

文章编号:1001-9081(2010)02-0385-05

基于面向对象的 MDA 三级建模技术

黄建庭,张立臣,古金峰

(广东工业大学 计算机学院,广州 510006)

(jiant_huang@hotmail.com)

摘要:模型在模型驱动架构(MDA)软件开发生命周期中驱动整个开发过程,建模在MDA中起关键作用,建模是否成功是模型转换的前提,也决定MDA软件开发的成败。通过对当前建模技术的研究分析,根据对系统抽象级别的不同,提出基于MDA的三级建模技术,即计算无关模型(CIM)建模,平台无关模型(PIM)建模,平台相关模型(PSM)建模。详细介绍三个不同抽象级别模型的建模过程,并对它们之间的模型转换加以说明,从整体上描述MDA软件开发的过程,得出一种MDA软件开发方法。

关键词:面向对象;模型驱动架构;UML建模;模型转换

中图分类号: TP311.52 **文献标志码:** A

Object-oriented three-layer modeling in MDA

HUANG Jian-ting, ZHANG Li-chen, GU Jin-feng

(College of Computer, Guangdong University of Technology, Guangzhou Guangdong 510006, China)

Abstract: Model drives the entire development process in the MDA (Model Driven Architecture) software development life cycle. Modeling, being a prerequisite for the conversion model, takes a key role in MDA and determines the result of MDA software development. Three kinds of MDA-based modeling techniques were presented: CIM (Computational Independent Model) modeling, PIM (Platform-Independent Model) modeling and PSM (Platform-Specific Model) modeling, in view of the research and analysis of current modeling techniques and different levels of system abstraction. This paper described the three abstract level modeling processes in detail and the conversion among them, thus wholly giving description of the MDA software development process and ultimately obtaining a MDA software development methodology.

Key words: object-oriented; Model Driven Architecture (MDA); UML modeling; model transformation

0 引言

随着计算机应用到各个领域的广度、深度不断扩大,软件的需求变得越来越复杂,传统的软件开发方法已经不能满足软件需求的增长,逐渐呈现出软件开发的生产效率、可移植性、互操作性、维护与文档等一系列问题。针对传统软件开发方法中存在的问题,国际对象管理组织(OMG)于2002年提出一种全新的软件开发框架(Model Driven Architecture, MDA)。MDA的关键之处是,模型在整个软件开发过程中扮演了非常重要的角色,在MDA中软件开发过程是由对软件系统的建模行为驱动的^[1]。在MDA软件开发中,开发者只需要借助MDA建模工具对PIM进行建模,然后通过MDA工具自动转换生成PSM模型及代码,整个过程模型作为产品而非工件。

目前在MDA研究领域中,模型转换是热点研究问题,但是没有详细论述MDA过程中各种模型的建模过程。文献[2]介绍OMG规定的建模MOF元层次,简述元建模与元语言。文献[3]介绍各种模型转换技术及模型转换技术研究现状。文献[4]提出一种基于UML profile的建模语言。本文从全局的角度提出了一种对基于面向对象的MDA过程中各种模型建模的方法,并详细论述建模过程。即对面向对象的CIM建模使用UML用例图,它可以完整表述业务逻辑,也使

于与业务人员的沟通。PIM建模、PSM建模使用基于MOF扩展的UML,根据业务需要,适当地加入一些元素,成为新的特定领域建模语言。更重要的是,PIM、PSM的建模语言使用MOF表述,可以很容易定义两个建模语言之间的转换规则,这些规则同样可以应用到PIM到PSM之间的转换,从而简化模型间的转换。

1 MDA及建模技术

1.1 MDA概述

MDA是一种基于OMG发布的一系列标准:统一建模语言(Unified Modeling Language, UML),元对象设施(Meta Object Facility, MOF),XML元数据交换(XML Metadata interchange, XMI),公共数据仓库元模型(Common Warehouse Meta-model, CWM)的开放软件开发框架^[1]。其核心思想是建立与具体实现平台无关、能完全描述业务的平台无关模型(PIM),然后根据具体的目标平台定义一套转换规则,并借助辅助工具将PIM转换成与系统具体实现平台相关的平台相关模型(PSM),最后将PSM转换成代码。在MDA过程中有三个重要的模型:

