

J2EE Multi-Layer Architecture Technique in the Research and Application of Collaborative Office System

Shijie Wu

Abstract In this paper, J2EE architecture is the main study object. Detailed analysis and elaboration are made based on the example of collaborative office system from the analysis of J2EE technology architecture. And the application advantages are demonstrated in the practical application of easy maintenance, scalable, safe, reliable and advanced features.

Keywords: J2EE technology architecture; Cooperative office; Application analysis

J2EE 多层架构技术在协同办公系统中的 研究与应用

吴士杰

天津海运职业学院, 天津 300350

摘要: 本文以 J2EE 技术架构为研究对象,从 J2EE 技术架构分析入手,选取协同办公系统为例,对 J2EE 技术架构做出了较为详细的分析与阐述,并以此论证了 J2EE 技术架构在实际应用中所表现出的易维护、可扩展、安全可靠以及先进实用等特性在内的多方面应用优势。

关键词: J2EE 技术架构; 协同办公; 运用分析

引言

随着网络技术与软件开发技术的快速发展,多层次的体系架构模式已经成为软件系统体系结构的首选,该体系架构能够有效设计、开发、部署、运行和管理系统。在多层架构的开发中,针对中间层的应用服务器的开发设计非常重要。

目前开发平台的选择中,主要有微软 .NET 平台和基于 J2EE 平台的两种解决方案。这两种方案主要针对分布式应用上为用户提供了整体的规范和指导,主要体现在分布式应用的开发、设计、合成、性能、可靠性、安全性等方面。作为竞争对手的两个开发平台,在体系架构和设计目标方面都有相似之处,但是在开发实现上又各持己见。根据对软件开发公司的平台案例进行优化比较,将 J2EE 和 .NET 两个平台面向不同项目分别比对。通过对比表 1 的结果,可以清楚的看到 J2EE 平台在各个项目中的优势所在。

表 1 J2EE 平台和 .NET 平台项目对比表

项目内容	J2EE 平台	.NET 平台
支持 Web 组件技术的性能	良好	良好
移植性	良好	一般
扩容性	很好	良好
再利用性	很好	良好
服务发布、实现	良好	很好
软件研发商	很好	良好
中小企业使用	良好	良好
大型企业使用	很好	一般

1. J2EE 技术架构分析

J2EE 技术架构是为了精简用户管理方案的开发、部署和处理相关的复杂问题所运用的 Java 2 平台。它保留了标准版中的诸多优势，同时对 JSP、EJB、XML 提供了全面的技术支持，这样大幅缩短了系统开发的时长。

为了极大的降低多层应用复杂的开发过程和高额的开发成本，J2EE 技术结构不仅提高了安全机制和运行的性能，以及良好的部署应用性。而且，将集成框架的开发平台进行了统一，这样大大满足了开发平台对可靠性和扩展性的需求。

1.1 J2EE 组件技术

J2EE 的编程理念就是基于组件技术，利用这些组件，在 J2EE 平台上可以得到所期望的服务。将数据和方法进行简单的封装被称之为组件，组件具备自身特有的方式和属性，方式是组件一些简单的功能，而属性是组件数据的访问者。

以下特性为组件技术所必备条件：

- 组件并不是源代码或函数库，而是一个单独编译的程序。
- 在不需要访问组件源代码的前提下，组件必须能够集成在一起，以便创建一个更大的程序，或者将使用该组件的程序与组件的代码进行链接。
- 组件技术必须具备支撑手段、事项、属性，以及支撑某种共同性质的通信途径。
- 在设计组件技术时，应当能够设置组件的属性值，并且支持持久性属性。并且，组件在激活时应当能够将属性的初始值作为设置值。
- 通过查询组件应当能够确定它所支持的属性、事件和方法，针对自身接口的描述是组件所应具备的。
- 组件技术应当适应各种各样的环境，并且能够正常的进行使用。

下面具体介绍一下 J2EE 技术架构中几个重要的组件技术。

1.1.1 JSP 组件技术

JSP (JavaServer Pages) 是将 Java 源文件和 JSP 标签插入到网页文件中，由此生成 JSP 文件，它是建立在服务器端 Java 应用程序技术规范之上的一种动态网页技术，这些 Java 程序可以重新定向网页及操作数据库等。客户端只需要通过 Web 浏览器就可以运行这种动态网页得到相应的结果，而程序操作都是在服务器端执行完成的。由于这种动态网页可以进行跨平台的操作，所以更便于系统研发人员进行开发设计。

