

利用UML建模设计EJB应用系统

陶 隼, 赵文耘, 杨 俊

(复旦大学计算机系软件工程实验室, 上海200433)

摘 要: 从EJB构件模型出发, 分析了EJB应用系统建模的特殊性。以开发一个网上书店的EJB系统为例, 结合EJB构件模型特点, 使用统一建模语言(UML)来建模和设计, 展示了UML在开发EJB应用系统中的运用。

关键词: 统一建模语言; 构件模型; 企业级JavaBean

Modeling and Designing EJB Application System with UML

TAO Jun, ZHAO Wenyun, YANG Jun

(Software Engineering Lab, Computer Science Department, Fudan University, Shanghai200433)

【Abstract】 In this article, the characteristics of modeling an EJB application system are analyzed on the basis of the EJB component. With the example of online bookstore system, along with the particularity of EJB component model, how UML is used to model and design this system is shown, hence the UML application in EJB system is illustrated.

【Key words】 Unified modeling language; Component model; Enterprise JavaBeans (EJB)

UML是面向对象分析与设计(OOA&OOD)方法发展的产物。它统一了Booch, Rumbaugh和Jacobson提出的OO方法的概念、符号表示及模型, 成为大众所接受的面向对象标准建模语言^[2]。随着基于构件软件工程(CBSE)技术的快速发展, 构件市场上出现了多种构件模型^[1]。EJB构件模型作为其中的一种构件模型, 提供了将Java对象封装成EJB构件的方式。EJB构件技术基于面向对象的技术, 向对象施加构件模型的约束, 所以用UML对EJB系统建模和一般的面向对象系统又有不同之处。

1 EJB应用系统

构件模型是构件类型、接口以及构件类型之间的允许的交互模式的集合^[5]。目前构件市场上存在的构件模型主要有微软的COM+, OMG的CORBA和Sun公司的EJB。

Sun公司的EJB规范对EJB服务器、EJB容器和EJB构件等核心内容进行了定义^[5], 见图1。

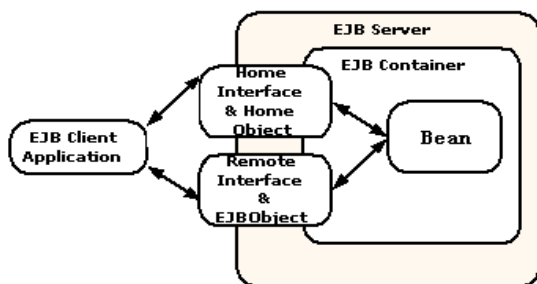


图1 EJB构件模型

EJB服务器是管理EJB容器的高端进程或应用程序, 它提供事务处理、持久化和安全服务的访问。EJB容器是一个管理一个或多个EJB构件的抽象, 它通过规范中定义的接口使EJB构件获得所需的服务。EJB构件是计算和数据存储的单元。

EJB规范提供了一个服务器端的构件环境, 它提供了一个多层的基于Web的企业应用透明层。因此开发者可以集中于应用逻辑的设计和环境的配置, 而不必关心安全、事务和

持久化的实现, 这将大大加快企业应用的开发周期^[5]。

2 EJB系统及其UML视图特点

EJB构件模型要求对其系统中的对象施加约束, 因此不同于一般的面向对象系统^[4], 主要表现在以下几个方面。

(1) 分布式对象计算环境下特定的对象访问和创建服务, 在EJB环境下是系统JNDI服务和HomeObject提供的远程对象定位与创建服务。

(2) 代理设计模式(Design Pattern)的频繁运用, 在EJB环境下反复出现的设计模式是代理模式(Proxy Pattern), 分布式对象环境下的代理模式可以实现事务控制等机制, 具体的实现方式决定于具体的EJB容器。

(3) EJB环境下具有EntityBean和SessionBean两种构件封装形式。

(4) EJB系统是服务器端系统, 这确定了最终系统的物理部署的形式。

EJB构件模型的特点决定了UML对EJB系统建模时各UML图的对应特点。

(1) 类图中的每一个EJB构件类都有对应的HomeObject类和RemoteObject类, 它们实现了扩展EJBHome和EJBObject接口的接口。

(2) 在类图中出现了代理模式的表示, 其动态特征在时序图和协作图中体现。

(3) 所有EJB构件是特定的Java类的实例, 在类图中它们实现SessionBean接口或EntityBean接口。

(4) 部署图中系统的物理分布是客户机-服务器形式。

3 UML建模实例分析

系统实例: 一个B2C网上书店系统。图书信息和用户账户信息利用持久化技术保存在服务器端, 顾客在客户端选购图书, 由购物车(BookCart)负责跟踪记录网上顾客的行为, 并且同步更新图书信息和用户账户信息。整个系统设计成一个EJB应用系统, 利用UML的多种视图完成系统从设计到实