1) 计算无关模型(Computational Independent Model, CIM)。CIM描述系统的业务知识和业务过程,不涉及任何软件系统的实现,构造CIM的语言一般是自然语言或者图表,

收稿日期:2009-08-06;**修回日期:**2009-09-16。 **基金项目:**国家自然科学基金重大研究计划项目(90818008);国家自然科学基金资助项目(60774095);广东省自然科学基金资助项目(07001774;04009465)。

作者简介:黄建庭(1984-),男,江西赣州人,硕士研究生,主要研究方向:分布式实时处理;张立臣(1962-),男,吉林长春人,教授,博士,主要研究方向:实时系统、并行处理、分布式处理、网络计算;古金峰(1982-),男,江西赣州人,硕士研究生,主要研究方向:嵌入式系统。

二者经常结合使用,包含了用来规定业务过程、参与者、相关部门、过程依赖等方面的术语,因此也被称为业务模型^[2]。

2) 平台无关模型(Platform Independent Model, PIM)。PIM也能完整描述系统业务,与CIM不同的是PIM涉及到系统的实现,它包含了对系统的所有规约,但是中立于任何实现系统实现平台。

3) 平台相关模型(Platform Specific Model, PSM)。PSM是与具体实现平台相关的平台相关模型,是最贴近代码级的模型,要求其构造语言必须足够精确,且具备一定的扩展性。目前比较流行的平台有CORBA、J2EE、.NET和实时操作系统(RTOS)等。

1.2 建模技术分类

模型在MDA软件开发过程中其非常重要的作用,文献[1]对模型的定义是:模型是以精确定义的语言对系统(或系统的一部分)作出的描述。而精确定义的语言是具有精确定义的形式(语法)和含义(语义)的语言,这样的语言适合计算机自动解释。模型总是以某种语言表述的,按描述语言不同对建模技术进行分类:

1) 自然语言建模。自然语言描述的模型具有精确的定义,能够很详细地描述模型的结构及其特性,比较符合人性的思维,容易被人所理解,但是用自然语言描述的模型不够直观,模型编写比较困难,很难被计算机解释,典型的自然语言描述的模型是需求文档。

2) 形式化语言建模。形式化语言建立在严密的数学逻辑基础之上,其建模能够精确地描述各种模型,但是模型的直观性不强,普通的建模人员或者程序员很难理解使用。

3) 统一建模语言(UML)建模。UML是构建系统模型尤其是软件系统模型的标准方法。它是一种形式语言有精炼、全面、伸缩性好、源自时间、公认建模标准等优点。它建立的模型非常直观,容易理解,其局限性就是不能适应特殊领域的建模,但是它提供了丰富的扩展机制:基于profile扩展和基于MOF扩展。

1.3 模型转换

模型转换通过抽象分别建立元模型和目标模型的元模型,对比得到的两个形式化模型的抽象组成元素及其语义,定义它们之间的映射规则^[5]。变换定义是由一系列的变换规则组成,这些变换规则是无歧义的规约,一条变换规则描述如何将源模型的一个元素变换成目标模型的一个或者多个元素。文献[6]提出了一种可复用并可对转换进行建模的基于模式的转换方法,并指明可以应用到EDOC到EJB的转换。

在MDA中的模型根据系统的抽象级别不同可以分为四层:计算无关模型(CIM)、平台无关模型(PIM)、平台相关模型(PSM)、代码模型。模型变换主要有以下几种:1) CIM到PIM的变换;2) PIM到PIM的变换;3) PIM到PSM的变换;4) PSM到PSM的变换;5) PSM到代码的变换。转换技术可分为:基于XMI的模型转换技术、基于图文法的转换技术、基于形式化的UML类图转换技术、基于MDA规范的模型转换技术^[3]。具体转换技术不再本文研究范围,不在赘述。

2 MDA的三级建模

综合以上对各种建模技术的分析,基于UML的建模技术最适合应用到软件开发的建模中。CIM使用UML的用例图来建模,以准确地表达业务知识和业务过程,方便业务人员及PIM建模人员理解。同一种元语言建立的模型可以很容易实

