Build Distributed n-tier Life Insurance System Based on J2EE
基于 J2EE 构建多层分布式人寿险系统

马 军 （西安卫星测控中心软件室 710043） 周艳梅 （中国太平洋人寿公司西安分公司 710001）

摘要：本文描述了基于 J2EE 的多层分布式系统的体系结构，构建多层分布式人寿险系统的设计思想、开发过程，以及采用的主要技术。系统应用 UML、ROSE 进行建模，以 RUP 来开发。

关键词：J2EE EJB ROSE

1 引言

随着 IT 技术以及 Internet 的迅速发展，分布计算已在 Internet/Intranet 上得到广泛应用。传统的 Client/ Server 结构更明显表现出局限性，正逐渐被 Browser/Server 结构所取代。基于 Browser/Server 结构，以组件开发、利用 Browser 作为最终用户界面的解决方案已成为今后企业应用软件开发的方向，一种满足这种需要的新企业解决方案的体系结构便应运而生，这便是 J2EE。保险业在我国正处在快速发展阶段，潜力巨大，而且人寿保险的保险期限往往跨越人的整个生命期，在长达几十年内，逐年缴纳续期保费，所以信息量就象滚雪球一样急剧增长，要对如此海量数据进行长期的科学管理，必须采用安全、高效、可伸缩、可移植、便于维护的多层分布式企业级应用系统，基于 J2EE 构建的多层分布式人寿险系统正是在这一前提下进行研究开发的。

本文描述了 J2EE 多层分布式系统的体系结构，在开发分布式人寿险系统时采用的 J2EE 技术和设计思想，以及整个系统的开发过程。

2 J2EE 的多层分布式系统的体系结构

J2EE 是由 SUN 推出的利用 Java2 平台来开发、部署和管理多级企业解决方案的体系结构，它为企业应用程序提供了各种不同的分布式中间件服务，如名字、安全、事务、消息和数据库等。

2.1 构建多层分布式系统采用的 J2EE 的技术

J2EE 的技术主要包括企业级 JavaBean (EJB)、Java命名和目录接口 (JNDI)、远程方法调用 (RMI/RMI-IIOP) 、JSP 、Java Servlets 、Java 消息服务 (JMS)、JDBC 、Java 事务处理 (JTA)、Java 事务处理服务 (JTS) Java Interface Definition Language (IDL)、Java 邮件服务 (Java Mail)等，以下描述了开发本系统时开发的 JSP、Servlet、EJB 以及应用到相关 J2EE 技术。

2.1.1 Enterprise JavaBeans API(EJB)

EJB 是 J2EE 中最为重要和核心的部分，是服务器端的组件，使得开发跨平台的、基于组件的企业应用程序非常容易，它通过提供对中间层服务的自动支持，如事务、安全、数据库连接等，降低了开发中间层服务的复杂性，EJB2.0 规范定义了 3 种基本的 bean 类型：* 会话 Bean，代表商务过程，包括有状态会话 Bean (Stateful session bean) 和无状态会话 Bean (Stateless session bean)，它们在服务器发生故障时无法生存，生命期相对较短。状态会话 Bean 提供了与客户端的会话交互，可以存储状态从而代表一个客户，每一个实例只用于一个单一的线程，无状态会话 Bean 提供某种单一 的服务，不维持任何状态，无状态会话 Bean 执行效率比较高，在开发会话 Bean 时，尽可能使用无状态会话 Bean。

• 实体 Bean，代表永久性的商务数据，通常用实体 Bean 代表数据库中的数据，每个实体 Bean 都有主键，是实体 Bean 实例的标识符，实体 Bean 在服务器故障发生后能继续存在，其管理模式分为容器管理模式 (CMP) 和自管理模式 (BMP)，CMP 会减少代码量和错误，但 BMP 的好处是比较灵活。

• 消息驱动 Bean，用来异步处理 JMS 的消息，是一个 JMS 消息的聆听者，间接地从它所在的容器中得到消息。

2.1.2 Java Servlet

Servlet 是用来扩展和加强 Web 服务器的网络组件，基于请求/响应 (Request/Respense) 机制，不像专有的服务器扩展机制 (如 Netscape Server API)，Servlet 具有服务器和平台无关性，Servlet 不但可以访问所有 Java API，包括通过 JDBC API 访问企业数据库，而且也可以访问具体的 HTTP 调用库。