作者简介: 陶 隼(1976~), 男, 硕士生, 主要研究方向为面向对象、构件技术; 赵文耘, 教授; 杨 俊, 硕士生

收稿日期: 2001-05-24

现和部署的建模。

3.1 用例视图

右面的UML用例图(图2)对这个网上书店系统作了用例分析,图中的系统用户有3个角色:购书客户、库存信息管理员和账户管理员。库存信息管理员和账户管理员负责图书和账户信息的管理。用例图捕获了系统的需求,而且驱动网上书店系统的设计。

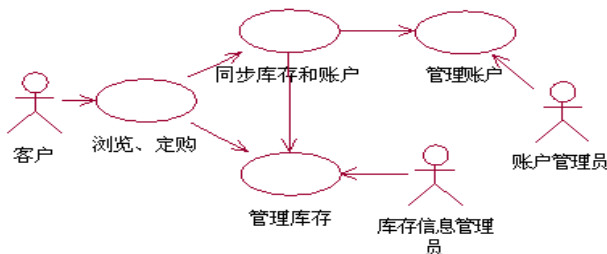


图2 用例图

3.2 逻辑视图

3.2.1 类图

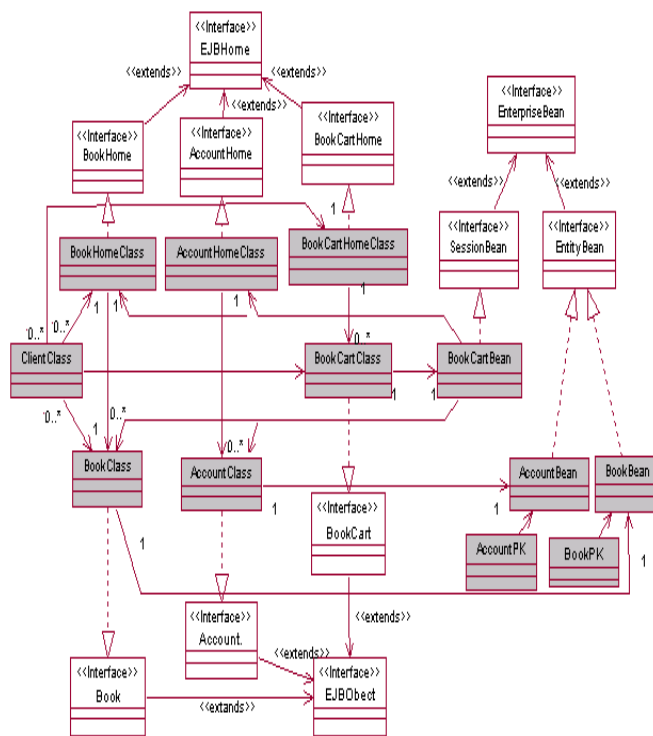


图3 类图

在网上书店的UML类图中(图3),系统中的应用概念被建模成类,这些类描述了一系列拥有信息和相互通信以实现行为的离散对象。对象所拥有的信息建模成属性,对象执行的行为建模成操作。对象—对象之间的关系被建模成关联。图中的类可以分为4种类型。

- (1)客户端类,图中的ClientClass。
- (2)HomeObject类,图中的BookHomeClass,AccountHome Class, BookCartHomeClass。
- (3)RemoteObject类,图中的BookClass,AccountClass,BookCartClass。
- (4) Enterprise Java Bean 类,图中的 BookBean,AccountBean, BookCartBean。

其中,客户端类通过远程方法调用定位远程对象和调用

远程对象方法,HomeObject类负责定位或创建RemoteObject类和Enterprise Java Bean类,是这两个类的Factory类。HomeObject类和这类是1:N的关系。RemoteObject类是Enterprise Java Bean类的代理类,两者之间是严格的1:1关系。在类图中出RemoteObject类和Enterprise Java Bean类构成Proxy设式^[6]。在UML的类图中Proxy模式表示如图4。