现模型间的转换,并保证模型转换的效率、准确性、可回溯性,因此PIM、PSM建模使用基于MOF扩展的UML。PIM到PSM基于MOF的模型转换如图1所示。

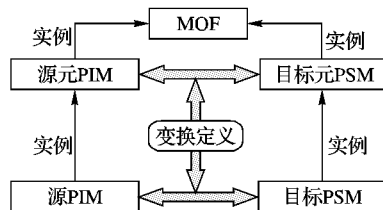


图1 PIM到PSM基于MOF的模型转换

2.1 CIM建模

计算无关模型的建模过程就是抽取固定的业务模式,通过形象的方式描述业务模式。用例用于表达系统所提供功能的片段,在用例图中用例是核心,用例图能准确捕捉系统的所有功能细节,并将其表达,它包括:行为者、用例、通信线、系统边界、用例描述、用例边界等元素。对CIM的建模还可以使用状态建模、交互建模、类建模、协作建模等,以及其他专门的建模语言。

CIM用例图建模过程如下:1) 获取一个需求,分析需求,识别出行为者、用例,并将潜在的行为者找出,精炼行为者,画出初始用例图。2) 编写出用例描述,将用例描述中遗漏的细节加入初始用例图中,确定用例之间的用例关系,完善初始用例图。3) 继续获取下一个需求,重复1),2)两个步骤直到所有需求分析完毕,再反复修改、精炼用例图使系统的模型更好、更准确。用例图建立完成后需与业务人员进行讨论,反复修改,以防止开发人员在业务理解上的差异造成对业务理解上的错误。

CIM用例图用于建立静态模型,结合CIM顺序图建立系统内部的交互模型,可以从静态、动态两个方面来完整表达系统。顺序图是由一组参与者组成的,用于描述交互发生的次序,时间在顺序图上是一项很重要的因素。

2.2 PIM建模

2.2.1 基于MOF的PIM元建模

面向对象的PIM元模型描述了进行面向对象建模所需要的域模型成分,为建立面向对象的PIM提供了规范的定义,其目的是在MDA开发框架下,首先建立平台无关模型,即建立面向对象的PIM。该元模型是对现有的面向对象模型的共同特征进行抽象,用于指导建立面向对象的PIM模型。

UML是一种面向对象的建模语言,其扩展方式有:UML profile和MOF,对比它们的特性,本文选择基于MOF扩展的UML,通过对UML扩展形成一种新的领域建模语言。基于MOF扩展的PIM元模型如图2所示。在新扩展的UML中,将EJBClass类分成三种不同的类:EJBDataClass、EJBKeyClass、EJBBeans,这三种类类型表示不同的领域类型。EJBDataClass用来表示数据持久化类,根据建立的数据结构对应地建立持久化类;EJBKeyClass用来表示关键字类,用来存储两个表之间的关系,其中包含主键及外键;EJBBeans用来表示Bean类,可以分为业务Bean和会话Bean,业务Bean用来描述一个业务功能,会话Bean表示用户与系统的交互。

2.2.2 PIM建模

PIM完整描述了系统的业务需求,但是又与具体实现的平台无关,它与PSM的差异取决于平台技术参考集合的规定,所以PIM与PSM之间是相对的,这里所指的平台是指:信息格式化技术、3GL和4GL及其扩展、分布式组件中间件、消

息处理中间件。

依照上述 PIM 元模型建立领域模型 PIM, 领域模型分为域类模型和域交互模型。域类模型主要描述系统的静态结构, 域交互模型主要描述业务规则的实现, 描述系统的动态交互特征。域类模型主要包括: 包 (Package)、类 (Class)、接口 (Interface)、属性 (Attribute)、方法 (Method) 及类之间的关联关系等。

域交互模型以平台无关的方式定义应用程序的事务、系统内部交互、系统与外部之间的交互、消息传递机制等。域交互模型能够自动生成会话组件,会话组件定义了应用系统的行为。域交互模型最主要的元素包含了属性、操作和视图。典型的域交互有系统的非功能特性,为了实现业务记录,可以定义一个记载日志的域交互 `LogService`,它包含 `openFile (String fileName)` 域操作,用于打开日志文件。