2.1.3 JavaServer Pages (JSP)

JSP 技术提供了一个简单的、快捷的方法来创建动态网页内容，使得能够快速的进行与平台无关的基于 Web 的应用程序的开发。JSP 和 Servlet 相似，其脚本编译为 Servlet。JSP 引擎将它和它所在的 HTML 文件一起合成 Servlet 的代码，然后它的执行就和 Servlet 的一样了，先编译成 .class 文件，由支持 Java 虚拟机的服务器来

进行处理，然后生成 Web 页面返回给浏览器。

2.1.4 命名和目录服务接口

名字和目录服务在 Internet 和 Intranet 中扮演这重要角色，JNDI 定义了如何标识组件和相关资源在网络中的位置，目录结构的每个节点称为 context，每个名字都是相对 context 的，不存在绝对名字的概念，它提供一个从 Java 平台访问商业信息的统一的、工业标准化的无缝连接，在不同协议下访问不同目录的 API。

2.1.5 JDBC（Java Database Connectivity）

JDBC API 用 Java 编程语言以一种统一的方式来对各种各样的数据库进行存取，使用 JDBC，可以很容易地将 SQL 语句发送到任何一个虚拟的关系型数据库，JDBC 定义了 JDBC-ODBC 桥、JDBC-Native Driver 桥、JDBC-Network 桥、纯 Java 驱动程序等 4 种不同的驱动程序。

2.1.6 Java 事务处理（JTA）和 Java 事务处理服务（JTS）

JTA 定义了可以存取各种事务监控的 API。JTS 是 CORBA OTS 事务监控的基本实现，JTA 和 JTS 规范为组件提供可靠的事务处理支持。

2.1.7 XML

XML 是一种可以用来定义其他标记语言的语言，也是一种元标示语言，被用来在不同的商务活动中共享数据，它和 Java 的发展是相互独立的，一些 J2EE 技术需要依靠 XML 描述它们的具体内容，如 EJB 的部署描述文件是基于 XML 的，通过 XML 描述 EJB 组件的部署属性。

2.2 J2EE 的框架和系统构架

J2EE 使用多层分布式应用模型，应用逻辑由组件来实现，一个典型的 J2EE 应用是由四个层次组成，即客户层、Web 层、应用业务层和企业信息系统层（EIS）。如图 1 所示，各个应用组件根据它们的所在层以及业务需求分布在不同的机器上。

客户层用来同用户交互，调用 Web 层或业务层的组件响应用户的请求。可以基于 Web 也可以不基于 Web，可以是纯 HTML 文挡、Applet，甚至是 Java 应用程序或 CORBA 对象。

Web 层产生表示逻辑，Web 层组件在技术上主要采用 Java Servlet 以及 JSP，功能上实现动态页面。它们通过 HTTP 响应客户的请求；调用应用服务器中的 EJB 进行业务逻辑处理，并将结果返回。

业务层包括解决商务问题的组件，构成了应用业务的规则，是整个应用的核心。通常由 EJB 容器内的 EJB 组件来实现，EJB 容器提供了分布式组件需要的生命周期、持久性、事务、命名服务等服务。

EIS 层为企业信息系统服务，负责为应用提供数据来源，提供对数据的管理、读写和存储。包括数据库系统、事务处理系统和遗留系统，是 J2EE 应用和非 J2EE 应用的连接点。

