

山东大学

硕士学位论文

基于JAVA EE平台的项目管理系统的设计与实现

姓名：张晓良

申请学位级别：硕士

专业：软件工程

指导教师：史清华

20080415

摘 要

近几年致力于软件开发项目管理和软件开发过程改进的方法论层出不穷，敏捷开发方法便是其中一支。同时，人们试图通过相关的软件系统来实践新兴的软件开发项目管理方法，更力图通过软件来改进软件开发项目的管理和过程，降低软件开发项目中的各种潜在风险。

亿帆环球科技公司（E5 Systems Inc.）根据敏捷开发方法，并结合自己公司的具体情况制定了一套符合自己公司的软件项目管理方法论，并力图开发一套项目系统，来实践这种管理方法论，并辅助公司的项目过程管理。

本文分析了当前国内外项目管理系统的发展和应用现状，并对应用于敏捷项目管理的项目管理系统提出了设想和要求。本文着重描述了从需求分析、总体设计，到详细设计，再到系统实现的一个项目管理软件开发的全过程。

在需求分析部分，根据公司的具体需要，对需求进行了研究和记录，并用 UML 用例图的方式进行表示。在总体设计部分，根据系统设计的一般原理，和系统需求分析的结果，得到本系统技术、功能、逻辑、数据、安全等方面的设计，为具体的系统开发提供了方向。

在详细设计部分，本文对该系统的后台管理部分、前端展示部分，以及跨平台系统接口设计部分做了详细说明，并规划出了系统的具体类和类之间的关系，得到了一个大致的系统技术原型。在这一部分，本文还对 AJAX 技术、跨平台方法调用等技术热点和难点做了重点解释。

在系统实现部分，本文对系统中各种技术要求的具体实现做了详细介绍，并辅以代码说明，同时还简单介绍了相关技术类库、框架的配置和使用。

关键词：敏捷开发方法，项目管理，JAVA EE，AJAX，RMI

ABSTRACT

In the current years, many methodologies about software project management and the improvement of software development process come out one after the other. The Agile Methodology is one these methodologies. At the same time, people are trying to use some concerned software systems to practice these new methodology of software project management, and they want to use these systems to improve the process and management of software project to reduce hidden risk of software project.

According to its own situation, E5 System Inc. designed a software project management methodology which is based on Agile Methodology. And it is trying to develop a project management system to practice this methodology and assist the project management.

This paper analyses the situation of project management system development and application, and gives some ideas and requirements for agile software project management system. This paper describes emphatically the process of requirement analysis, specification design and implement of a project management system.

In requirement analysis, requirements are gathered and recorded according to the company's situation and the UML use case diagram is used to present these requirements. According to the result of requirement analysis and the general principle of system design, the entire system design is acquired. It includes the technique, function, logic, data and secure design of this system and it provides the direction of the system development.

In the specific design, the paper describes the background management part, the frontend presentation part and the system interface for multi system in detail. And it gives a design of classes and relationships among these classes in this system. It also explains the technical difficulty of AJAX and Remote Method Invocation.

In the system implementation, this paper explains each required technique in this system thoroughly and uses some code as examples. And it also describes the usages and configurations of libraries and frameworks which are used in this system.

Key words: Agile Methodology, Project Management, JAVA EE, AJAX, RMI

原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律责任由本人承担。

论文作者签名: 张晓良 日 期: 2008.4.15

关于学位论文使用授权的声明

本人完全了解山东大学有关保留、使用学位论文的规定，同意学校保留或向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅；本人授权山东大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或其他复制手段保存论文和汇编本学位论文。

(保密论文在解密后应遵守此规定)

论文作者签名: 张晓良 导师签名: 史清华 日 期: 2008.4.15

第1章 绪论

1.1 系统开发背景

自 20 世纪计算机引起的信息技术革命开始，计算机已经渗入了我们工作生活的各方面。伴随而来的，软件开发行业的迅速膨胀，各种软件开发项目成为信息技术行业发展的前沿阵地。而与之匹配的理论和实践指导的学科——软件工程，却是一个新兴学科，在很多方面依旧很不成熟。软件项目管理作为软件工程学科的一部分，现阶段仍处于一个经验总结和实践摸索的阶段。虽然现在已经有了很多软件项目管理的模式与经验，但软件开发项目特有的不确定与不可度量的特性依然使软件开发项目处处陷阱，严重影响着软件开发项目的质量^[1]。

近几年致力于软件开发项目管理和软件开发过程改进的方法论层出不穷，如专注于规范过程和文档约束的统一开发过程，立足于快速应变、快速开发的敏捷开发方法。与之匹配的软件系统也开始渐渐走入人们的视线。人们试图通过相关的软件系统来实践新兴的软件开发项目管理方法，更力图通过软件来改进软件开发项目的管理和过程，降低软件开发项目中的各种潜在风险。

亿帆环球科技公司（E5 Systems Inc.）根据敏捷开发方法，并结合自己公司的具体情况制定了一套符合自己公司的软件项目管理方法论。现在亿帆环球科技公司已经有了一套自己的用于管理项目人员工作与公司花销的人力资源管理系统。该公司现试图在现有的人力资源管理系统的基本上加入符合自己公司项目管理方法论的软件项目管理后台，并全新开发与该系统使用共同数据资源的前端项目管理交互系统。本次改进主要专注于项目需求的获取与跟踪、软件开发过程的缺陷确认与跟踪、项目报表生成、项目风险预测与报警。而全新开发的项目管理交互系统主要用于向公司客户实时反映项目状态、并提供相关报表，实时收集客户回馈信息。通过此交互系统，实现敏捷开发的要求之一：真实客户参与^[2]。通过有目的地将客户真实地引入到需求、设计、开发、测试等软件项目开发的各个环节中，来变化的管理用户需求，减少软件开发项目越到后期风险越大的状况出现。

公司新模块和新系统的主要目的是为了辅助公司领导层把握项目状态、预测项目风险，并为公司领导层决策提供详尽的项目信息分析和支持。根据亿帆公司的软件开发项目管理方法论，客户实时参与需求管理、开发过程小步迭代、项目产品短期发布、即时客户回馈、项目风险预测是方法论的关键点。此次开发就力图将这些问题通过软件来规划、实现。另外，新开发的交互系统直接用于公司和公司客户进行信息沟通，所以系统的用户体验也是一个重要问题。为此，这次开发中使用了大量的 AJAX 相关的新技术，来增强 WEB 系统的表现能力。

上述是此次开发的目标与关键点。同时，这又是对亿帆公司软件开发方法论和新技术的一次实践和总结，也用于寻找和测试亿帆项目管理方法论和新技术的优劣和盲点。所以说此次开发除了产生可用的软件产品以外，相关的经验积累将是更宝贵的财富。

1.2 国内外研究开发现状

目前，信息技术界对软件项目管理的研究大多还是停留在理论研究与实践指导方面，大多数还是通过文本、规章制度等书面性质的约束来实现对软件项目各个环节的统一管理。目前国内能对软件项目管理所有阶段提供统一支持的软件系统为数并不多。现在在实际项目管理工作中，常用的用于项目管理控制的软件大致有以下两种：

1. 以 Microsoft 的 Project 软件为代表的桌面级的项目过程管理控制系统。传统的项目过程管理控制系统，Project 无疑是最成功的一个。它以一个单机应用程序的形式，提供了项目过程规划、时间安排、里程碑设置、资源分配、工作时间管理等功能，同时它还可以实现检查资源配比是否合理、资源压力是否过大、关键路径检查等项目管理的主要工作^[3]。虽然 Project 软件发展至今已日渐成熟，但是传统的单机应用程序难以扩展、难以和新的软件系统结合、难以根据实际工作变化做出改变。所以 Project 在日新月异的软件行业仍有其不能避免的弊端。

2. 以 XPlanner 为代表的新型的网络应用型的项目过程管理控制系统。随着网络应用的成本不断降低、网络技术的不断成熟，以 XPlanner 为代表的新型的

基于网络架构的项目过程管理控制系统不断出现。XPlanner 是以敏捷软件开发过程方法论为指导的项目过程管理平台。它提供了项目管理、项目资源管理、需求跟踪、持续集成跟踪、缺陷分配和跟踪、项目报表生成和导出等功能。同时它还充分利用了网络应用轻便、易于扩充的特性，添加了很多功能亮点，如可以和常见的缺陷管理跟踪系统无缝结合、可以和常见的版本管理系统结合等功能。但 XPlanner 是以敏捷软件开发过程方法论为指导开发的管理系统，所以可能并不适用于运用其他方法论进行管理的项目实践活动。

虽然软件工程至今已有有了近 30 年的研究史，但相关方法论和辅助工具仍在处于一个经验总结和力求完善的阶段。无论什么样的实践行为，无论成败与否，我们都可以视为这是在方法论研究道路上的一次摸索。亿帆科技公司力图在公司管理与项目管理的本质上找到一致之处，并统一规划统一管理，来降低公司管理层面上的浪费与责任不清。而且亿帆公司更努力将这种思想用软件系统表现出来，用软件规约代替含混的规章制度。这将是亿帆科技公司将敏捷方法扩展到公司全部项目，乃至整个公司管理过程的一次有意义的探索。

1.3 本文的主要工作

目前而言，亿帆环球科技公司已经有了一个具有简单项目管理和人力资源管理功能的软件系统，但系统功能较为单一，而且主要侧重于人员管理方面，与真正的软件项目管理软件系统差距较大。本次开发工作的目标就是在目前系统的基础上，开发一个能覆盖软件项目整个生存周期的管理模块，并使用 AJAX 相关的新技术开发一个客户交互系统，用于展示项目状态、信息和数据报表等。基于现有系统新开发的模块可以视为整个项目管理系统的管理后台，而新开发的交互系统可以视为整个项目管理系统针对公司客户开发的前台交流系统。