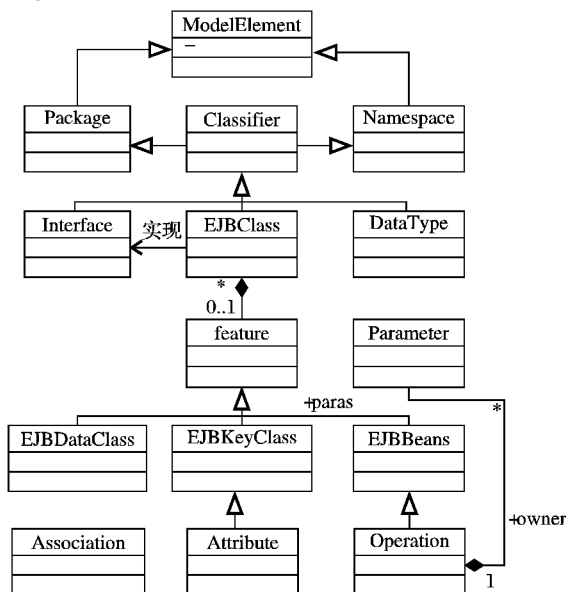


图2 基于 MOF 扩展的 PIM 元模型

2.3 PSM 建模

2.3.1 基于 MOF 的 PSM 元建模

普通的 UML 并不能表述基于特定平台技术的 PSM,为了建立模型驱动的 PSM 并进行相应变换的研究,需要进一步增加描述 PSM 的规范,通过基于 MOF 扩展 UML 方式来对 PSM 进行建模。

J2EE 是当前分布式企业计算的热门中间件平台, 基于 J2EE 平台软件开发采用 MVC 架构与 MDA 在某种程度上具有一定的相似性, 选择基于 J2EE 平台描述 MDA 软件开发过程具有一定的普遍性。基于 J2EE 的 PSM 有三个平台相关的模型: R-PSM (关系模型)、EJB-PSM (EJB 模型)、UI-PSM (用户界面模型)。R-PSM 的元模型如图 3 所示, R-PSM 元模型包含 EJJBDataClass、EJBKeyClass 两个类类型元素; EJB-PSM 的元模型如图 4 所示, EJB-PSM 对 UML 作深层次的扩展, 增加许多元素, 类元素包括 EJJBDataClass、EJBKeyClass、EJBBeans, EJBBeans 一般化为 SessionBean 和 EntityBean, 而 EntityBean 一般化为 EJBEntityComponent, 属性一般化为 EJBServingAttribute; UI-PSM 的元模型如图 5 所示, UI-PSM 新增 UIClass 类元素, 类元素一般化为 UIDataClass 和 UIBeans, UIBeans 同样一般化为会话 bean 和实体 bean, 实体 bean 一般化为 UIComponent, 用户组件 UIComponent 一般化为 UIFORM、UIGraphics、UIGrid 等页面控件元素, 关联 Association 一般化