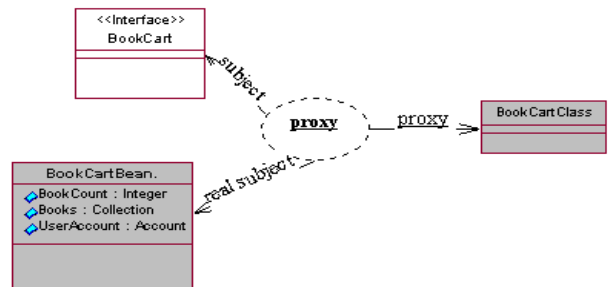


图4 proxy模式的类图表示

3.2.2 时序图

时序图表示了系统中随时间安排的一系列消息,时序图的一个用途是显示用例图中用例的行为序列。下面的时序图显示了网上书店系统中“同步库存和帐户”这个用例的时序图,从这个时序图中也能反映出代理模式的运用,见图5。

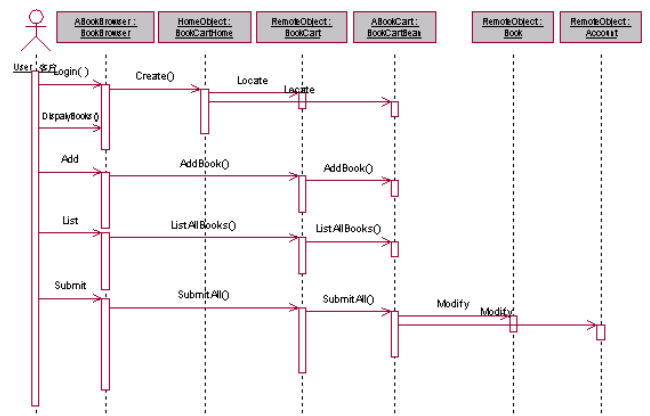


图5 时序图

3.2.3 状态图

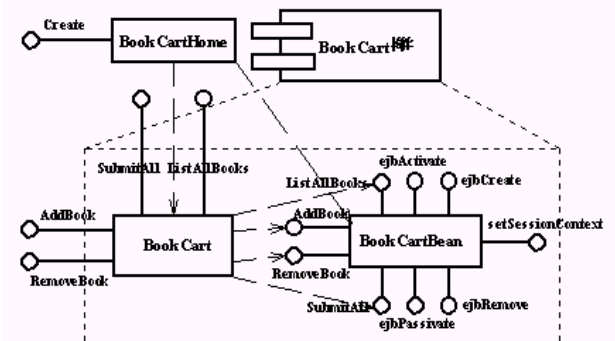


图6 状态图

状态图对系统中类的对象的可能生命历史建模。与一般的面向对象系统不同的是:在EJB构件模型下对象有了新的状态。对于一个EJB构件对象,有两种基本的状态:Activated状态和Passivated状态,这两种状态之间的切换由

EJB容器控制,同时EJB框架提供了ejbPassivate()和ejbActivated()两个hook方法^[5], 见图6。

3.3 部件视图

UML模型对逻辑分析和物理实现均可以表达,物理视图提供了将类映射至部件的机会。UML规范对部件的定义是^[3]:与一系列接口一致和为其提供实现的、物理可替换的系统组成部分。在网上书店系统实现阶段的部件图是系统物理实现的表示。以BookCartBean对象为例,它对于用户来说是不可见的,所有客户调用的方法都通过EJBObject对象代理。所以,EJBObject对象和BookCartBean对象之间有一个紧密耦合的关系。因此在本文中将实现BookCart接口的Java类和BookCartBean类封装成为BookCart部件,BookCart部件是在UML概念下的实现层次上的部件,见图7。

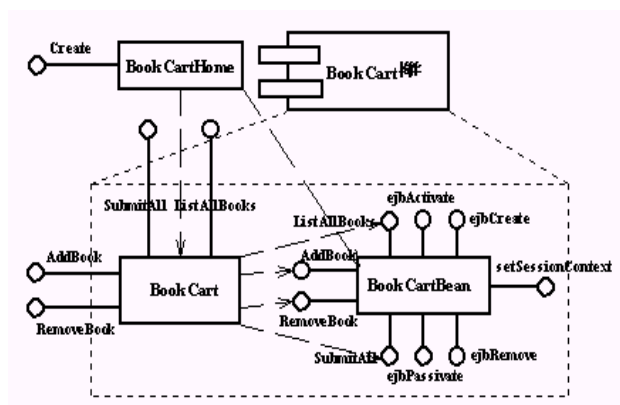


图7 BookCart部件

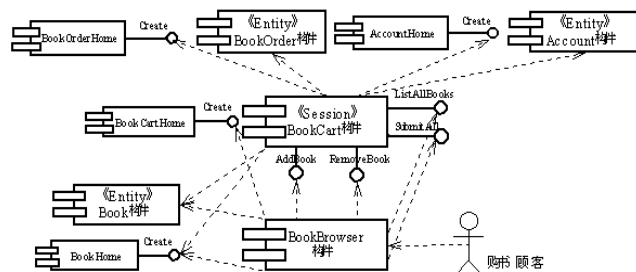


图8 系统部件图

(上接第131页)