根据亿帆公司实践需要，目标新系统和新模块的主要功能包括增强现有系统的项目管理功能、增加项目需求收集、需求控制、需求跟踪模块，增加软件缺陷跟踪模块，增加项目预算与实际开销总结模块，增加各种项目报表功能，增加项目风险估算、预警与邮件提醒功能；开发一个与之相对应的，用于和客户进行交流的新的前端展示交互系统。新的交互系统主要目标是实现一个良好的客户界面

接口，提高客户使用系统时的用户体验；提供丰富的数据展示手段，动态的向客户展示项目变化过程与数据；提供丰富的数据报表与项目风险估算，让客户对项目状态有准确认识；提供客户信息回馈与收集功能，力图通过互联网实现客户与项目之间的完整的信息交流。

本文的主要工作有：

1. 改进当前系统架构不合理的架构设计和开发框架，并为了实现新系统，引进新的技术。

当前亿帆公司的系统是基于经典的 JAVA EE 轻量级三层架构模式的软件系统。虽然使用的是经典设计，但系统中存在很多对这些框架的误用和滥用。本文在架构设计方面力图扭转这些误用和滥用，使系统架构设计更灵活、更安全，为下一步系统功能扩展提供基础保障。

2. 开发完备的、符合亿帆科技项目管理方法论的项目管理模块。本文将给出项目管理模块的完整用例和其中重要部分的部分实现。

3. 开发使用 AJAX 技术前端展示系统。

本次开发，前端展示系统是重点之一。此系统用于和公司客户交流项目状况。由于此系统是直接与公司客户进行交互，所以此系统在实现数据交流和信息展示的同时，也很重视用户体验。所以传统的 WEB 展示模式已经不能满足新的展示需求。所以在进行这部分开发的时候，运用了大量的 AJAX 和富客户端相关技术来实现更高级的人机交互界面和数据展示。本文将阐释新系统的详细设计和新技术的应用方法。

4. 完成新功能模块和新系统的集成、部署和测试。

本次开发略微复杂，有一部分是基于旧系统的二次开发，有一部分是新系统的全新开发。所以新旧功能模块的整合、新旧系统的集成都是关键问题。本文将介绍如何利用 Spring 作为业务容器实现新旧模块的沟通和如何实现新旧系统的信息沟通。

1.4 论文的组织结构

全文共分为六章。

第一章是绪论，主要介绍系统的开发背景，阐述国内外相关思想和技术的研究现状，以及说明了本文所要解决的主要问题。同时还说明了本文的组织结构。

第二章是系统需求分析，描述了本次开发的需求获取的方法与过程，并对系统需求进行了明确和界定，总结出了系统功能性需求。并通过对此类的项目实际应用的总结，分析出非功能性需求。

第三章是系统总体设计，根据系统设计的一般原理和原则，对系统的总体架构设计和功能架构设计进行阐述，规划系统的大体轮廓，并确认系统的重点部分。

第四章是系统的详细设计，是本文的重点。在本章中，从系统各个功能模块出发对每个功能模块的具体实现进行详细描述，阐释其具体技术实现方案，并对系统中相关的对象、对象关系、数据库进行设计。

第五章是系统的实现，介绍如何将系统的详细设计在互联网平台上进行实现，如何将新模块融入到系统中，如何实现新老系统进行交流沟通。同时，也对数据库的整体结构和设计进行了说明。

第六章是总结和展望，对本次开发工作和论文创作工作进行了总结，并对软件项目管理软件的设计提出了更多设想。

第2章 系统需求分析

需求分析是理解真实世界中的问题和用户需求，并找出能满足这些需求的解决方案的过程。需求分析的目标是在开发活动开始以前，对问题的解决方法得到更好的认识。由此可见，需求分析在整个项目过程中具有举足轻重的地位。它甚至决定了以后开发活动的大方向。能否得到一个准确全面的需求分析结果将直接影响软件开发项目的质量和成败。

软件需求又分为两个部分：功能性需求和非功能性需求。功能性需求是指系统必须完成的那些业务要求，即为了向它的用户提供有用的功能，产品必须执行的动作；非功能性需求是指产品必须具备的属性或品质，如可靠性、性能、响应时间、容错性、扩展性等等。

本章将结合本次开发的具体情况，对目标系统的功能性需求、非功能性需求进行分析。

2.1 系统概述

目前，亿帆科技已有一个具备基础项目和人力资源管理功能的软件系统，该系统现在具备的基础功能模块有项目的设置与管理、人员管理、人员工作时间管理、项目基本财务支出管理等基本功能。

本次开发的目标是在此系统的基础上，开发一个软件开发项目管理的后台管理模块。增强后的新系统提供的管理功能应该能覆盖软件开发与维护项目的整个生命周期，包括项目状态的报告与修正意见管理，需求获取、分析与变更的管理，项目里程碑的计划与回馈管理，项目风险管理与控制，客户回馈问题管理，项目资金预算管理，项目资源规划管理，项目客户管理，软件缺陷跟踪系统配置与信息获取，以及与这些功能相搭配的配置管理模块。同时针对这些新功能，开发一个用户界面友好的信息展示交互系统。除此以外，出于安全因素考虑，还需要为这些系统和模块开发一个相搭配的基于权限和角色的认证机制。

依赖于现有系统的管理模块与新开发的信息展示交互系统的本质关系是后

台管理功能与前台展示界面的关系。但是由于两者的参与者不同，所以信息展示的过程中会对后台管理的信息进行筛选和处理，同时前台展示系统还承担着客户信息收集的责任。

本次开发的系统结构如下图所示：

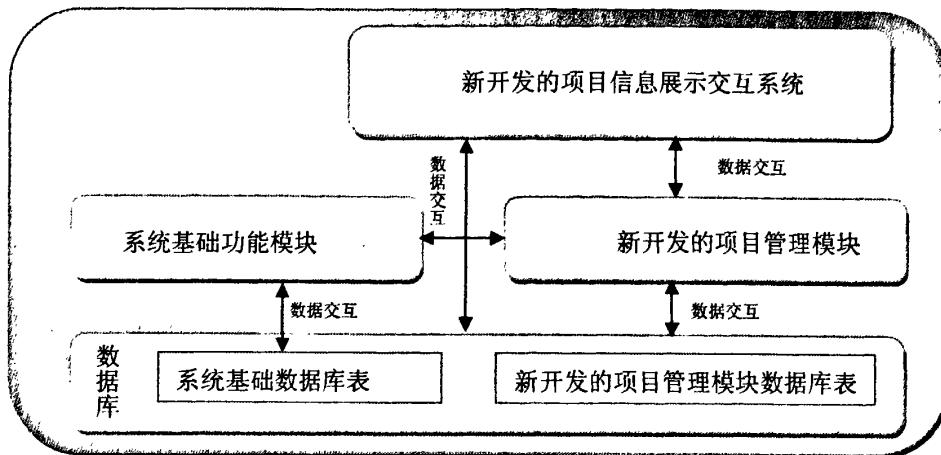


图 2-1 亿帆公司项目管理系统开发结构图

由此可见，这次项目管理功能的开发可以视为是利用现有公司系统基础资源的二次开发。开发的目标除了产生新的项目管理的软件系统，还要保持新老系统的相互融合，避免功能模块的重复开发以及资源浪费。

2.2 系统需求的获取与方法

需求分析的目标是通过对现实问题和用户需求的理解和调查，找到问题确定主要问题并建立需求模型。由于亿帆科技公司是一个海外开发的公司，其总部设在美国而开发设计人员在中国，所以在需求分析的过程中，我们采取了更符合我们公司具体情况的方式与方法。

在需求获取与分析过程中，缺少用户参与用户需求不断变化往往是造成需求获取与分析效率低下和工作有效开展的主要原因。

为了避免这种因素对需求分析过程的消极影响，首先从公司实际工作需要出发，确认此次需求获取的主要来源与需求分析的参与者以及最终产品的潜在使用者，包括公司的管理层代表、项目经理代表、客户经理代表、业务分析员代表、开发人员代表，以及公司客户代表。通过各方面人员的全程参与和充分的相互交

流来保证需求获取的最大化，扫清需求分析中的盲点。

其次，在需求分析的过程中，采取极限编程方法论中所提倡的真实用户参与和用户故事的方式，来使用户投入到整个需求分析，甚至某些试用环节，来最大限度的提前修正对需求的错误理解和挖掘潜在的用户需求，从而减少开发阶段后期的潜在问题和降低需求频繁变化带来的高风险。

根据上述方法，我们在实际工作中采取会议讨论法（包括网络视频会议）、用户角色扮演、简易原型演示等手段来组织对需求分析、挖掘的过程。

最终根据需求分析的结果，产生系统用例模型来对其进行记录和研究。用例就是如何使用系统来达到目标的一组情节，其本质是通过写出多种使用系统的情节（完成不同项目相关人员目标）来发现和记录功能性需求。用例可以被看成是一种承诺或契约，规定了系统以后将会以怎样的方式运行^[4]。

另外，需求获取和分析并不是一次就可以完成的工作。在整个系统的开发、实现过程中，用户的需求可能随时会产生变化，而这些变化都有可能对正在进行的开发工作造成重大影响。由于亿帆公司的开发方法论是基于敏捷开发过程的，小步迭代、短期发布、增量设计等原则在整个软件的开发过程中的到充分应用，所以对于此次开发来说，整个开发过程也可以认为是一个不断进行需求确认的过程，可以比较容易的应对开发过程中的用户需求变化。