为 ComponentUsage、UIServingAttribute。

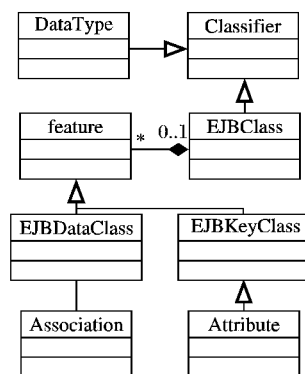


图 3 R-PSM 的元模型

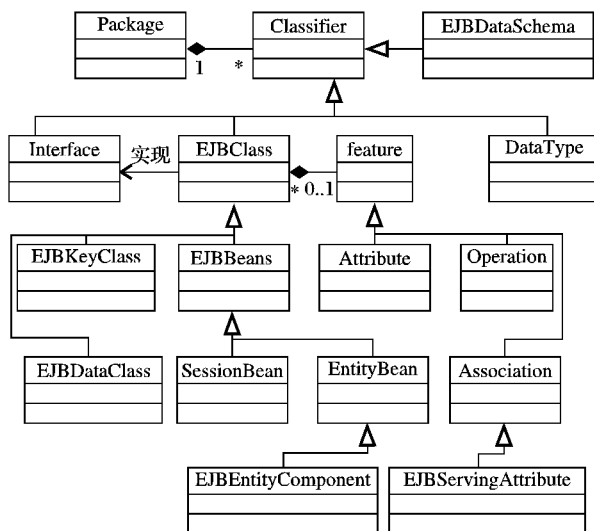


图 4 EJB-PSM 的元模型

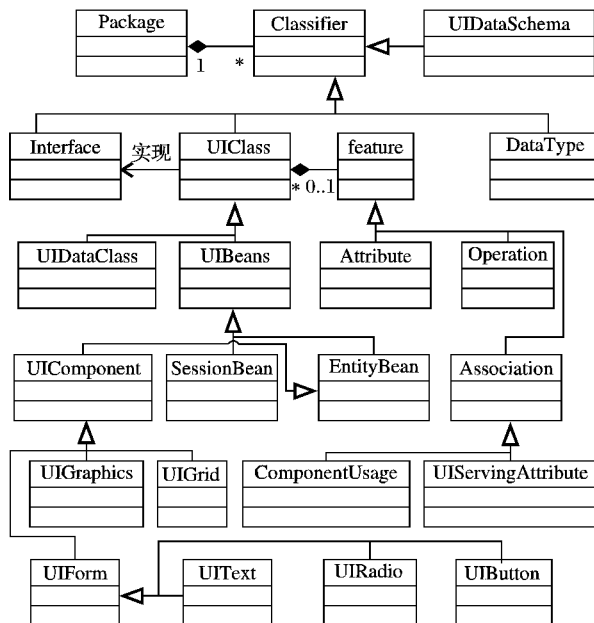


图5 UI-PSM 的元模型

2.3.2 PSM 建模

根据上述 PSM 元模型,可以建立三个相应的平台相关 PSM:R-PSM、EJB-PSM、UI-PSM。数据持久性是 J2EE 开发的一个重要部分,比较常用的数据库是关系数据库。R-PSM 展示了应用程序数据持久性关系,为每一个 PIM 类建立一张表,所有属性都是表中的字段。当属性类型不是数据类型而是类时,表中的字段应包含未见,外键指向表示类的表。对于

PIM 中每个关联类建立一个单独的表,关联关系分为:一对一、一对多、多对多。一对一关联转换时分别在对应的两个表中加入关联字段;一对多关联转换时可以将关联转换为关联类,在关联类中增加相应的两个类的主键;多对多关联转换时将多对多的关联关系转换为一对多的关联关系,然后再按照一对多的关系进行处理。

EJB-PSM 是处理业务、事务、安全、持久性的中间层模型。可以简单地为每一个生成一个组件来建立 EJB 的组件模型,但是在远程访问分布式环境中,组件交互频率非常高,加重了系统的负担,所以需要控制对组件的粒度,达到一次远程调用就完成与对象所有属性的交互效果,可以选择粗粒度的 EJB 模型。对于所有非其他 PIM 类的组合成分的 PIM 类,将其变换为一个 EJB 组件和一个数据模式;每一个 PIM 类生成一个主键类、数据类;每个 PIM 关联类转换为 EJB 关联,并归入一个数据模式;每个 PIM 属性都转换为 EJB 数据类的 EJB 属性;每个 PIM 操作都转换为 EJB 组件的操作;PIM 关联类转换为两个 EJB 关联和一个 EJB 数据类。

UI-PSM 规定了一些列为用户提供 HTML 内容的 UI 组件的定义,展示了应用系统最终 UI 界面要实现的各功能模块及其之间的关系。UI 界面实现与用户的交互,是系统与用户交互的接口,事件驱动的源头,UI 界面中的每一组件元素都会触发系统与用户之间的交互。UI-PSM 的主要元素有:UI 组件的容器、UI 数据模式、UI 功能组件、UI 非功能性组件(验证组件、安全控制组件)。为了系统的安全,可以定义一个系统安全控制组件,要登录用户必须使用系统指定的数字证书,并验证数字证书的合法性。

3 基于 MDA 的三级建模技术应用

J2EE 是一套全然不同于传统应用开发的企业级应用框架,包含许多组件,主要可简化且规范应用系统的开发与部署,进而提高可移植性、安全与再用价值。J2EE 开发的系统具有很好的可伸缩性、灵活性、易维护性等优点,是目前企业级分布式解决方案的首选。

