-Code 映射的自动化。最后将生成基于 J2EE 的 Web 应用程序的实现,包括 EJBs、Servlet、Context Class 和 JSP。图1描述了 MDHDM 的整体结构。

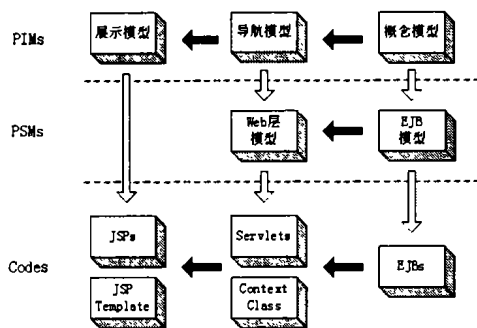


图1

### 2.2 PIMs

如前文所述,MDHDM 的三个设计步骤相应的产品是概念模型、导航模型和展示模型,它们构成了 PIMs。下面将分别说明这三个模型。

**2.2.1 概念模型** 概念模型是基于系统用例的,我们使用 OO 建模技术对领域模型的数据结构和行为进行描述。概念模型作为 Web 应用程序的核心部分针对的是业务逻辑,而与导航和展示无关,从而保证与导航模型和展示模型的松耦合。概念模型使用了 UML 进行描述,在将来从概念模型到导航模型的映射中,类通常映射为导航节点而关联通常映射为链接。

**2.2.2 导航模型** 超媒体特性是 Web 应用程序极为重要的特性,这决定建立合适的导航模型是所有 Web 应用程序开发方法重点所在。

MDHDM 的导航模型由导航类(Navigational Class)、导航访问器(Navigational Accessor)、菜单(Menu)等导航节点以及连接这些节点的关联组成。

在系统的实现中,将根据导航类生成动态网页的内容和链接。导航类包括基本导航类(Basic Navigational Class)和装饰导航类的导航上下文(Navigational Context)。

基本导航类提供了节点的必需信息,导航上下文根据节点所处的环境对导航节点进行装饰。由于导航上下文是可嵌套的,因此使得导航模型更加简明灵活。

导航访问器提供了对相关节点的导航方式,如 Index, Guided Tour, Query 等等。

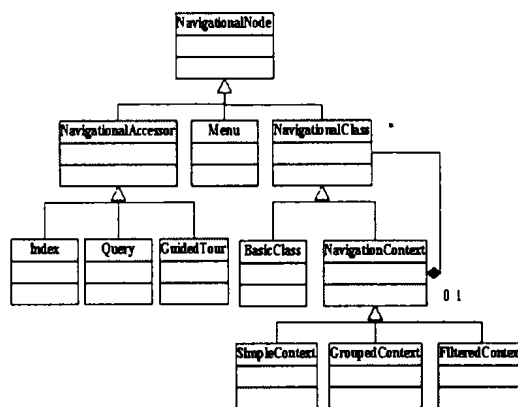


图2 导航元模型局部

MDHDM 为导航模型定义了遵从 MOF 的元模型,而不是像 N. Koch 那样用 UML Profile 对 UML 进行扩展用来描述导航模型<sup>[6]</sup>。我们这样做是考虑到 UML 的语义与导航语义相差比较大,用 UML 描述导航模型不够自然,而且 UML 复杂的元模型使得导航元模型过于复杂,不利于导航模型的管理和自动生成。在图2中给出了 MDHDM 的导航元模型的局部描述。

**2.2.3 展示模型** 在传统软件设计中,往往忽略界面设计,而将其推迟到实现时利用 IDE 进行。这样界面模型设计与概念模型设计脱节,容易造成两者不统一。在 MDHDM 中将展示设计作为整个设计过程中的重要部分来看待,并通过自动化映射保证展示模型与概念模型、导航模型的一致性。藉此克服了传统软件设计方法的不足。