2.3 需求问题描述

根据需求分析的结果，通过最终的用例模型，可以确定系统的主要功能和开发目标。另外，系统还应该具备使用友好、安全、能适应大数据量计算、数据精确等用例文档无法描述的非功能性需求。具体功能性需求和非功能性需求，如下：

2.3.1 系统参与者

由于我们使用用例来记录系统的功能性需求，所以在需求分析的第一步我们需要明确系统的参与者。参与者（Actor）是具有行为能力的事物，可以是一个人（由所扮演的角色来识别）、计算机系统或组织。经分析确认，本系统的参与者主要有公司客户、项目管理者、公司管理者、系统管理员等四类。四类参与者

的相关人员和能够参与的系统功能如下表所示：

表 2-1 系统参与者

参与者	相关人员	功能描述
公司客户	和公司有项目关系的相关客户	登录前端展示系统 浏览与自己有关项目的相关信息和生成数据报表 针对项目添加回馈信息
项目管理者	包括公司的项目经理、客户经理	具有“公司客户”角色的所有权限 能登陆管理后台模块管理与自己有关的项目相关内容，如项目状态、需求变更、关键事件、客户回馈等，并能检查、管理与自己有关项目的财务预算、风险预测、人员调配等
公司管理者	包括公司管理层人员，如公司 CEO、董事长等	具有“项目管理者”角色所有权限，与之不同的是项目管理者只能查看与之相关的项目信息，而公司管理者可以查看公司任何一个项目的相关信息
系统管理员	此软件系统的管理员	具有上述角色的所有权限，并能配置系统的一些基础设置，如缺陷管理系统的服务地址等

如上图所示，用例中的参与者所有的能力具备一定的继承关系。按照参与者所具备能力的由少到多排列，后面的参与者可以继承前面参与者的能力，并根据实际需要进行扩展，来满足自己的功能要求。这种设计为系统中基于权限和角色认证的安全认证系统打下了基础。

2.3.2 系统功能列表

根据需求分析，该系统的功能大致可以分为两部分：一是后台管理功能，二是前台展示功能。功能列表如下：

表 2-2 后台管理功能列表

功能	功能细化
用户登录与登出 (Login & Logout)	用户登录, 通过权限角色验证, 获取相关访问能力 用户登出, 注销当前访问信息
项目状态简报管理 (Status Summary)	添加、删除、修改项目简报 项目简报提交与审批流 根据指定条件检索项目简报
项目里程碑管理 (Key Events)	添加、删除、修改项目里程碑 记录项目里程碑的变化过程 根据指定条件检索项目里程碑
项目需求变更管理 (Change Request)	添加、删除、修改项目需求变更 记录需求变更过程 根据指定条件检索需求变更
项目风险管理 (Risk Management)	添加、删除、修改项目已出现或潜在的风险问题 记录风险问题的状态改变 根据指定条件检索风险问题
项目新需求管理 (Issues Management)	添加、删除、修改项目后期出现的新需求 记录新需求的状态变化过程 根据指定条件检索新需求
项目财务预算管理 (Cost Management)	根据月份添加、删除、修改项目预算 根据时间检索项目预算
项目资源管理 (Resource Planning)	创建项目所需的人力资源计划, 并按照项目成员的角色进行划分和统计
客户账户管理 (Client Users)	为制定项目创建用于登录前端展示系统的客户账户。
缺陷跟踪系统配置管理 (Mantis Configuration)	为指定项目创建 Mantis(一种缺陷跟踪系统)服务的相关信息, 包括服务器地址, 服务器账户, 并提供服务测试功能。该功能用于配置 Mantis 服务器, 以请求获取当前项目的缺陷数量以及修正情况
系统参数管理 (Parameters)	配置系统相关信息和关键字颜色等 主要用于为前端展示系统的动态效果提供后台配置支持。

表 2-3 前台展示功能列表

功能	功能细化
用户登录与登出 (Login & Logout)	用户登录，通过权限角色验证，获取相关访问能力 用户登出，注销当前访问信息
项目概要统计 (Overview)	统计并显示项目当前相关信息，包括当前项目状态简报、最新需求变更、最近里程碑状态、最新项目风险问题、最新需求等 生成报表并提供导出成 Microsoft Word 文档功能
项目状态简报详细统计与信息回馈 (Executive Summary)	检索、统计项目状态简报信息 显示指定项目状态简报信息细节，并提供导出成 Microsoft Word 文档功能 项目状态简报审批 针对某一项目状态简报添加回馈信息
项目新需求详细统计与信息回馈 (Issue)	检索、统计新需求信息，并提供导出成 Microsoft Excel 文档功能 显示指定新需求信息细节，并提供导出成 Microsoft Word 文档功能 新需求状态更新 针对某一个新需求添加回馈信息
项目风险问题统计与信息回馈 (Risks)	检索、统计项目中的风险问题，并提供导出成 Microsoft Excel 文档功能 显示指定风险问题细节，并提供导出成 Microsoft Word 文档功能 风险问题状态更新 针对某一个风险问题添加回馈信息
项目里程碑统计与信息回馈 (Key Events)	检索、统计项目里程碑，并提供导出成 Microsoft Excel 文档功能 显示指定里程碑细节，并提供导出成 Microsoft Word 文档功能 里程碑状态更新 针对某一个里程碑添加回馈信息
项目需求变更统计 (Change Request)	检索、统计需求变更历史 点击显示指定需求变更细节
关键信息图表展示	动态显示项目状态和进展程度图表 动态显示项目消耗工时信息 动态显示项目财务状况信息 动态显示项目缺陷数量以及修正比例

2.3.3 系统用例描述

根据上述系统参与者和功能列表，可以确定参与者与系统功能之间的关系，从而画出系统需求的用例图。用例图是对系统语境的勾画，显示了系统的边界，哪些东西在系统边界之外，以及如何使用系统。