下面通过开发基于 J2EE 平台的网上商城系统(OLSS)来说明 MDA 的三级建模技术。网上商城系统提供销售各种生活百货、日用品,为吸引顾客,每天推出特价商品组合。用户通过浏览器进入站点,选择自己喜欢的东西然后将它加入购物车,最后登录并进入结账中心结账,后台处理订单,安排送货。

3.1 OLSS-CIM 建模

计算无关模型的建模过程就是抽取固定的业务模式,通过形象的方式描述业务模式。用例用于表达系统所提供功能的片段,在用例图中用例是核心,用例图能准确捕捉系统的所有功能细节,并将其表达,它包括:行为者、用例、通信线、系统边界、用例描述、用例边界等元素。对 CIM 的建模还可以使用状态建模、交互建模、类建模、协作建模等,以及其他专门的建模语言。限于篇幅,本文只给出用例图模型,通过分析系统的需求分离出创建账户、验证信息、记录失败信息、创建订单、处理订单、反馈成功信息等主要六个用例,以及分离出用户、认证系统、管理人员等主要三个行为者,网上商城系统的 CIM 用例模型如图 6 所示。

3.2 OLSS-PIM 建模

因为 CIM 建模是抽取固定业务模式对系统的业务进行建模,模型的多样性、复杂性、与系统无关性决定了 CIM 很难自动转换为 PIM,但是 CIM 和 PIM 的建模语言都使用统一建

模语言或者是专业的建模语言,机器可读性很强,且容易被理解、无二义性,手工从 CIM 到 PIM 的模型转换就变得非常容易。

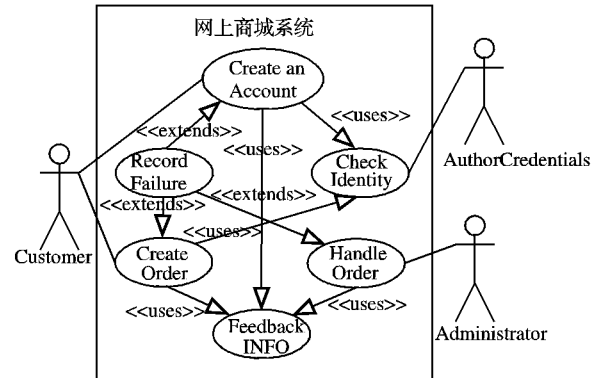


图6 网上商城系统的 CIM 用例模型

根据 OLSS-CIM 建立 OLSS-PIM。在 OLSS-PIM 中,每种特价商品组合都包含各种商品,这表明在货物中包含这些商品的数量属性。每一单货物的总价格根据客户所选的特价商品组合和其他非特价商品的价格及数量来决定。订单上的价格是客户所选所有的商品价格打折后的总和,再根据客户选择的送货方式及送货范围收取一定的运输费用,另外,如果客户有代金券或者优惠券,可以根据券值的大小在计算总费用时减掉等额的费用,最后得出用户应支付的费用。网上商城系统的 OLSS-PIM 如图 7 所示。