在 MDHDM 中我们沿用了 OOHDM 使用的 ADV(Abstract Data View)表示展示模型。这样展示模型仅仅示意性地提供了界面元素的位置、尺寸和颜色等信息,而没有限定最终界面的展示形式。这样避免了平台不同而引起的设计差异<sup>[6]</sup>。

出于与导航模型类似的考虑,我们也为展示模型定义了遵从 MOF 规范的元模型,为简明起见,此处不再详细说明。

### 2.3 PSMs

在 MDHDM 中,由 PIMs 生成的 PSMs 是基于 OOHDM-Java 体系结构的。OOHDM-Java 将 OOHDM 开发方法应用于 J2EE 平台,完善了 J2EE 的 Web 层结构,进一步将导航与展示分开,从而提供了一个更为完善的 MVC 结构<sup>[4]</sup>。

MDHDM 中的 PSMs 包括 EJB 模型和 Web 层模型,分别由概念模型和导航模型生成。由于基于 Web 的应用程序的界面结构比较简单,我们将把展示模型直接映射为代码而不再提供 PSM。

EJB 模型由概念模型映射生成,并利用 UML Profile for EJB 来进行描述。概念模型中定义的类将映射成为 Entity Beans 类或 Session Beans 类。对于导航节点相应的类,在 EJB 模型中将实现 NavigationalView 接口。

Web 层模型由导航模型映射生成。导航模型中的导航节点类对应于 Web 层模型中的节点(Node)。由于在不同上下文中节点中的内容可能不同,因此要为不同上下文定义不同的生成器(Node-Creator),生成器根据导航上下文(Navigational-Context)生成正确的节点。

### 3 MDHDM 应用示例

在上一节中,我们给出了 MDHDM 的概要说明。在本节中我们将给出 MDHDM 的应用之一,并以一个基于 Web 的图书馆管理程序的一部分作为示例,进一步介绍 MDHDM。为简明起见,我们只说明了系统的结构模型而省略了行为模型。

#### 3.1 PIMs

如前所述,应用 MDHDM 将设计分为三个步骤:概念设计、导航设计和展示设计。概念设计使用了面向对象的设计方法,根据系统用例得到用 UML 描述的系统的领域模型。图3是图书馆管理系统中类模型的局部描述。

概念设计完成后将进行导航设计。导航设计有两个内容,分别是定义导航上下文和定义系统导航结构。如图4中所示,导航节点 Book 有三个导航上下文,即 byTitle、byKind 和 byRelated。在该例中,导航上下文之间没有标出关联,默认为三个上下文可以相互转换。为了便于表示,在下文中借用了 N. Koch 定义的符号<sup>[6]</sup>来描述导航模型和展示模型。

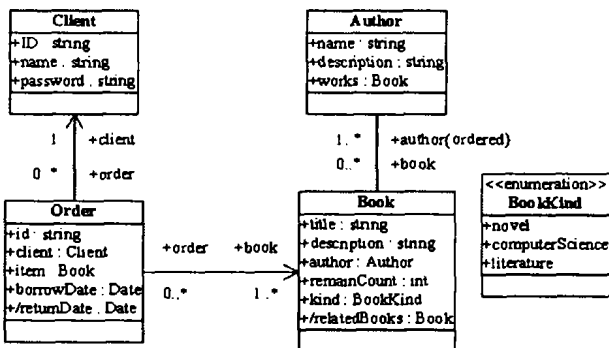


图3 图书馆管理系统类模型局部

定义过导航上下文就可以使用导航节点和它们彼此的导航关系来定义系统的导航结构。图书馆系

统的导航模型如图4所示,由主页出发可以导航到图书和作者的查询;而查询的结果或者是一个 Index 或者是一个节点;Book 通过 Index 来导航到作者信息;同样 Book 也通过 Index 来导航到相关作品。