1.1.2 JSF 组件技术

JCP (Java Community Process) 制定的一个针对 Web 应用框架的标准，及 JSF。丰富的组件层次结构和良好的请求处理生命周期是 JSF 所具有的，能结合服务器端的 Java 应用程序与 JSP 同步使用。并且，在缩减互联网应用开发与维护周期的同时，将内容丰富、功能强大的用户界面提供给互联网应用。利用 JSF 提供的可扩展、可利用、用户的界面框架基于组件，通过快速开发工具 RAD 的支持，可以通过拖拽组件的方式对 Web 用户界面进行可视化编辑，将用户界面上的组件与一个数据源进行捆绑，并且服务器端处理由客户端产生的事件，基于 Java 的 Web 用户界面的开发难度被大大的降低，提高了开发效率。

1.1.3 EJB 组件技术

EJB (Enterprise java bean) 是 Java 组件中一种用 Java 语言编写的用来实现开发人员指定的应用

逻辑，设计、开发、编译、布置服务器端的各个组件是它的主要功能。多个 EJB 可以组成一个应用（或一系列应用）的业务逻辑，只有和这些 EJB 组件彼此脱离后，才能够使用这些业务逻辑的前台应用。在 J2EE 体系结构中包含有一个 EJB Container Server，EJB Container 可以根据安全规则的使用、需求组件的装载、公开操作的调用，将事件支持服务提供给组件。EJB 是系统开发必备的组件技术之一，利用 EJB Container 可以完成众多任务。

1.1.4 JMS 组件技术

JMS (Java Message Service) 组件技术是 Java 平台上有关面向消息中间件的技术规范，它给 Java 程序提供了一种新建、收发、读取企业消息的常用手段，成为 Java 程序访问这些企业消息系统的一种公用机制。Sun 公司与合作伙伴开发的 JMS API 定义了一组公共的应用程序接口和相应语法，使消息组件能够与 Java 程序进行通信。使用 JMS API 之后，可以大大提高消息应用从某种环境转移到另一种环境下的难易程度。并且，设计人员可以在各种消息中间件之间进行操作，便于掌握及使用。

1.1.5 JDBC 组件技术

JDBC (Java Data Base Connectivity) 组件技术，是一组由 Java 程序编写的类和接口组合而成的，它可以将统一访问的接口提供给多种关系的数据库。只需要用 JDBC 编写一个程序，其他类型的数据库就可以不用再分别进行编写了。设计人员不再需要了解这种数据库存储系统的特定语法，可以把精力集中在数据上，真正实现开发数据库应用程序的一次编程、到处运行。

1.2 J2EE 的多层模式

在 J2EE 的多层应用结构中，不同的组件分布在相同层的不同设备中，而不同的组件代表着不同功能的应用。J2EE 多层结构将不同层面切分成许多层，能够为不同的服务提供独立的层，避免了传统模式中，客户端难于升级或改造，重用业务逻辑和界面逻辑非常困难的现象，可以更加方便灵活的进行维护扩展。

图 1 是 J2EE 典型的多层结构图：

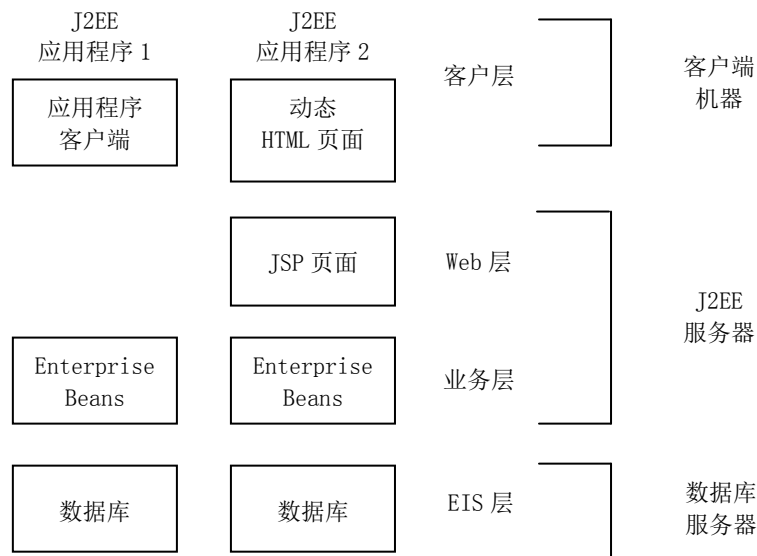


图 1 J2EE 多层结构图

下面具体介绍一下结构示意图中每个功能层的具体含义：

客户层

客户层的主要功能是为客户提供实现功能的界面显示，便于用户进行信息查看以及信息的采集和录

入。客户层分为两种类型，及基于非 Web 的客户层（独立的应用程序）和基于 Web 的客户层（主要作为企业 Web 服务器的浏览器）。

Web 层

为 J2EE 应用提供了 Internet 功能服务，J2EE Web 组件由基于 Web 的 applets、显示 HTML 页面的 servlets 以及 JSP 页面组成。

