

基于 EJB 的图书馆读者信息管理系统研究与实现

Research and Implement of Library user's Information Management System Based on EJB

王秀义 (山西工业职业技术学院 037003)

摘要:本文介绍了如何利用基于 CMP 的实体 Bean 来开发一个图书馆读者信息管理系统,并且阐述了构架和部署一个实体 Bean 的具体方法和步骤。

关键词:EJB 企业 Bean 实体 Bean CMP

1 引言

J2EE 是一种构架标准,代表了最新的应用程序构架发展状况,全面支持二层和三层应用程序,以及基于 Web 的应用程序和 Web 服务应用程序。而 EJB 构架是 J2EE 平台的核心,EJB 构架规范了用 Java 语言开发和部署服务器端应用程序的业务逻辑。EJB 应用程序按企业 Bean 这种组件的形式组织业务逻辑,每个企业 Bean 都封装了应用程序的部分业务逻辑。EJB 构架使得企业应用程序更容易开发,能够将 EJB 应用程序部署到任何与 J2EE 兼容的服务器上,可以重用每个企业 Bean,简化了构建复杂的企业应用程序的过程,实现了业务逻辑与表示逻辑的分离,容易开发 Web 服务,并且客户组织可以充分选择与 J2EE 兼容的服务器,也方便了应用程序的管理,与客户现有的应用程序和数据集成。

2 EJB 构架概述

2.1 EJB 应用程序

企业 Bean 是组件,充当分布式企业应用程序的一部分。每个企业 Bean 都封装了应用程序的部分业务逻辑。企业 Bean 一般与资源管理器、其他企业 Bean 和其他企业应用程序进行通信。

运行时,企业 Bean 存在于一个 EJB 容器中。EJB 容器提供了企业 Bean 的部署和运行时环境,EJB 容器提供下列服务:远程访问、安全性、事务以及并发性和实例生命周期管理。

一个企业应用程序可以包含一个或者多个企业

Bean。如果一个应用程序包含多个企业 Bean,则可以把它们部署在企业网络上的一个或者多个容器中。

客户通过企业 Bean 的客户视图 API 访问企业 Bean。企业 Bean 可以提供远程客户视图 API、本地客户视图 API 或者 Web 服务客户视图 API。远程客户视图 API 包括 Bean 的远程组件和远程宿主接口,不论客户与企业 Bean 所使用的 Java 虚拟机 (JVM) 是否相同,都可以用相同的 API 调用企业 Bean,如果 JVM 在不同的容器中,则客户通过网络与企业 Bean 通信以执行调用。本地客户视图 API 包括本地组件接口和本地宿主接口,允许容器使用一个优化的内部路径执行调用,从而省去了远程调用的开销,而且比远程客户视图 API 有更多的功能。Web 服务客户视图 API 能够以 Web 服务的形式向其他 Java 或者非 Java 客户提供功能,允许企业 Bean 的方法作为 Web 服务操作提供给客户,并且可以用基于 XML 的 Web 服务协议进行访问。

2.2 企业 Bean 类型

EJB 构架定义了 3 种企业 Bean 类型:会话 Bean、实体 Bean 和消息驱动 Bean。会话 Bean 主要用来处理客户端复杂的业务逻辑,以减轻客户端的负担。实体 Bean 主要用来对数据库中的数据进行操作和维护。消息驱动 Bean 主要用来将 EJB 应用程序与面向消息的企业应用程序进行集成。

会话 Bean 是短暂的对象,它定义了以客户会话期间,通过方法的调用来掌握用户的信息。根据方法请求时是否维持客户状态,会话 Bean 又可分为无状态会

话 Bean 和状态会话 Bean。当客户终止与有状态的会话 Bean 的交互操作时,会话就会终止,并且这个 Bean 也不再拥有状态值。无状态的会话 Bean 并不掌握其客户的信息或状态,只是用户能够调用 Bean 的方法来完成一些具体操作。无状态的会话 Bean 只是在方法调用的时候,才会获得用户的参数变量;而当方法调用完成以后,它并不会继续保持这些变量。

实体 Bean 是持久的对象,它是用对象的形式来展现数据库中的数据记录,并且利用对象所定义的业务方法来执行一系列的数据表/记录存取工作。由于实体 Bean 对应于数据库中的记录,所以数据库记录的任

何改变也应该被同步到组件池中相关的 Bean 中,该特性称为实体 Bean 的持久性。实体 Bean 的持久性可以被实体 Bean 自己管理 (BMP, Bean - Managed Persistence),也可以让 EJB 容器管理 (CMP, Container - Managed Persistence)。前者由开发人员在实体 Bean 中提供数据存取代码,后者由 EJB 容器自动处理数据存取的调用。

实体 Bean 与状态会话 Bean 在生命周期和编程方面有着很大的差别。一个状态会话 Bean 只能被一个客户使用,但是一个实体对象可以在多个客户间共享。状态会话 Bean 的对象状态通常在主存中维护,而实体 Bean 的对象状态通常在一个数据库中进行维护。会话对象的状态可以与一个事务同步,但是不可还原,实体对象的状态通常随事务而变化,并且是可还原的。不能保证会话对象在故障下还能存活,并重新启动它的容器,发生故障后,客户持有的对会话对象的引用不再有效。实体对象可以在故障下存活,并重新启动它的容器,容器重启后,客户可以继续使用实体对象的引用。