本系统的简单用例图如下：

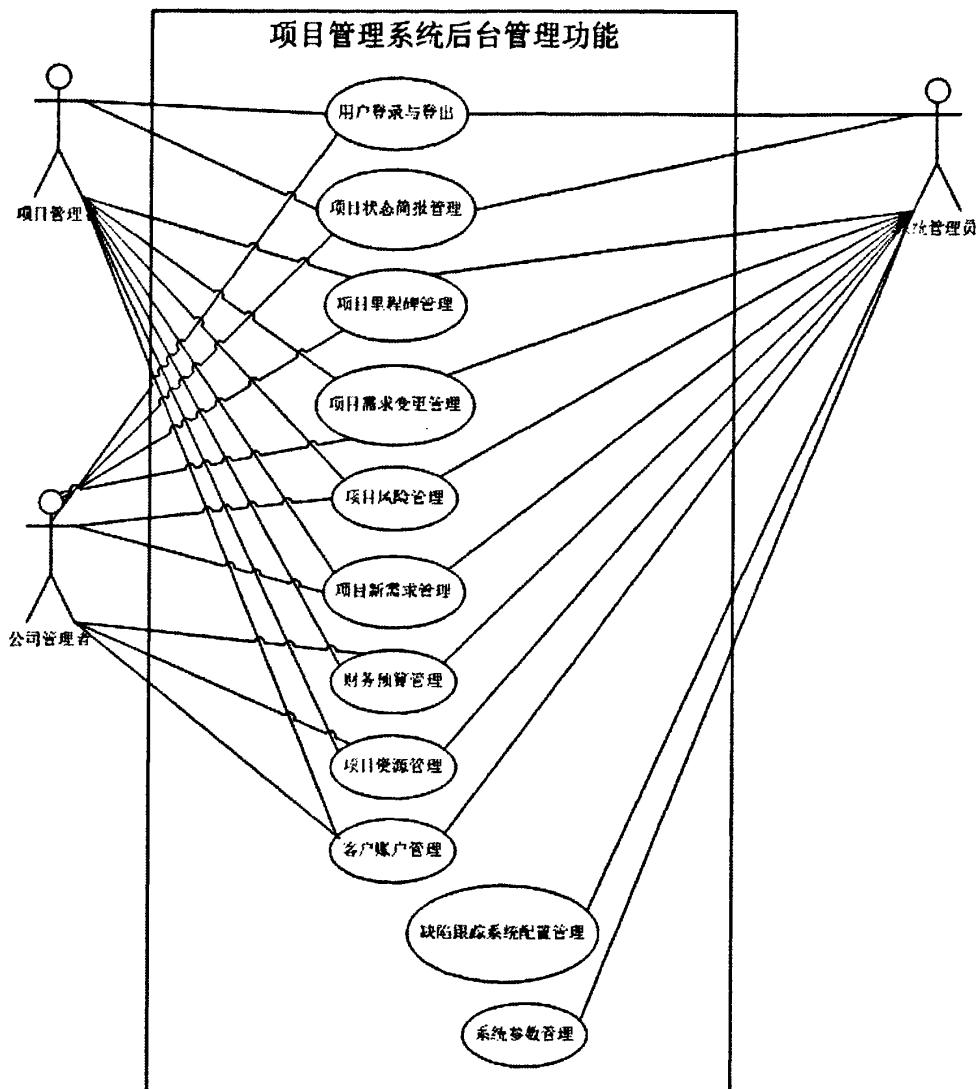


图 2-2 项目管理后台用例图

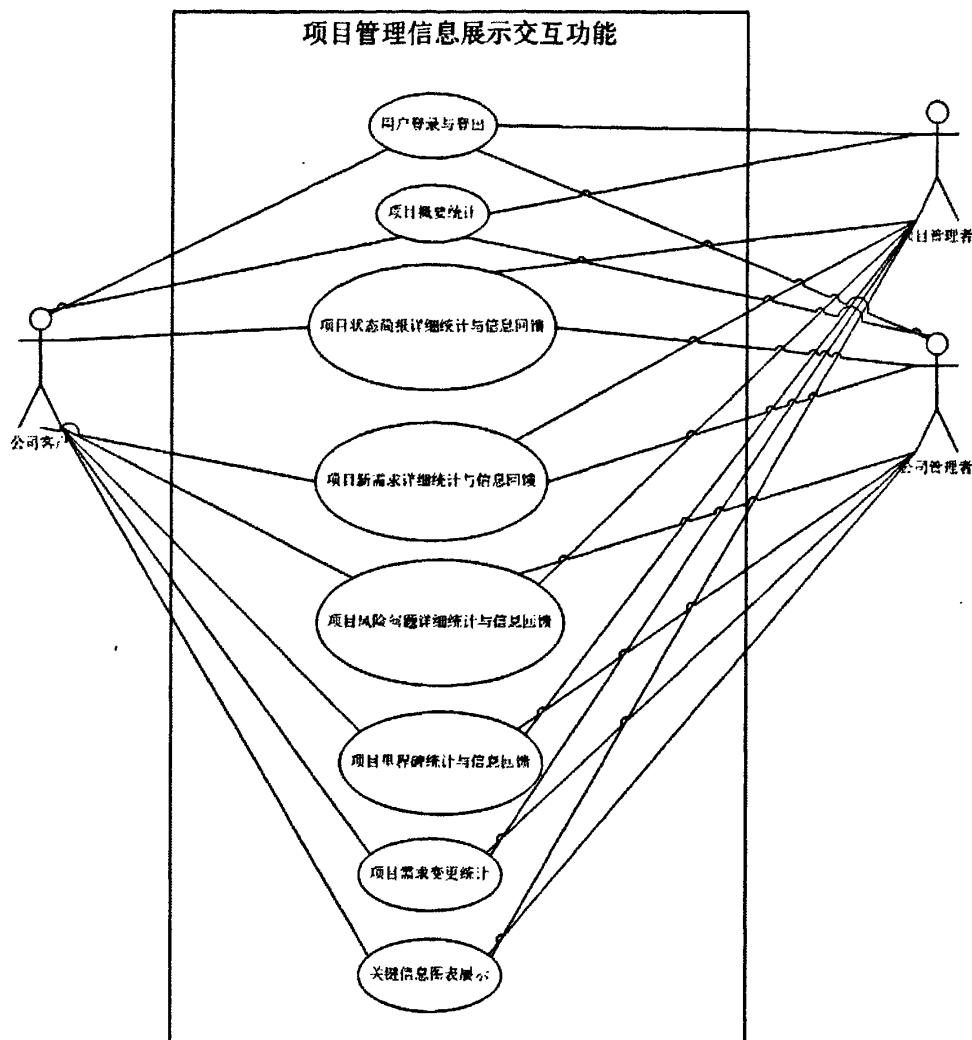


图 2-3 项目管理前端展示用例图

上述图中一个代表功能需求的用例有可能是一个独立用例，也有可能是根据具体场景的不同通过几个不同用例组合实现的。

由于系统功能众多且功能大体类似，下面只选取具有代表性的几个用例进行细化，结合其使用场景及约束条件等进行详细说明。

1、后台项目状态简报管理

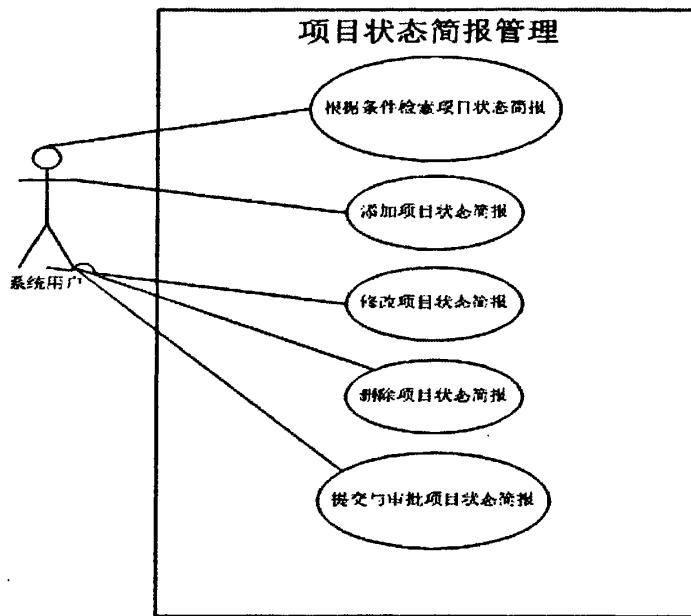


图 2-4 后台项目状态简报管理用例图

项目状态简报是指每周对项目当前工作进展情况的记录和总结，包括对本周工作完成情况的总结和汇报，对当前项目的主要困难和问题进行阐述和分析，并对下周工作做出计划和预测。

该部分的主要参与者有项目管理者，公司管理者。项目管理者定期根据项目具体情况提交项目状态添加并提交自己项目的状态简报。然后由公司管理者进行审阅。公司管理者在进行审批行为的时候需要输入自己对项目进展的态度以及评注信息。被公司管理者审批通过的项目状态简报才会被前台展示系统检索并显示。

项目管理者和公司管理者都可以对与自己相关的项目状态简报进行增、删、改、查等操作。

表 2-4 后台项目状态简报管理用例说明

用例标识	STATUSREPORT_SUMMARY_01		
用例名称	项目状态简报管理		
创建时间	2006-12-08	最后修改时间：	
参与者	项目管理者、公司管理者		
用例描述	供项目管理者、公司管理者在线交流项目状态与工作计划		
前置条件	参与者已登录后台管理系统		

表 2-4 后台项目状态简报管理用例说明（续表）

后置条件	参与者完成项目状态简报的输入与评论
基本事件流	1、参与者登录系统 2、进入项目状态简报管理页面 3、根据页面提示可进行按条件检索、增加、修改、删除、审阅等动作 4、保存所做修改
扩展事件流	1、增加项目状态简报：参与者选择增加简报功能，选择要添加简报的项目并输入简报内容，确认并保存 2、修改项目状态简报：根据提示条件检索找到目标简报，点击简报标题进入修改页面，修改并保存 3、删除项目状态简报：根据提示条件检索找到目标简报，点击删除按钮，根据提示确认并删除或取消删除动作 4、审阅项目状态简报：根据提示条件检索待审批的项目简报，点击审批按钮，根据提示确认审批并输入评注信息或取消审批动作
备注	1、项目管理者和公司管理者权限不同。项目管理者只能对自己相关项目的状态简报进行操作，而公司管理者可以对所有项目的状态简报进行操作 2、只有公司管理者具备审批项目简报的能力

2、前端项目状态简报详细统计与信息回馈

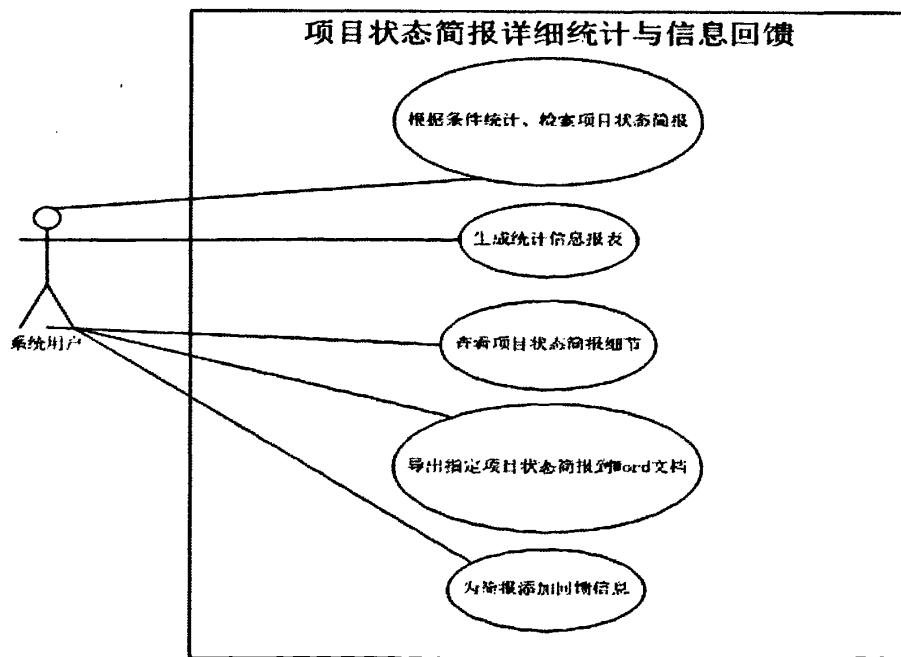


图 2-5 前端项目状态简报详细统计与信息回馈用例图

前端用于展示项目状态简报的统计、展示以及回馈信息交互的功能与后台项目状态简报的主要面向用户不同，所以两者的侧重点有很大不同。后台管理的主

要功能侧重于信息本身的管理，而前端展示与信息回馈更注重使用更友好的方式将项目状态信息展示出来。

此用例的主要参与者是公司客户，而他们在项目管理过程中并不是起主要作用，而只是作为顾问存在的，在实际工作中，更多的是为他们提供状态信息和听取他们的意见。所以前端展示系统并不报告信息管理方面的功能，而只是提供用户回馈信息采集的功能。前端展示系统的主要功能就是提供友好的、优化的用户查询交互界面和多样的信息展示与导出功能。