每层组件都有为其提供服务的容器，分别是客户端容器、Web 容器、EJB 容器和 EIS 软件。

其中 Web 层和业务层为 J2EE 应用的中间层。

采用这种模式开发的系统具有易维护、扩展性和可重用性强，运营效率、安全性、开发效率高等优点。

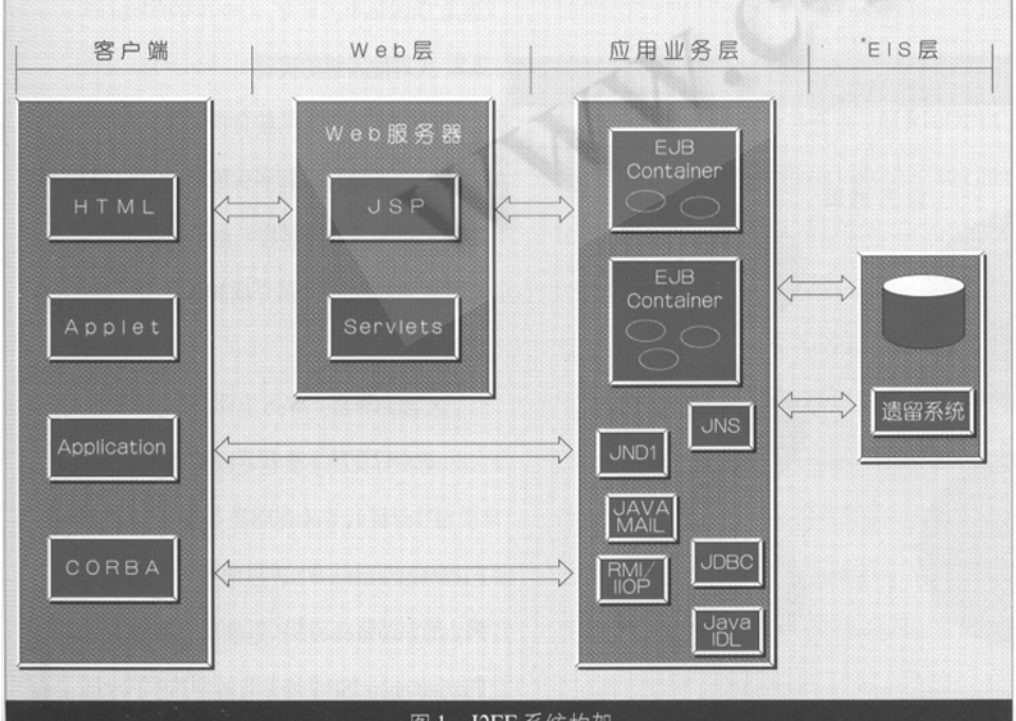

图 1 J2EE 系统构架

3 基于 J2EE 人寿险系统设计与实现

人寿险业务系统负责处理从保户与保险公司签订投保合同起到保险责任终止这一漫长过程中的所有业务处理过程，其中包括：承保核保（首期缴费、投保单录入、核保和出单）、缴费管理、理赔管理、给付管理、借款/还款管理、银行代收/付、客户管理、产品管理、用户管理、系统维护等诸多方面的内容。在开发该系统时采用统一过程模型，该模型是由 Rational 公司提出，主要用来描述使用 UML 开发的过程。该过程具有用例驱动、构架为中心、迭代开发与增量开发相结合的特点。

在开发该系统时以 UML 和 Rational Rose 进行需求分析和系统设计，建立 Use Case View、Logical View 和 Component View。在初始阶段，用 Rose 建立用例模型；在细化阶段用 Rose 建立