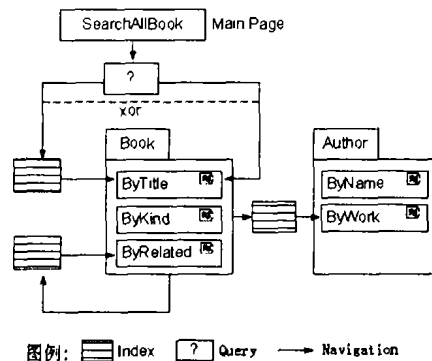


图4 图书馆管理系统的导航模型

概念设计和导航设计完成后,就可以展示设计了。针对导航节点 Book 将包含超链接列表 Authors 和 Related books 分别对应了导航模型中的 Index 导航访问方式。两对按钮 Next by kind 和 Prev by kind 与 Next by title 和 Prev by title 对应于导航上下文 ByTitle 和 ByKind 的默认的导航访问方式,即 Guided Tour 方式。

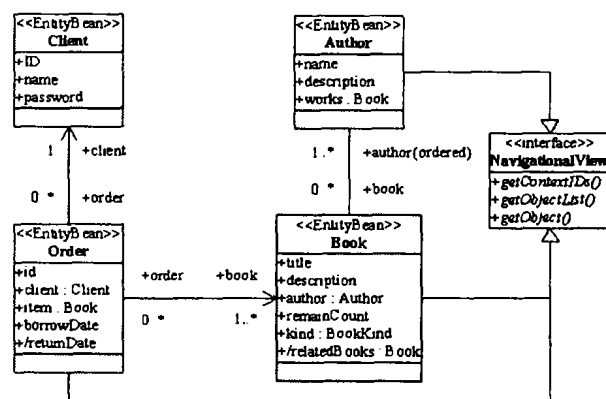


图5 图书馆管理系统的 EJB 模型局部

#### 3.2 PSMs

PIMs 定义完成后,将使用基于 MDHDM 的设计工具自动生成系统的 PSMs。图5是与图3对应的 EJB 模型部分。如图所示,由于 Book, Author 和 Order 都将参与导航,因此它们也要实现接口 NavigationalView,以供构造节点时使用。图6是图书馆管理系统的 Web 层模型的局部,由导航模型映射而成。为了在 Web 页面中生成导航节点的展示,每一个导航节点都拥有一个与之相对应的节点生成器(Node-Creator)。对于每个基本导航类,将有一个可以生成其必需数据的生成器类,如 BookNodeCreator。对于每个导航上下文也有继承自 ContextDecorator 的上下文生成器类,如 BookNodeCre-

atorByTitle 和 BookNodeCreatorByKind。由于上下文是可嵌套的,为了避免复杂的继承关系,我们在这里使用了 Decorator 模式。上下文生成器将调用导航上下文接口(NavigationalContext)生成上下文相应的内容和链接,最终由导航上下文对象通过 NavigationalView 接口来访问 EJBs 以取得必要的数

据。  
PSMs 自动生成以后,用户可以对对其进行必要的改进和精化,然后可以利用 PSMs 生成系统的实现代码。限于篇幅,本文中省略了示例最终的代码。

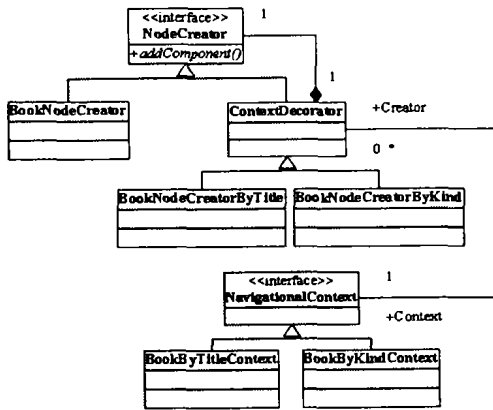


图6 图书馆管理系统的 Web 层模型局部

**结束语** 本文提出了一个基于 MDA 的 Web 应用程序开发方法 MDHDM。我们说明了应用 MDHDM 设计 Web 应用程序的三个步骤,并给出了与其相应的平台无关模型(PIMs)以及由 PIMs 映射而成的基于 J2EE 的平台相关模型(PSMs)。由于我