表 2-5 前端项目状态简报与信息回馈的用例说明

用例标识	DASHBOARD_SUMMARY_01		
用例名称	项目状态简报展示与信息回馈收集		
创建时间	2006-12-10	最后修改时间：	
参与者	公司客户、项目管理者、公司管理者		
用例描述	动态向参与者提供项目状态简报信息统计、展示与数据导出		
前置条件	参与者已登录前端展示系统		
后置条件	完成参与者对项目状态简报的回馈信息采集		
基本事件流	1、参与者登录系统 2、在左侧项目树中选择欲查询的项目 3、在右侧工作区功能选项卡中选择项目状态信息简报(Executive Summary) 4、根据屏幕条件提示进行检索 5、对检索结果导出（导出成 Microsoft Excel 文档） 6、可双击项目状态信息简报的 ID 号，进入查看简报细节并添加回馈信息或进行导出操作		
扩展事件流	1、添加回馈信息：根据页面提示选择条件检索到目标简报，双击简报 ID 号进入简报明细页面，在明细页面下部信息输入处输入回馈信息，确认并添加 2、导出项目状态信息简报到 Microsoft Word 文档：根据页面提示选择条件检索到目标简报，双击简报 ID 号进入简报明细页面，点击 Export to Word 按钮进行导出		
备注	1、公司客户、项目管理者和公司管理者权限不同。公司客户、项目管理者只能对自己相关项目的状态简报进行操作，而公司管理者可以对所有项目的状态简报进行操作 2、在此处添加的回馈信息同样会反映在后台管理系统的相关位置		

3、其他功能用例

由于项目管理系统并没有复杂的业务逻辑，其重点是项目信息的采集、管理、回馈以及项目数据的统计、展示，各模块除了具体信息类型区别较大以外，业务逻辑非常相似，所以此处不再一一描述各个用例的详细细节，只选取业务逻辑差距较大的简单介绍相关的用例信息。

项目关键信息的图表显示是用具有动态效果的柱状图、折线图、饼状图等图表来显示系统统计模块统计的项目关键信息。这些关键信息包括项目进展状况、项目工时消耗情况、项目财务状况、项目缺陷数量与修正比例等。这些信息都是由系统中的守护进程定时对系统数据进行归纳统计，并保存起来的。而此功能的主要要求除了用规定图形展示规定内容的功能性需求以外，更多的要求是数据可靠性、数据统计性能、信息展示友好程度等非功能性需求。

2.3.4 非功能性需求

软件产品的非功能性需求是指产品为了满足用户业务需求而必须具备的功能性需求以外属性或品质，如可靠性、性能、响应时间、容错性、扩展性等等^[5]。

长久以来，大多数软件开发组织在需求方面的工作重心只放在功能性需求上，对非功能性需求的理解不充分，重视程度也不够。这种情况将严重影响软件产品的质量，甚至还会使功能性需求的设计出现偏差。所以对非功能性需求的准确定义也具有非常重大的意义。

亿帆科技公司的项目管理系统平台在设计之初就明确了平台的主要用户、主要业务。从这些关键问题出发，在系统设计之初，就对非功能性需求提出了严格的规定。

本系统的非功能性需求定义如下：

1、 可用性。由于本次开发在一定程度上是基于已有系统的二次开发。新开发的功能模块和已有模块有很多数据交互。所以本系统首先要保证软件产品的可用性，即新开发功能不影响原有系统功能的良好运作，新开发功能能在统一的系统基础上满足预期的业务需要。

2、 系统可靠性。由于本系统是一个会长期运行的对外提供服务的软件平台，所以其故障出现率应该控制在一个可以接受的范围内，不能超过 1 次/1 月。而且不应该出现不能修复的故障。

3、 系统可维护性。系统以及系统数据应该是可备份可维护的。系统定于每月初维护一次，同时对系统及数据库进行备份。当运行出现严重问题时，系统应该可以从历史备份中恢复。

4、 安全性。由于本系统管理的数据中包含公司具体项目的敏感数据，所以要求能满足信息传输、系统使用时的安全性。在开发时，应该主动采用一些加密协议和系统安全认证机制。

5、 数据准确性。本系统中有一部分数据计算与统计功能。这些功能的结果将直接辅助公司管理层对项目管理、项目计划等工作安排进行调整。所以这些数据应该非常准确并具备要求的精度。

6、 友好的用户交互界面。由于此系统的一大部分用户是公司的客户，而且本系统非常强调客户参与。友好的用户交互界面可以提高客户的满意程度以及对公司的好感度。所以友好的用户交互界面也是重要因素之一。

7、 可扩展性。项目管理的方法论是在不断的发展变化的，本系统有可能会随着方法论的扩展变化而不断的加入新功能，所以在不影响现有功能的同时能够容易地对系统进行扩充也至关重要。

8、 性能需求。在众多的用户交互和数据统计过程中，要注意系统性能的平衡。要优化消耗资源比较大的数据统计过程，使后台操作尽量不影响用户的操作交互，降低操作相应方面的延迟。通常来讲，一次用户数据的读取不应该超过 10 秒钟。

第3章 系统总体设计

3.1 系统设计目标和原则

系统设计师软件开发中的重要阶段。通过需求分析，我们已经得到了功能性需求和非功能性需求的列表和详细说明。在系统设计阶段，将在需求分析的基础上对系统的技术、逻辑、功能、数据、安全等方面做出概要设计。主要工作有根据实际情况划分系统架构层次，从用例中抽象出用于指导开发的业务领域模型，定义层与层之间、模型与模型之间的服务接口约束，解决系统数据持久化与数据交互问题，并为系统性能、安全、部署等问题提出可行的解决方案。本阶段的目标是得到一个可用的、健壮的、能满足开发需要的系统框架，为下一步的详细设计提供技术基础。

在系统设计时，需要遵循如下原则：

- 1、 标准化和开放性。本系统架构完全遵循通用的 JAVA EE 技术标准体系的要求，并使用业内通用的技术进行实现。该系统使用的基础服务平台、网络体系结构、网络通讯协议以及安全标准也符合国家和国际通用标准。高度的标准化和开放性使本系统更容易评测和扩展。
- 2、 先进性和成熟性。本系统充分利用业内得到承认以及实践证明的先进技术，综合运用当前先进的计算机网络技术、软件开发模型、以及网络安全技术来构建基于多层架构的分布式系统平台。
- 3、 适用性和健壮性。系统架构要能充分满足系统的功能性、非功能性需求，并能应对实际开发中产生的技术难题，为系统提供一个健壮的骨架。
- 4、 易用性和可扩展性。系统架构的目的之一就是要预见并解决系统中可能存在的技术难题，从而提高实际开发工作的效率。而随着开发活动的进行，系统功能的增加，系统架构也要易于扩展，来应对需求和技术的变化增长。
- 5、 可维护性和可重用性。系统架构灵活，易维护、可重用可有效提高生产率、降低成本和提高软件产品质量。提炼架构中的可重用模块也有利于对业务需求和领域模型的加深理解。

3.2 系统架构设计

3.2.1 系统技术架构总述

本系统采用 JAVA EE 体系结构作为系统的基础技术平台。

JAVA EE 是由 SUN 公司提出的，一种用来开发分布式企业软件应用系统的平台。随着企业应用和分布式应用的要求、标准的不断丰富与发展，SUN 公司联合了多家业界巨头，把所有与企业开发相关的标准、API 在 JAVA 平台上整合起来，构成了 JAVA EE 平台。

针对企业开发来讲，JAVA EE 平台具有很多优势：JAVA EE 为企业级运算的许多领域（比如数据库连接、企业业务组件、面向消息的中间件、Web 相关组件、通信协议以及互操作性）设立了标准；JAVA EE 是一种标准的开放平台，基于此开发的软件组件能够在不同厂商的产品中相互移植；JAVA EE 的系统平台一般都实现了许多底层架构和基础部分，因此大多数企业可以不再开发中间件，集中精力构建符合自己商业需要的应用；JAVA EE 增进了现存各种异构系统之间的互操作性。JAVA EE 技术平台的技术高度和成熟程度已经得到了业内的广泛确认^[6]。

与传统的 C/S 结构的软件系统的多层架构相类似，JAVA EE 技术平台为基于 B/S 结构的软件系统平台同样定义了符合 B/S 结构开发模式的多层架构规范。

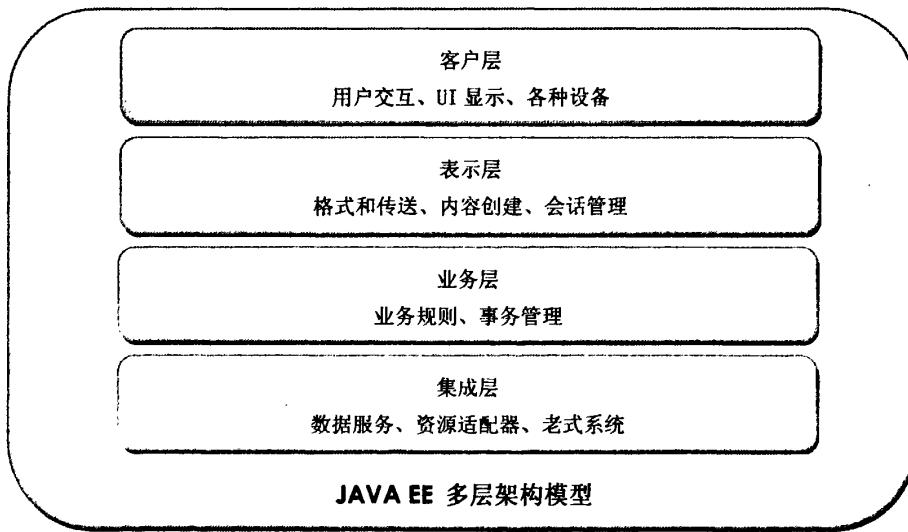


图 3-1 JAVA EE 应用程序的多层架构模型

如上图所示，JAVA EE 应用程序开发通常是一种四层模型。

客户层是用于处理访问系统或者 Web 应用程序的设备。这里通常也是出现用户界面 (UI) 的地方。这一层的组件例子包括 Web 浏览器、Java 小程序 (Applet) 等。