类视图来描述类以及他们之间的相互关系；在构造阶段用Rose建立组件模型，反映系统组件间的相关性，同时可以正向和逆向处理；在移交阶段，Rose主要完成模型的更新。

3.1 人寿险系统的设计过程

本系统依据系统需求，Use Case View以及Logical View，运用J2EE构架，将整个系统分成四层，即客户层、Web层、业务层（EJB层）和数据层。如图2所示，层次的划分直接影响到系统的性能和扩展性等多个方面。其中客户层（Client Tier）在本系统中是Web浏览器，Web层(Web Tier)由JSP和Servlets组成，形成所有的交互界面，采用MVC结构,JSP对应View,负责显示问题,Servlet对应Controller，在JSP中尽量少嵌入Java代码，在该层不包括任何业务逻辑。业务层（EJB层）是整个系统最关键的部分，主要处理系统的业务逻辑，如承保核保、缴费管理、理赔管理、给付管理等业务流程以及用户管理、日志、系统运行的配置参数等，包括实体Bean和会话Bean,如图3所示，采用Session facade设计模式,即利用Session Bean封装Entity Bean，来负责调用Entity Bean的方法，客户端只允许与Session Bean交互,这样可以缩短系统响应时间，减少资源利用；在设计会话Bean时，采用粗粒度企业Bean，把一组相关的用例映射为会话Bean，如设计首期缴费会话Bean，封装所有用于首期缴费有关的实体Bean，并且映射了与首期缴费有关的全部用例Session Bean用来处理业务逻辑和工作流，是客户端工作的抽象，如设计理赔处理会话Bean封装了完成理赔处理工作流全过程，单一的服务设计为无状态会话Bean，如设计加密/解密无状态会话Bean，完成系统内敏感信息的加密/解密。实体EJB的设计，将所有的数据都封装到实体EJB中,在设计时采用粗粒度和细粒度相结合的方式，对分布要求不高的采用本地模型、细粒度，即用Entity JavaBean代表数据库中的数据，数据库中的表对应Entity Bean中的类，表中的字段对应类中的属性；分布要求高的采用远程模型，复合实体、粗粒度。所有对Entity Bean的操作都代表了底层数据库中数据的变动，若一次操作对数据库中表的一个记录发生关联就用CMP EntityBean来实现；若一次操作对数据库中表的多个记录发生关联就用BMP EntityBean来实现，BMP EntityBean包含与数据访问相关的代码,以及开发动态查询功能；数据库连接采用JDBC-ODBC来访问数据库；数据库服务器选用Oracle 8i。

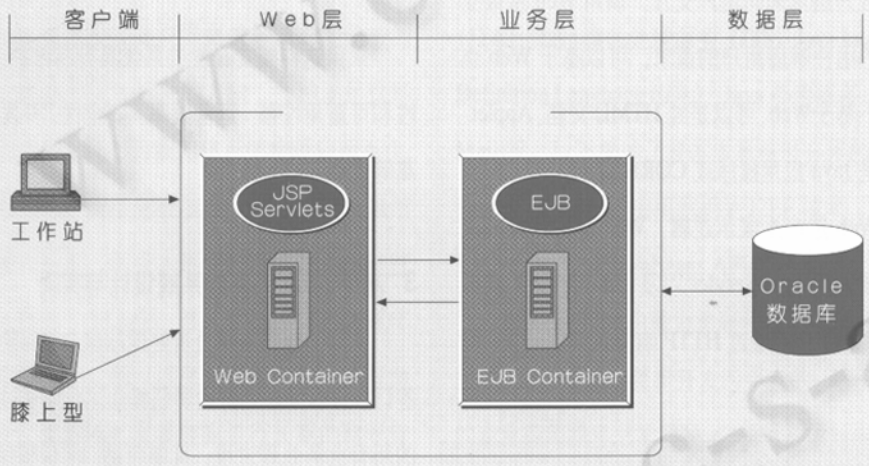

图2 J2EE人寿险系统构架

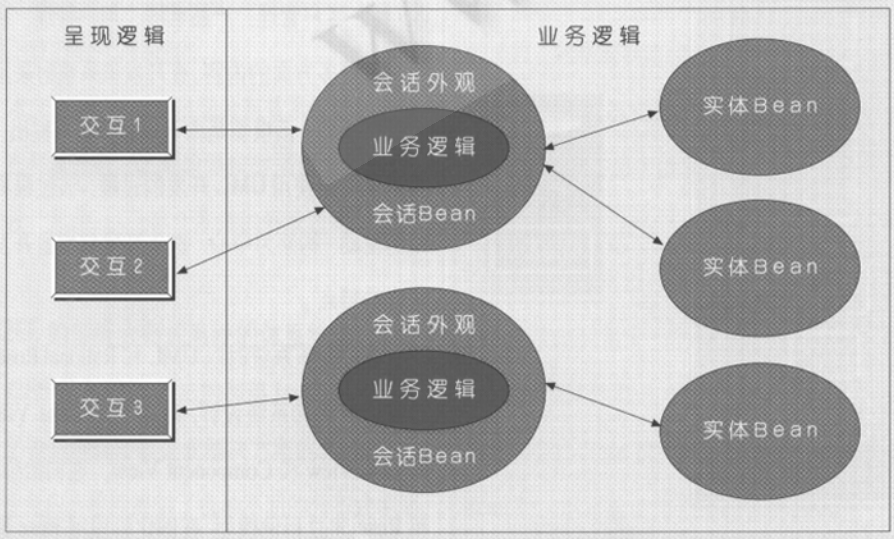

图3 会话Bean、实体Bean交互示意图

3.2 人寿险系统的实现

如上所述，本系统负责处理从保户与保险公司签订保险合同起到保险终止这一漫长过程中的所有业务处理过程，由于篇幅关系,以承保核保的首期缴费为例介绍其具体实现步骤和关键技术。