显然,如果对SA_r的认证能够提前,则不需要增加IKE SA建立过程中消息交换步数。可利用HASH_I将SA_r在第5步送给响应者认证,如果通过则继续送消息6,否则终止该ISAKMP SA的建立。修改后的HASH_I和HASH_R如下:

$$\text{HASH}_I = \text{PRF}(\text{SKEYID}, g^i | g^j | \text{CKY}_I | \text{CKY}_R | \text{SA}_i | \text{SA}_r | \text{ID}_{ii} | \text{ID}_i)$$

$$\text{HASH}_R = \text{PRF}(\text{SKEYID}, g^i | g^j | \text{CKY}_R | \text{CKY}_I | \text{SA}_r | \text{ID}_{rr} | \text{ID}_r)$$

虽然这样会稍微增加计算HASH_I的工作量,但却提高了安全性。与原始的HASH_I和HASH_R相比,在HASH_I中增加了SA_r,在HASH_R中用SA_r替代了原来的SA_i。

本文在讨论IKE的基础之上,分析了IKE SA建立过程中

图8是整个网上书店系统的部件图,图中的每个部件实现支持一些接口并使用其它部件的接口。部件之间的依赖由接口来确定,系统的部件图描述了整个系统中部件的组织 and 它们依赖关系。EJB系统的运行还依靠构件运行环境:EJB容器的底层支持,在整个EJB构架中,框架支持设计成对用户透明,因此在下面的部件图中没有包含这部分信息。

3.4 部署视图

EJB系统中大部分的计算任务和系统数据都以EJB构件形式存在于服务器端^[5]。EJB系统的部署图是客户机/服务器方式的部署图。

4 结论

本文围绕开发EJB应用系统的整个过程,利用统一建模语言建立模型,描述了一个网上书店系统从需求捕获,系统设计,到物理实现和系统部署的一系列完整的视图。EJB应用系统基于EJB构件模型,在EJB构件模型中的EJB构件是加以构件模型规范约束的Java对象,这确定了EJB应用系统使用UML建立系统模型的可能性,也产生了EJB应用系统UML建模的特殊性。本文的网上书店建模实例中就包含了在EJB特定构件模型下系统模型的“特殊性”,这些“特殊性”包括了代理设计模式、对象状态、构件实现和物理部署,它们分别体现在文中的各个UML视图中。

参考文献

- 1 Kozaczynski W, Booch G. Component-based Software Engineering. IEEE Software, 1998-09/10
- 2 Rumbaugh J, Jacobson I, Booch G. The Unified Modeling Language Reference Manual, 1999
- 3 Joao Pedro Sousa, Garlan D. Formal Modeling of the Enterprise JavaBeans Component Integration Framework. CMU/SEI, 1999-03
- 4 Bachman F, Bass L. Enterprise JavaBeans Specification (Volume II): Technical Concept of Component-based Software Engineering. Sun Microsystems, CMU/SEI, 2000-05
- 5 Haefel R M. Enterprise JavaBeans. O'Reilly, 1999-08
- 6 Kobryn C. Modeling Components and Frameworks with UML. Communications of ACM, 2000, 43(10)

可能出现的一种入侵,在整个IKE的过程中,其它的安全问题,这些还有待于进一步的研究。

参考文献

- 1 Doraswamy N, Harkins D. IPSec: the New Security Standard for the Internet, intranets, and virtual private networks. Trentice Hall TTPR 2000-01
- 2 Zhou J. Fixing A Security Flaw in IKE Protocols. Electronics Letters 1999.35 (13):1072-1073.
- 3 程朝辉. 一种基于数字签名的IKE协议的实现与分析. 计算机科学, 2001, 28(3):43-45
- 4 Dayan Y, Bitan S. IKE Base Mode. Draft-ietf-ipsec-ike-base-mode-02.txt. 2000