表示层是处理表示逻辑的地方，即处理从客户层传来服务请求采用的逻辑。许多客户服务功能都可以在表示层处理：管理会话、提供签名功能、提供对业务服务的访问，以及向客户层准备和发送响应信息。

业务层是由应用程序实时地请求或者提供业务服务的地方。基本的逻辑（有时也包括业务数据）都可以在这一层中表示出来。这一层拥有应用程序逻辑的丰富内容。

集成层（有时也成为企业信息系统层）处理的是外部资源。这些部分可能是老式应用程序，也可能是应用程序请求的各种数据存储。在集成层和业务层之间通常会有一种比较紧密的关系。当出现了数据或服务请求的时候，业务逻辑将使用集成层对它们进行访问。

此次开发的项目管理系统就是一个完整的基于 JAVA EE 平台的 B/S 结构的应用程序。目标程序将运行在一个实现了 JAVA EE 标准的容器内，并遵循各种 JAVA EE 标准的约束。

本系统根据实际需要归纳和简化了 JAVA EE 的四层结构模型，主要的结构层次与 JAVA EE 轻量级开发所常用的三层架构模式比较类似，分别是表现层、业务层、持久层。表现层用于产生用户界面、包装处理客户请求。业务层封装大量的服务接口和实现，通过依赖注入的方式管理各个服务对象之间的关系。通过对对象和对象关系表示，来确认针对每个具体业务的领域模型。持久层通过对对象关系映射机制将业务层封装好的持久化对象存入数据库或将数据库数据封装成对象返回给业务层。表现层通过服务定位器对业务层的相关服务进行请求和调用，业务层通过持久化处理代理来和持久层进行数据交互。

通过这些方式，整个系统被有机的粘合在一起，既相互依赖又具有足够的独立性。这种设计方式非常符合高内聚、低耦合的面向对象化设计原则。

另外，由于系统还要求有良好的用户交互界面，而传统的单纯的基于 HTML 的用户界面并不能充分满足这个要求。所以在系统架构中，我们还加入了近两年

兴起的 AJAX 和富客户端技术。

AJAX 是 Asynchronous JavaScript and XML 的缩写，即异步 JavaScript 和 XML 操作的缩写，具体是指通过客户端 JavaScript XML 请求对象异步地向服务器请求数据的技术。与此相关的技术有异步（Asynchronous）、XMLHttpRequest、JavaScript、CSS、DOM 等。现在 AJAX 这个技术名词的覆盖面有所扩展，把允许浏览器与服务器通信而无需刷新当前页面的技术都涵盖在内。AJAX 的技术关键在于异步请求和用户界面的无刷新改变^[7]。

富客户端技术是在近几年出现的通过扩展用户客户端访问程序（如浏览器）来增强界面表现效果，以带来更好的用户体验的技术。本系统中采用的富客户端技术有 DOM（动态文档模型）和 Flash 等。

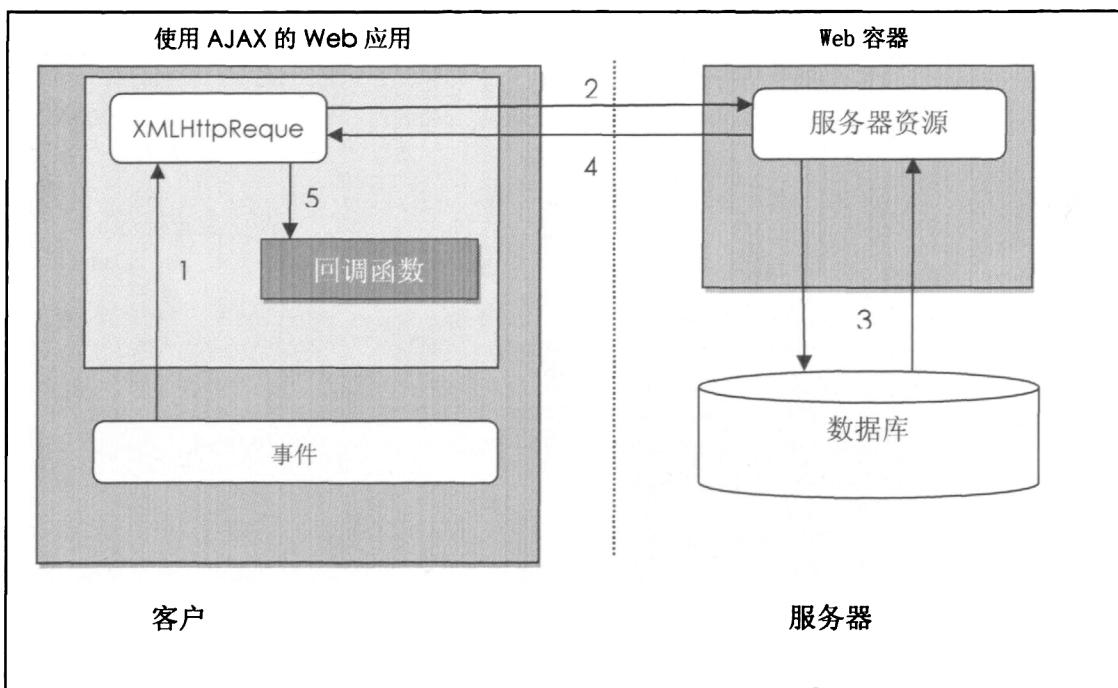


图 3-2 标准 AJAX 交互示意图

AJAX 和富客户端技术是相辅相成的。客户端界面程序通过触发界面事件请求 XMLHttpRequest 对象在不影响当前使用的正在进行动作的情况下向服务器动态请求服务器端资源。服务器在捕获 XMLHttpRequest 请求，进行处理后，将数据包装成标准格式（通常是 XML 数据格式）并返回。客户端 XMLHttpRequest 收到服务器返回的信息后，调用回调函数处理信息，并使用 DOM（文档对象模型）技术在不刷新整个页面的情况下对数据区域进行更新。

由此可见，使用 AJAX 和富客户端技术 Web 应用程序，相对于传统的 Web 应

用程序，客户端界面要承载更多的数据处理和页面处理工作。

所以在本系统中使用到 AJAX 和富客户端的地方，添加了一个客户界面层。客户界面层实现了 AJAX 的动态异步请求功能，并具备简单的数据处理能力，能动态调整用户界面布局和数据更新。客户界面层的请求发往表现层中特殊的请求包装器，由这种请求包装器对 AJAX 请求进行处理，然后再通过上述三层架构的处理方式向后台各层进行服务调用。

3.2.2 系统功能架构设计

通过系统需求分析阶段的分析，我们已经对系统应具备的功能有了清晰的认识。由于本系统功能的侧重点在项目信息管理方面，并没有太复杂的多环节业务流程，所以可以根据功能性需求分析的用例来划分系统功能模块，而这些模块也可以容易的映射成开发所用的领域模型。

依照需求分析，该项目管理系统主要分为后台信息管理和前端信息展示两大部分。其中，两个部分的用户登录登出功能所依赖的模块是基于权限和角色的验证系统，对这两个功能的请求会触发后台的验证系统。