消息驱动 Bean 是异步调用的企业 Bean。它没有客户视图 API,客户通过把消息发送给消息驱动 Bean 而访问它们。与会话 Bean 或者实体 Bean 不同,消息驱动 Bean 没有业务方法。消息驱动 Bean 定义了一个消息监听方法,EJB 容器调用它来传递消息。此外,消息驱动 Bean 也不保存消息监听方法调用间的任何状态。它的生命周期非常类似于无状态会话 Bean。

2.3 企业 Bean 结构

每个企业 Bean 由企业 Bean 类、企业 Bean 客户视图 API 和部署描述符组成。企业 Bean 类是一个 Ja-

va 类,实现业务方法和企业 Bean 对象的生命周期方法。企业 Bean 客户视图 API 由企业 Bean 宿主接口、企业 Bean 组件接口和企业 Bean Web 服务终点接口组成。企业 Bean 宿主接口定义 create 和 remove 方法 (控制企业 Bean 对象的生命周期),以及 find 方法和宿主方法。企业 Bean 组件接口和 Web 服务接口定义业务方法,客户可以在各企业 Bean 对象上调用这些方法。部署描述符是一个 XML 文档,包含与企业 Bean 有关的声明性信息。

3 开发和部署 EJB

网上图书馆读者信息管理系统主要实现以下功能:读者注册、读者登录、更改读者信息和删除读者信息。

我们利用基于 CMP 的实体 Bean 来开发上述系统,并且利用 Sun J2EE Application Server 作为 EJB 的发布平台,利用 Cloudscape Database 作为后台数据库来存取一定的数据。实现方法与步骤如下:

(1) 定义宿主接口 ReaderHome

[ReaderHome.java 代码]

```
package library. ejb;
```

```
import java. rmi. RemoteException;
```

```
import javax. ejb. CreateException;
```

```
import javax. ejb. FinderException;
```

```
import java. util. Collection;
```

```
//这是实体 Bean 的 Home 接口
```

```
public interface ReaderHome extends EJBHome
```

```
{
```

```
//建立两个返回远程接口对象的抽象方法 create()
```

```
//这个方法与 Bean 类 ReaderBean 中的 ejbCreate() 和 ejbPostCreate() 相对应
```

```
//create() 方法用于建立一条新的用户数据记录
```

```
public Reader create (String cardid)
```

```
throws DuplicateKeyException, CreateException, RemoteException;
```

```
public Reader create (String readname, String cardid, String password, String address, String email) throws DuplicateKeyException, CreateEx-
```

```
ception, RemoteException;
//建立一个返回 Reader 远程接口对象的抽象
方法 findByPrimaryKey()
//这个方法用于搜索一条或多条用户数据记录
public Reader findByPrimaryKey(String cardid)
throws ObjectNotFoundException, FinderException,
RemoteException;
public Collection findAll() throws FinderException,
RemoteException;
}
```

(2) 定义组件接口 Reader

[Reader.java 代码]

```
package library.ejb;
import util.*;
import java.rmi.RemoteException;
import javax.ejb.EJBObject;
//这是实体 Bean 的 Remote 接口
public interface Reader extends EJBObject
{
//建立 3 个 business 方法: setReaderInfo(),
setCardId(), setPassword()
//这 3 个 business 方法必须与 Bean 类 Reader-
Bean 中的 business 方法相对应
//这 3 个方法用来获取用户提交的数据。
public void setReaderInfo(String readname,
String cardid, String password, String address,
String email) throws RemoteException;
public void setReadName(String readname)
throws RemoteException;
.....
//建立 3 个 business 方法: getReaderInfo(),
getCardId(), getPassword()
//这 3 个 business 方法必须与 Bean 类 Reader-
Bean 中的 business 方法相对应
//这 3 个方法返回服务器端的数据。
public Hashtable getReaderInfo() throws Re-
moteException;
```

```
public String getReadName() throws RemoteEx-
ception;
.....
}
```

(3) 定义 Bean 类 ReaderBean

Bean 类 ReaderBean 定义了宿主接口 ReaderHome 中定义的生命周期方法的实现和组件接口 Reader 中定义的业务方法的实现。主要包括:定义了 5 个数据表字段变量 (readname、cardid、password、address、email), 这些变量的数量和类型与后台数据库中相关数据表中字段完全对应;声明了 9 个回调方法;声明了 12 个 business 方法。代码略。

4 结语

Java 的 EJB 技术的主要思想就是让“恰当的专家做恰当的事情”,应用领域的开发人员将开发精力放在应用逻辑方面,而不用考虑底层的计算技术;而计算机专业开发人员去处理底层的计算技术细节,而不用考虑应用领域的专业知识。

EJB 计算技术代表的是一种前沿技术,它使得复杂的多层结构应用系统开发变得容易。EJB 技术建立在已取得极大成功的 Java 语言之上,采用 EJB 技术能有效地进行软件复用,提高开发人员的效率,降低软件的开发和维护成本,提高软件的质量,控制所构建系统的复杂性。EJB 技术将使得 Java 在企业计算中的地位得到加强,为基于 Java 的应用系统提供了一个框架,和目前的许多系统和模型相比,EJB 具有许多优越性,种种迹象表明,EJB 有可能成为多层结构分布式应用系统的服务器端构件模型的首要选择。

参考文献

- 1 Vlada Matena 等著,施平安等译,EJB 应用指南(第 2 版),清华大学出版社,2004。
- 2 刘特编著,J2EE EJB 应用编程实例,清华大学出版社,2003。
- 3 <http://www.cn-java.com>