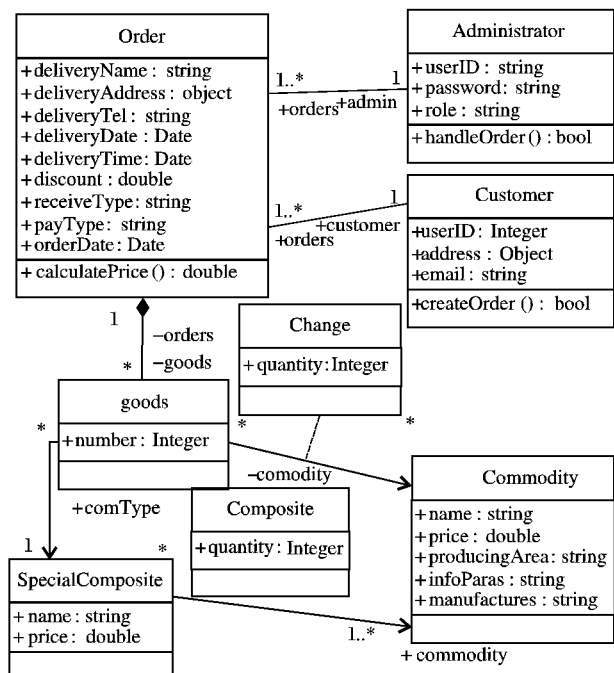


图7 网上商城系统的 OLSS-PIM

3.3 OLSS-PSM 建模

本文所讨论的是用 UML 描述系统的 PIM 转换到基于 J2EE 平台上的 PSM,由于 PIM 和 PSM 的元模型都是基于 MOF 的,所以 PIM 元模型到 PSM 元模型的变换可以通过 MOF 规定它们的变换定义,再定义元模型的变换规则,这些变换规则实现 PIM 元模型到 PSM 元模型的变换。同样,用这个变换定义生成 PIM 到 PSM 模型之间的变换规则,以实现模型转换的自动化。当前研究人员提出的一些方法:将 UML 模型视为图,利用形式化的图文法(graph grammars)进行转换,

这是一类基于图的转换方法^[8];基于ATL(Atlas Transformation Language)^[9]的模型转换方法;从EDOC profile到CORBA profile映射的ODAC^[10]方法等。目前MDA工具有很多,开源的MDA工具包括:MOFScript、IBM的Moel Transormation Framework(MTF)框架、ATL引擎、MTL引擎、Eclipse的GMT插件、Kent Modeling Framework(KMF)、AndroMDA、Middlegen等^[3]。商业化的MDA工具包括:ArcStyler^[7]、MCC(ModelComponent Compiler)模型组件编译器、Codagen Architect、OptimalJ、Model-in-Action等^[3],其中OptimalJ是针对J2EE平台的MDA开发工具。

根据PIM以及PSM的建模规则,可以将OLSS-PIM转换为OLSS-PSM,基于J2EE的PSM有三个平台相关模型:R-PSM(关系模型)、EJB-PSM(EJB模型)、UI-PSM(用户界面模型)。R-PSM的建模首先需要对数据类型进行转换:字符串转换为CHAR(40),整数变换为INTEGER,日期变换为DATE,对于只包含属性的对象类型把数据类型内联到当前的表中。其次,所有类建立一个单独的表,类属性是表中的列。最后,一对一的关联分别在关联两端的类中增加对面关联端的关键字作为外键,一对多、多对多转换为关联建立一个单独的表,并将关联两端类的主键作为列加入表中。

EJB-PSM的建模规则如下:对于所有非其他PIM类的组合成分的PIM类,将其变换为一个EJB组件和一个数据模式;每一个PIM类生成一个主键类、数据类;每个PIM关联变换为EJB关联,并归入一个数据模式;每个PIM属性都变换为EJB数据类的EJB属性;每个PIM操作都变换为EJB组件的操作;PIM关联类变换为两个EJB关联和一个EJB数据类。由于其他组件模型图相似,限于篇幅,本文只给出EJB-PSM的部分组件模型图,网上商城系统部分EJB-PSM组件模型如图8所示。