业务层

被称为 EJB 层，它是体系结构中的重要部分。在 J2EE 技术结构中，中间层的位置为业务层，它对数据交换起着承上启下的作用。业务层的主要任务是实现企业级信息系统的业务逻辑，制定业务的规则、实现业务流程等与业务需求相关联的系统设计。业务层的开发结果，直接关系着能否与实现层之间实现解耦。

企业信息系统层

企业信息系统层主要是处理企业信息软件，它主要包括了如 ERP (Enterprise Resource Planning)、主机事务处理、数据库系统，以及其他遗留的信息系统。

1.3 J2EE 设计模式

设计模式是对设计代码经过多次测试，并进行归类编排所总结出来的经验。设计人员通过运用设计模式，可以更加方便的分析代码，并且可以多次反复的使用代码，大大提高了代码的牢靠性。

构成设计模式的四个必要因素：

- 模式名。它是用来描述模式的实施计划和实施成果的专用称号。
- 疑问。设计模式除了描述了特定的设计疑问，和导致不灵活设计的类或对象结构。而且对设计疑问以及疑问存在的因果关系进行了解释，并描述了运用设计模式的正确时机。
- 实施计划。包括设计的组成部分，以及其相互关系和各自的职责和协作方式。解决方案不单单描述一个特定而具体的设计或实现，而是提供设计问题的抽象描述，且用一个具有一般意义的元素组合（类或对象组合）来解决这个问题的具体方案。
- 成果。绝大多数的软件成果注重的是时间与空间之间的权衡，以及对疑问实现过程中的描述。模式成果表述了运用模式时应当衡量的问题，以及模式对系统扩展性和便捷性的作用。

以 J2EE 计划为基础，运用 J2EE 技术标准解决不同项目和时间中多见问题的合集即为 J2EE 设计模式。运用得当的 J2EE 设计模式，可以大大降低系统研发所需的成本，缩短研发周期，并提升系统的重复利用率。

2. J2EE 技术架构应用分析

本文以协同办公系统的实现为例，针对 J2EE 技术架构的运用做出了详细分析与说明。协同办公系统的主要的核心目标是支持办公管理的应用服务，在互联网应用技术基础上，采用分布式事务管理的设计理念，并结合各客户端的应用需求，实现对公文流转、公共服务、会议管理、事务管理等各功能模块的管理。系统的设计模型共分为四层，其中最底层为数据层，主要针对信息数据、基础数据和知识数据进行数据交换和保存。与数据层相关联的是 J2EE 应用服务器，它的主要功能是将同步管理、功能管理、运行状况等方面的系统服务提供给上一层的工作流引擎和应用服务层。工作流引擎是系统的核心理念，它不仅创建并管理着工作流的实施，而且能够解释工作过程的定义，以及调用各种外部 IT 的应用，最终实现与工作流

参与者的相互交流，双方面互动。最上层为应用服务层，它包括为用户提供办公服务的各功能模块，如公文管理、事务管理、会议管理等。用户可以通过门户网站或 Internet/Intranet 来实现日常办公的各项应用。图 2 为协同办公系统设计模型图。

3. 结束语

总之，J2EE 技术架构在实际应用过程中，多方面呈现出其所具备的应用优势，以及非常卓越的性能表现，具体体现在以下几个方面。

1. 稳固的可靠性

J2EE 技术架构具有持久而又稳固的可靠性，它可以部署在鲁棒性非常好的操作系统之上，以满足部分客户对服务器端全天候无故障运转的要求。

2. 开发的高效性

开发人员使用 J2EE 应用服务器可以将精力集中在开发业务逻辑方面，而不用顾虑底层系统服务的繁琐性，从而大大降低了开发人员的工作量，缩短了开发周期。

3. 病毒的防范性

J2EE 对程序编译环节反复进行检测，对程序运行时存在的潜在故障做出预警，这样极大的避免了因服务器端感染病毒，而给用户带来的严重损失。

4. 支持异构环境

设计不依赖特定操作系统的 J2EE 可移植程序，开发人员只需要一次开发就可以在各种平台上进行部署。并且，客户也可以将与 J2EE 兼容的其他组件部署在异构环境中，这样既缩短了 J2EE 方案的实施周期，也大大降低了方案实施所需要的费用。