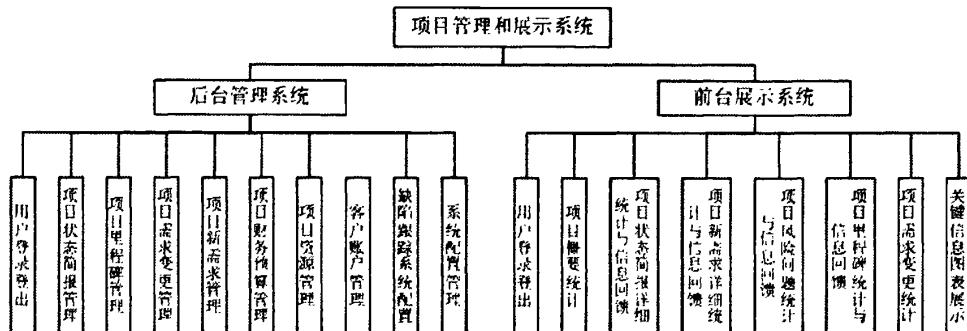


图 3-3 项目管理与展示系统功能架构图

上图具体功能模块细节可结合第二章表 2-2、表 2-3 以及相关用例进行理解。

3.2.3 系统逻辑架构设计

系统的逻辑架构是为了便于理解系统的结构与组织。其主要功能是确认系统中的业务实体及业务实体间的关系。这些业务实体就是现实业务需求映射到软件解决方案中的领域模型。

本系统架构逻辑视图如下：

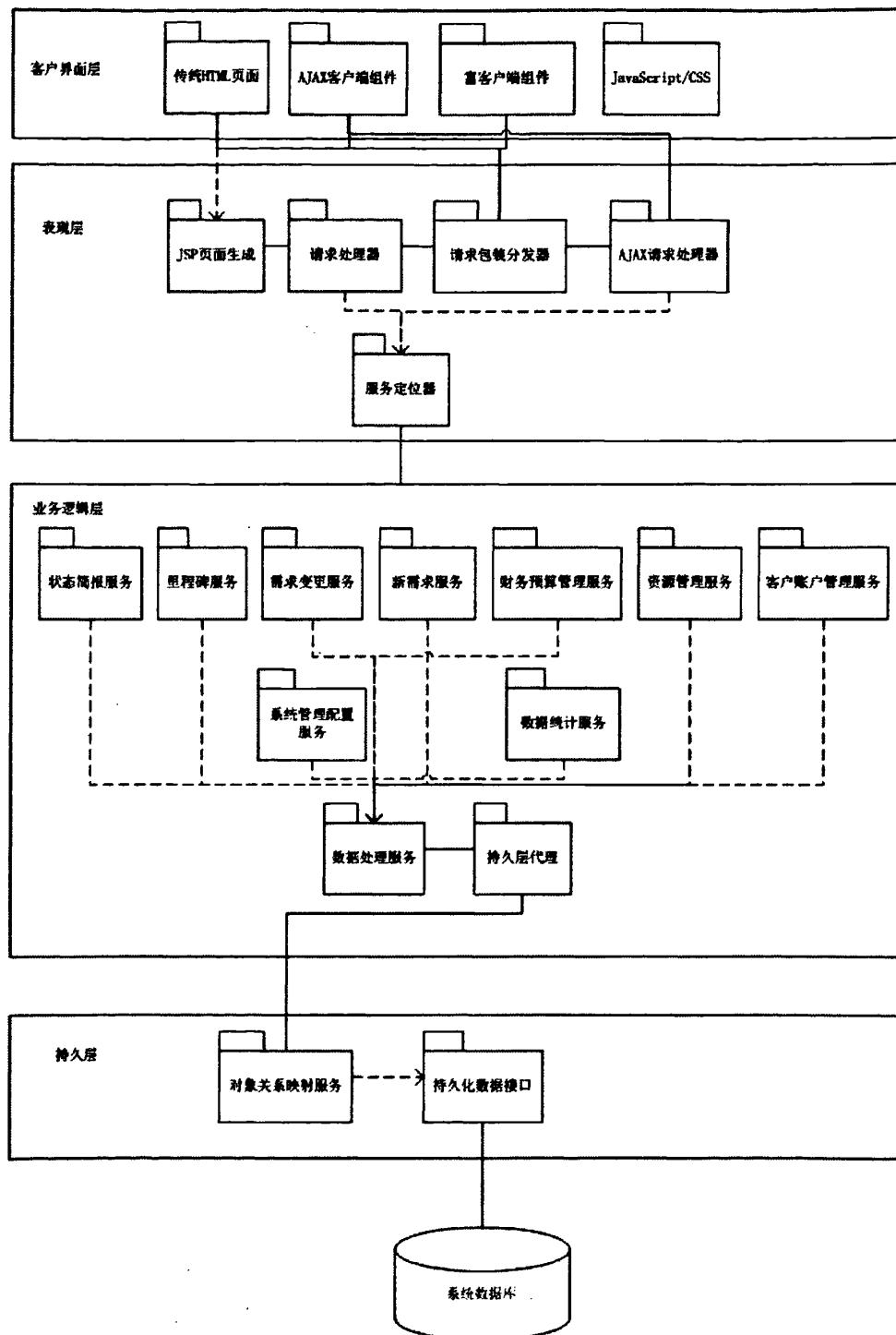


图 3-4 系统逻辑视图

逻辑视图说明：

1、 客户界面层。由于使用 AJAX 和富客户端组件的原因，原本功能单一的用户客户端被赋予了一定的数据处理和用户界面动态刷新与布局能力。所以在这个系统中，用户界面及其相关组件被单独列为一层。这一层中的 HTML 页面和 JavaScript、CSS 代表传统客户端界面组织形式。而 AJAX 和富客户端组件则代表具备数据处理能力的新技术用户界面。

2、 表现层。表现层的主要功能是处理客户终端发送过来的请求，处理请求并向业务层请求系统服务，等数据处理结束后根据 JSP 定义生成客户端响应并回送或回送 AJAX 数据。请求包装分发器相当于 JAVA EE 设计模式中的前端控制器，用于包装捕获到的用户请求，并根据用户请求动作的不同分发给不同的请求控制器。请求处理器与 AJAX 请求处理器是相应用户请求的实体，两者分别用于处理和回应传统 HTTP 用户请求和 AJAX 用户请求。服务定位器是表现层与业务逻辑层沟通的桥梁，表现层通过它向业务逻辑层请求服务并得到可以提供业务服务的对象实体^[8]。

3、 业务逻辑层。本层除了数据处理服务和持久层代理两个模块，其他模块都是和能解决具体问题的业务领域模型一一对应的。这些模块有可能是一个独立的服务提供者，也有可能是由众多的服务提供对象组合而成的。组合好的服务对象作为服务提供者供层内或直接上级层的其他模块调用。表现层和业务逻辑层所操作的主要数据对象是运行于业务逻辑中的数据对象，这种数据对象我们称为值对象，而实际用于持久化数据存储的对象是经过我们整合优化后的数据对象，这种对象我们称之为持久化对象。按照系统架构设计的规范，值对象是不应该直接交给持久层进行持久化操作，所以此层添加了一个数据处理服务模块，将值对象转化成持久化对象，并执行数据校验和转换。持久层代理是业务逻辑请求持久化操作的统一门户，用于简化持久层复杂多变的持久化操作。

4、 持久层。本系统使用对象关系映射框架来简化数据存取操作。对象关系映射服务就是指提供这一服务的持久化框架。这个服务通过与业务逻辑层的持久化代理模块结合，来对外界提供持久化服务。持久化服务接口是指符合 JAVA 标准的持久化服务接口，这里面包括常用的 JDBC 组件、JAVA 连接器结构、JAVA 消息服务等。

系统逻辑结构并不等于实际开发部署中的物理结构。系统逻辑结构的主要目

的是为系统各层面各模块的开发提供标准和规范。在实际开发中，要遵守逻辑视图中的设计约束，同时，对于逻辑结构中的缺陷和不足也要适度地修正与维护。

3.2.4 系统数据架构设计

由于本次开发是基于现有系统的二次开发，而且出于安全性和性能考虑，目标系统的后台管理平台和前端展示平台是物理上分离的两个应用。所以本系统的数据存取较普通的信息管理系统略微复杂，既有现有系统数据模型的改造，又有新数据模型的创建，还有跨应用、跨平台间的数据共享。

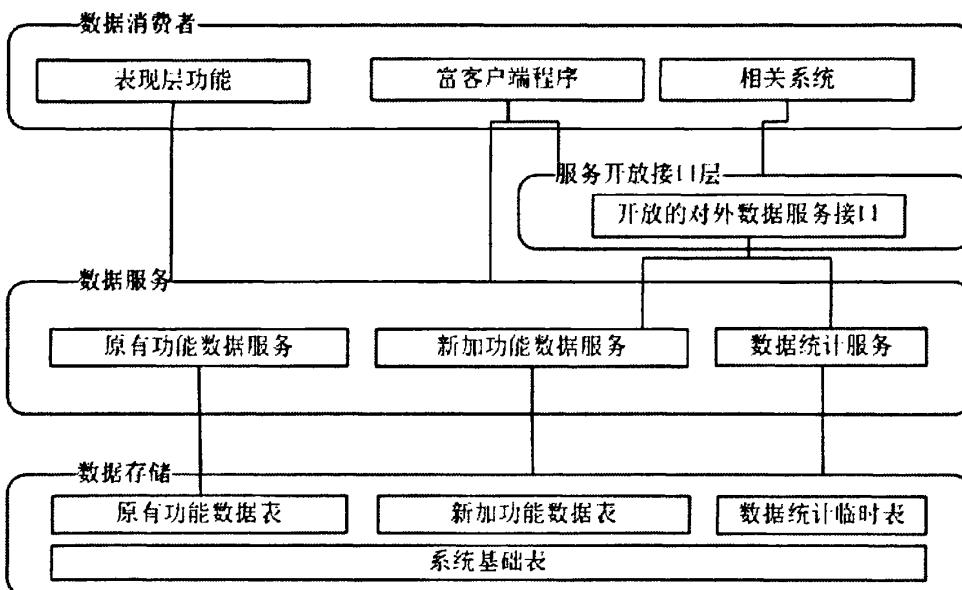


图 3-5 系统数据架构图

如上图所示，在物理数据存储方面，为了满足新功能的需要，系统在现有存储结构的基础之上，加入了新加功能表和数据统计临时表。新加功能表用于保存新功能要求的数据信息。大量的数据统计计算会降低系统效率，而且根据统计要求，系统需要具备保存某个时间点项目状态的能力。数据统计临时表就用于解决这两个问题，暂存系统统计信息和项目临时状态。

数据服务与逻辑视图中的持久层服务和业务逻辑层服务相对应，用于提供能满足业务需要的数据存取、整合、转换、校验等数据服务。在现有功能数据服务的基础上，根据已有的领域概念模型，加入新加功能数据服务和数据统计服务。

为满足后台管理平台和前端显示平台的数据交互、网络应用平台和富客户端

程序的数据交互、项目管理系统和公司其他系统的数据交互，在数据服务的基础上，对外开放一个服务接口层。服务接口层通过能用于网络传输的数据接口定义，向外部传输规定格式的数据，从而实现跨平台、跨应用直接的数据传输。这种网络服务调用技术的范例有 Web Service、基于 HTTP 的远程方法调用等。