- 用户界面、Web组件的开发

用户界面的主要包括以下JSP和Servlets

Firstincome.JSP是首期缴费JSP，Firstincomea.JSP是增加首期缴费JSP，Firstincomea.JSP是删除首期缴费JSP，Firstincomeu.JSP是修改暂收费JSP、Firstincoemf.

JSP是查找暂收费JSP，Firstincomeo.JSP为打印暂收费JSP，Sefirstincome.java是首期缴费的Servlet。

表单间信息的交互通过JSP的内建对象来实现，用会话对象来维持表单的会话状态。

• 用JNDI定位EJB组件

采用JNDI来定位EJB组件，JNDI可使应用程序能够用统一的方式访问多种名字和目录服务，在JNDI中一个目录结构中的每个节点称为context,每个名字有一个context对应。首先为JNDI初始化取得系统属性，Properties props = System.getProperties()，接着生成初始的上下文，Context cxt = new IntialContext(propos),通过初始的上下文Context，就可以经过目录树定位到需要的EJB组件，SbfirstincomeHome sbfirstincomehome = (SbfirstincomeHome)ctx.lookup("sbfirstincome")，从而就可以使用EJB中的方法了。

• EJB的设计

在本系统中的EJB包括实体Bean和会话Bean。要控制EJB的粒度大小，接口尽量简单。实体Bean的设计，Ebtemincomelongbean,主键是,EbtemincomelongPK,用来封装暂收费长险数据的一个实体Bean,Ebtemincomeshortbean,主键是，EbtemincomeshortPK,用来封装暂收费短险数据的一个实体Bean。

会话Bean的设计，Sbfirstincome,为有状态会话Bean，封装了首期缴费有关的用例，如增加、删除、修改、查找和打印，客户通过Sbfirstincome调用Ebtemincomelongbean和Ebtemincomeshortbean中的方法。Sbfirstincomel,为无状态会话Bean，包括首期缴费有关保费计算等方法。

• EJB与数据库的连接

用JDBC-ODBC桥进行EJB同数据库连接，简单的说JDBC可做三件事，与数据库建立连接，发送SQL语句，处理结果。一般经历如下几步：建立数据源，配置数据源如配置同ODBC；装入JDBC驱动程序，Class.forName("sun.jdbc.odbc.jdbcOdbcDriver")；建立连接，Connection con= DriverManager.getConnection(url,"用户","口令")；执行SQL语句，createStatement创建Statement对象，用于发送简单的SQL语句。PrepareStatement由PrepareStatement创建PrepareStatement,用于发送带有输入参数的SQL语句。pstmt = con.perpareStatment("Sql语句")；查询结果，查询结果保存在RestultSet中；释放数据库的连接，con.close。

4 结束语

本系统是用Rational Rose 2001 Enterprise Edition进行系统建模，用Jbuider6.0 Enterprise Edition开发，应用服务器是Borland Appserver 4.5，数据库服务器是Oracle 8i。本文从实际应用角度描述了基于J2EE的企业应用的开发过程。随着各行各业竞争的日益激烈,将企业自己的应用建立在J2EE平台之上，将给企业带来无穷的益处，从而将促进企业进一步发展，有着广阔的前景。

参考文献

1 王常力，廖道文，集散型控制系统的设计与应用，清华大学出版社，1993。

2 曹茂叶，500KV变电站自动化若干问题的探讨，电网技术，1998，22(8)。

3 周丽，35KV变电站综合自动化系统的研究，继电器，1998，26(2)。

4 周沿东，基于Intranet的新一代电力企业管理信息系统，电力系统自动化，1999，23(9)。