参考文献：

- [1]敖杰刚，分布式并行计算下 GML 空间数据复制同步更新机制研究[d]，江西理工大学，2012
- [2]蒋永生，彭俊杰，张武云，计算及云计算实施标准:综述与探索[J]，上海大学学报(自然科学版)，2013，(1)：50~65
- [3]Longzhuang Li，Space-Aware Data Integration for Ocean Observing Systems[J]，Procedia Environmental Sciences,2011,11
- [4]傅伟，邓素平，涂刚，熊平，基于 J2ME-J2EE 的实验室信息管理系统的设计与实现[J]，廊坊师范学院学报(自然科学版)，2012，(6)：67~71
- [5]刘湛，J2EE 的核心 API 与组件，hops ://www.ibm.com/developerworks/cn/java/j zee/#5
- [6]钟露明，基于知识管理的协同办公系统设计及实现[J]，计算机与现代化，2013，(6)：6~7
- [7]赵杨，国家创新系统中的信息资源协同配置研究[d]，武汉大学，2010
- [8]姚丽华，于广州. 基于动态 JSP 技术的 Web 应用软件开发[J]，信息与电脑(理论版)，2013，(3)：35~50
- [9]王尔睿，吴颖丹，协同办公系统的安全设计[D]，电脑开发与应用，2013，5：88~92
- [10]逢久，浅谈系统集成之数据整合[D]，电子技术与软件工程，2013，(11)：150~156

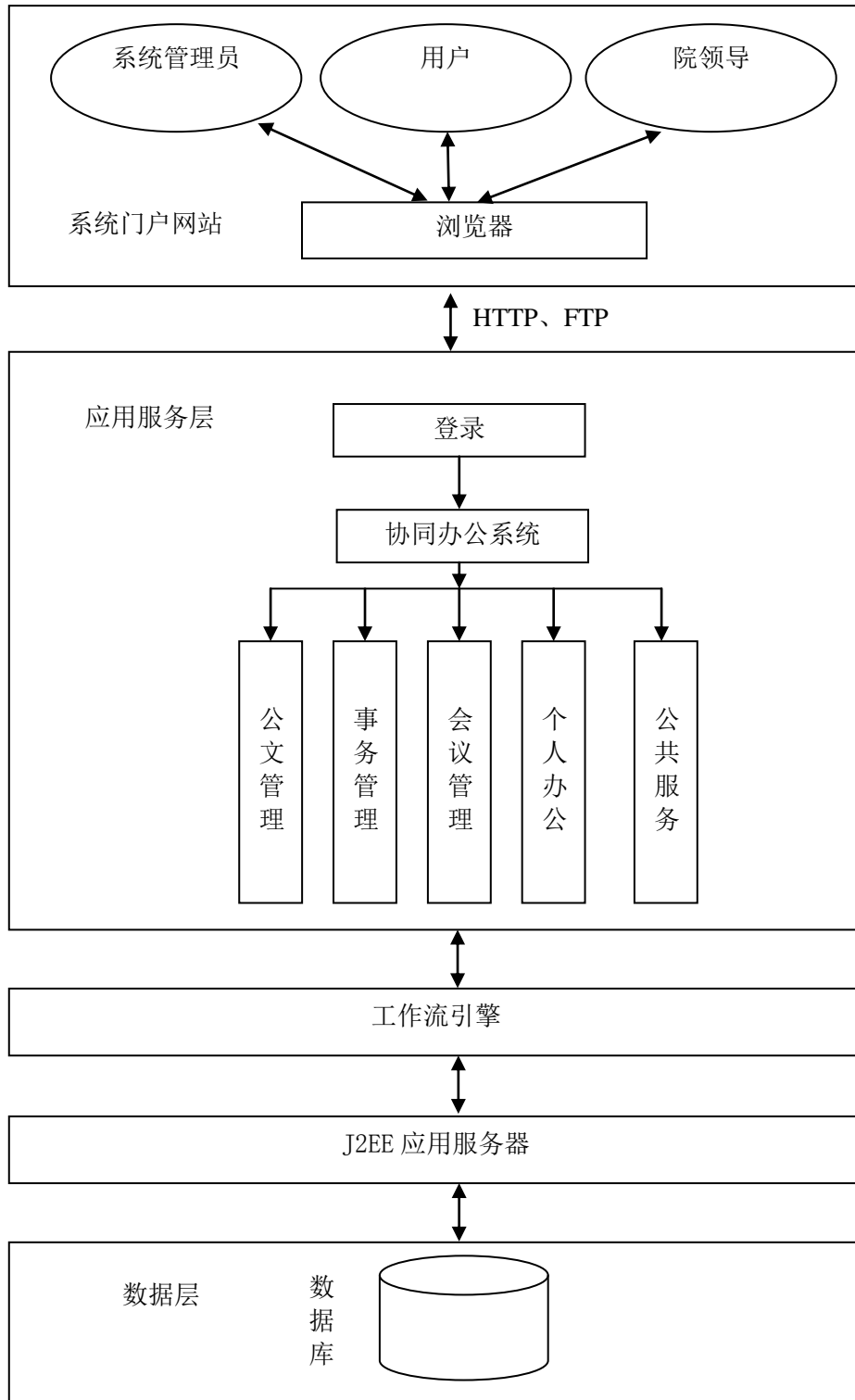


图 2 协同办公系统模型