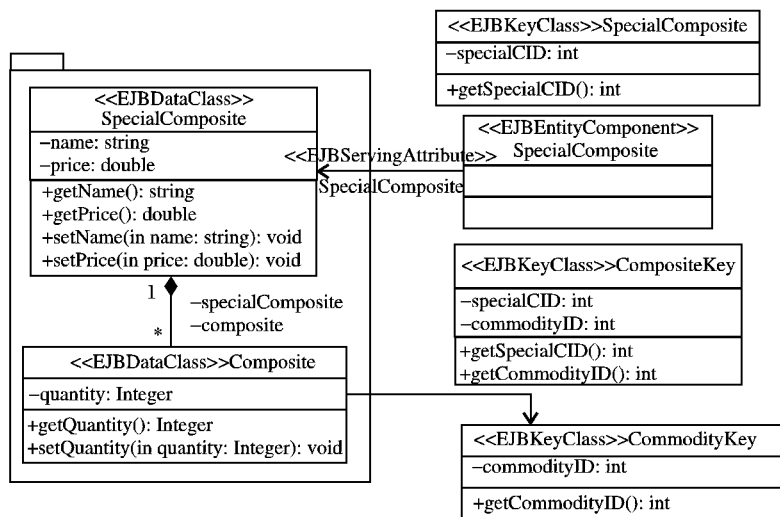


图8 网上商城系统的部分EJB-PSM组件模型

同样,UI-PSM的变换规则与EJB-PSM的变换规则差不多,不同的是图形用户界面中的每个<<UIComponent>>:组件展示服务器返回的信息要通过<<ComponnetUsage>>与<<EJBEntityComponent>>组件进行通信,所以存在组件之间、PSM间的交互,在UI-PSM模型中<<UIComponent>>与<<EJBEntityComponent>>之间存在着依赖关系。限于篇幅,不再给出UI-PSM组件模型图。

PSM到代码的转换可以借助MDA工具来完成,目前比较先进的且支持J2EE平台的MDA工具是由Compuware公司开发的OptimalJ,它可以支持模型到代码的自动转换,还可以让程序员对

部分程序进行添加、修改等操作,并支持代码与模型间的同步功能,使模型在以后的维护过程中起到文档的作用。

4 结语

MDA是OMG针对传统软件开发方法中出现的生产效率、可移植性、互操作性、维护与文档等一系列问题提出的一种新的软件开发框架,它很好地解决了传统软件开发中存在的问题。不过MDA还处在初期阶段,要大规模地应用到工业生产中,还需要较长的一段时间以及一些标准的指定。目前已经出现很多支持MDA的工具,有关MDA的标准也在不断的制定完善中。

模型在MDA中占核心地位,建模的成败决定MDA软件开发的成败。本文根据MDA软件开发的思路,对MDA整个生命周期中出现的不同模型,按照不同的系统抽象级别,提出三级建模技术,并详细介绍了三种不同抽象级别的模型建模过程及建模方法,最后通过例子来说明此方法的可行性,旨在找到一个通用的MDA软件建模方法及整个软件开发方法。接下来的工作是:对MDA建模技术的进一步研究、细化,完善MDA的三级建模技术方法;结合面向对象的MDA,研究面向方面的MDA软件开发方法。

参考文献:

- [1] KLEPPE A, WARMER J, BAST W. MDA explained: The model driven architecture: Practice and promise[M]. [S. l.]: Addison-Wesley, 2003.
- [2] 江春. MDA方法与基于UML的MDA建模[J]. 沈阳工程学院学报:自然科学版, 2008, 4(1): 67-69.
- [3] 张德芬, 李师贤, 古思山. MDA中的模型转换技术综述[J]. 计算机科学, 2006, 33(10): 228-230.
- [4] CAPLAT G, SOURROUILLE J L. Model mapping in MDA[C]// Fifth International Conference on the Unified Modeling Language, LNCS 2460. Berlin: Springer, 2002.
- [5] DUDDY K, GERBER A, LAWLEY M, et al. Model transformation: A declarative, reusable patterns approach[C]// Proceedings Seventh IEEE International, Enterprise Distributed Object Computing Conference. Washington, DC: IEEE Computer Society, 2003: 174-185.
- [6] HUBERT R. ArcStyler: The Architectural IDE for MDA[EB/OL]. [2009-08-01]. <http://www.iosoftware.com>.
- [7] AGRAWAL A. Graph rewriting and transformation(GREAT): A solution for the model integrated computing (MIC) Bottleneck [C]// ASE'03: 18th IEEE International Conference on Automated Software Engineering. Los Alamitos: IEEE Computer Society, 2003: 364-368.
- [8] BEZIVIN J, HAMMOUDI S, LOPES D, et al. Applying MDA approach for Web service platForm[C]// Proceedings of the 8th International, Enterprise Distributed Object Computing Conference. Washington, DC: IEEE Computer Society, 2004: 58-70.
- [9] GERVAIS M P. Towards an MDA-oriented methodology[C]// Proceedings of the 26th Annual International, Computer Software and Applications Conference. Washington, DC: IEEE Computer Society, 2002: 265-270.
- [10] 张德芬, 李师贤. 面向方面的MDA开发方法[J]. 计算机应用与软件, 2008, 25(1): 73-75.