们将 MDA 和现有的 Web 应用程序开发方法相结合提出了 MDHDM,因此 MDHDM 在吸收了现有开发方法的优点的同时,利用 MDA 的特性也很好解决了它们所具有的缺陷。

目前 MDHDM 对于导航设计和展示设计中的行为建模的支持并不完善,这也是我们下一步工作的重点所在。同时,我们将进一步改进基于 MDHDM 的设计工具,使其更好地支持 Web 应用程序的设计。

### 参考文献

- 1 Miller J, Mukerji J. MDA Guide Version 1.0.1. Document number omg/2003-06-01. Retrieved from: <http://www.omg.org/mda>, 2003
- 2 Frankel D. Model Driven Architecture. Applying MDA to Enterprise Computing. John Wiley & Sons, 2003
- 3 Retschitzegger W, Schwinger W. Towards Modeling of Data Web Applications - A Requirement's Perspective. In: Proc. of the America's Conf. on Information Systems, 2000. 149~155
- 4 de Janeiro R, Plata L. A Software Architecture for Structuring Complex Web Applications. In: The 11th Intl. World Wide Web Conf. Hawaii, USA, 2002
- 5 Schwabe D, Rossi G. An Object Oriented Approach to Web-Based Application Design. Theory and Practice of Object Systems (TAPOS), Special Issue on the Internet, 1998, 4(4): 207~225
- 6 Koch N, Baumeister H, Mandel L. Extending UML to Model Navigation and Presentation in Web Applications. In Modeling Web Applications, Workshop of the UML'2000. Ed. Geri Winters and Jason Winters, York, England, Oct. 2000

(上接第152页)

```

Next k;
For l=1 to q
  If SD-r (ti, ul左) < ε or SD-r (ti, ul右) < ε Then
GDMA = GDMA ∪ ui;
  Next l;
Next i;
RDA = RDA - GDA; RMB = RMB - GMB; RDMA = RDMA - GDMA;
{DA 表出发的处理} GMB = φ; GDMA = φ;
For j=1 to m
  For k=1 to p
    If SD-r (wj, vk左) < ε or SD-r (wj, vk右) < ε
Then GMB = GMB ∪ vk;
    Next k;
    For l=1 to q
      If SD-r (wj, ul左) < ε or SD-r (wj, ul右) < ε
Then GDMA = GDMA ∪ ui;
      Next l;
    Next j;
RMB = RMB - GMB; RDMA = RDMA - GDMA;
{MB 表出发的处理} GDMA = φ;
For k=1 to p
  For l=1 to q
    If (SD-r (vk左, ul左) < ε or SD-r (vk右, ul右) < ε or
SD-r (vk左, ul右) < ε or SD-r (vk右, ul左) < ε)
Then GDMA = GDMA ∪ ui;

```

```

Next l;
Next k;
RDMA = RDMA - GDMA;
END.

```

*min-out* 算法的时间复杂度为  $O(mn^2)$ , 其中,  $m$  为关系中属性个数,  $n$  为关系中元组的数目。无论是表内冗余还是表间冗余, 消除冗余算法的复杂度较高, 需要系统大量的时间和空间, 因此要尽量减少使用的频率, 在确定会产生冗余的情况下才进行冗余信息处理。

### 参考文献

- 1 刘永山, 杨楠. 糊空值环境下的关系模型与关系操作. 计算机工程与科学, 1997, 19(1): 73~77
- 2 郝忠孝. 空值环境下数据库导论. 北京: 机械工业出版社, 1996. 1~76
- 3 何新贵. 模糊知识处理的理论与技术. 北京: 国防工业出版社, 1998. 14~86, 321~343