该系统预计的数据服务消费者有用于处理和相应用户请求的表现层功能模块、具备数据处理能力的富客户端程序及其其他预计的数据相关系统。

3.2.5 系统安全架构设计

根据非功能性需求规定，该系统平台的需要具备一定的安全性。涉及安全的方面主要有网络传输的数据安全性，以及用户访问的有效性确认。

为实现网络数据传输的加密，本系统决定采用国际网络安全数据传输协议之一的 HTTPS 协议作为系统网络访问的基本协议。该协议符合国际标准，而且易于实现。

在用户访问的有效性确认和控制方面，系统依据 JAVA EE 规范要求，加入了用于拦截过滤用户访问的网络请求过滤模块，来保证真正数据操作的安全性。

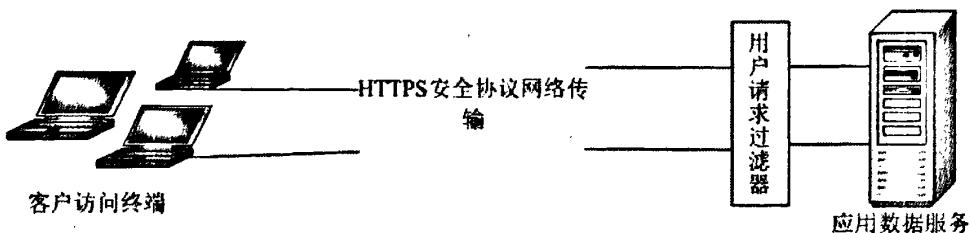


图 3-6 系统安全架构图

第4章 系统详细设计

4.1 概述

系统详细设计的目的是在系统总体设计的基础上，对系统中的逻辑对象、数据实体和界面逻辑关系进行进一步整理和细化，确定参与系统活动的类和对象，并对这些类和对象做出定义。系统详细设计所要考虑的主要问题是如何将能实现功能需求的业务领域模型用可进行实际开发设计的具体类和对象组合表示出来。

根据统一软件开发过程（UP）和统一建模语言（UML）的规定，系统详细设计主要包括业务领域模型在静态模型和动态模型方面的设计。其中静态模型方面的设计主要使用类图进行表示，针对数据库方面的设计可以用实体关系图表示。动态模型可以使用交互图等进行表示。

系统详细设计所要涉及到的基本概念有：

领域模型与领域模型模式：领域模型是应用程序问题域的对象模型，是概念、类或者感兴趣领域的现实对象的可视化表示。领域模型模式采用良好的老式面向对象分析与技术实现整个业务逻辑，它既是问题域的描述优势业务逻辑设计的蓝图^[9]。此次开发的分析设计都是依照领域模型模式进行的。

类图。统一建模语言（UML）中表示类的属性、行为以及类与类之间关系的图形模式^[10]。

交互图。统一建模语言（UML）中用于展示对象之间如何通过消息交互的图形模式。交互图可以用协作图和时序图两种图来表现。

POJO。POJO 是指 Plain Old Java Object，就是指不实现任何特殊接口的简单 Java 对象，是“无侵入”特性的代名词。无侵入可以使用户的业务逻辑代码不依赖于具体应用框架代码。由于本系统是基于 JAVA EE 轻量级开发平台的，所以本章内所涉及的所有 JAVA 对象都是 POJO 对象，不会有任何接口或平台约束。POJO 对象的生成、包装、传递等操作有时可能会借助于外部框架来实现。

本章的主要内容是针对后台管理平台、前端展示平台、开放服务接口等三个方面阐述系统的详细设计内容。阐述方式基于领域对象模型模式。由于系统功能

点众多，而且各功能点的特点、类型相似，所以在描述过程中，每方面只选取具有代表性的领域模型进行解释。

4.2 后台管理平台详细设计

4.2.1 后台管理平台静态模型

后台管理平台的主要任务是根据用户操作，捕获用户输入数据处理，并将用户数据根据相关的业务流程存入数据库。此后台管理平台所要处理的用户操作类别众多、业务流程众多，由于版面显示，先给出该平台的抽象设计类图：

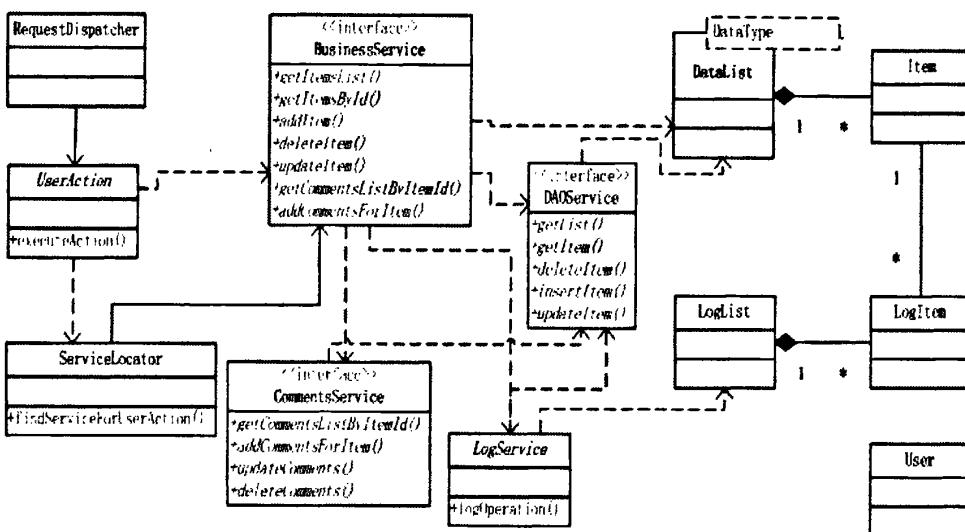


图 4-1 后台管理平台抽象类图

在此图中，对后台管理平台的各主要实现部分用抽象类和接口的方式进行了约束和概括。

此图中，`RequestDispatcher` 是具体类，用于捕获用户请求并将请求进行包装，同时将请求分发给具体用户操作类。

`UserAction` 是处理用户操作的抽象类，其中 `executeAction` 是关键的抽象方法。在开发过程中，新添加的针对某个用户操作的具体类需要继承此类，并实现 `executeAction` 方法。

`ServiceLocator` 是具体类，用于在用户操作类的实现子类进行业务操作和

数据存取的时候向后台请求可以满足服务的服务类对象。

`BusinessService` 是可以提供业务逻辑操作的接口，为各种业务逻辑提供基本的实现要求和约束。由于本系统的管理模块众多，所以 `BusinessService` 在这里所代表的其实是各种业务逻辑接口的集合。

`CommentsService` 是为各管理模块数据提供注释服务的接口，规定的各种注释服务的基本操作。

`LogService` 是用于记录用户动作的抽象类。在开发过程中，具体实现此功能的类需要集成它并实现其关键方法 `logOperation`。

`DAOService` 是提供持久化数据存储的服务接口，对数据库操作进行了确定。`DataType` 是用与传输数据容器型的泛型类，其中 `DataType` 是指在实际过程中该类所包含的数据类型。

`LogList` 是指携带用户操作历史记录的容器型具体类。

`Item` 是指具体的数据类。在这里 `Item` 实际代表指后台各种数据类型的集合，包括简报、里程碑、需求变更等具体数据类型。

`LogItem` 是代表操作历史记录的每一项的数据类。它和 `Item` 是多对一的关系，即一条数据记录可以有多条历史操作记录与之对应。

`User` 是代表参与业务活动的用户的数据类。

上图明确了系统各模块各功能直接的抽象实现约束，下面将结合项目状态简报管理模块的具体类图对上图进行示例说明：

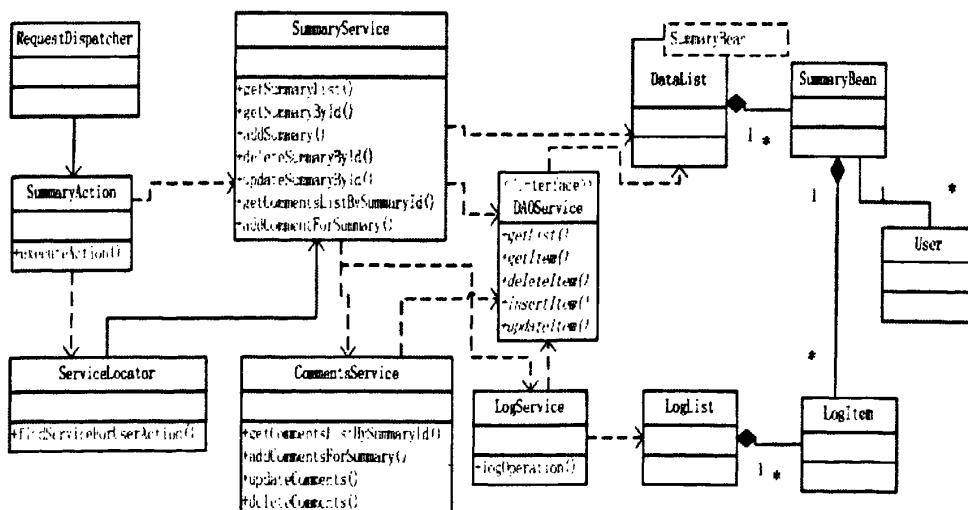


图 4-2 项目状态简报管理类图

图 4-2 中的 SummaryAction、SummaryService、CommentsService 等具体类是分别是图 4-1 中的对应部分的实现。关于各类的功能与责任可以参照图 4-1 的说明进行理解。其中 DAOService 接口在本图中仍未实现，是因为本系统开发中，具体数据库操作的实现是由外部对象关系映射框架完成的。

4.2.2 后台管理平台动态模型

后台管理系统对用户操作的处理方式是这样的：用户通过浏览器点选操作，向系统发出请求；系统以 RequestDispatcher 对象作为处理请求的第一步，RequestDispatcher 需要对请求进行验证和包装，创建相应用户请求的 UserAction 子类对象，并调用其 executeAction 方法；UserAction 的子类对象对用户发送过来的请求和数据进行第一步处理，然后根据业务逻辑的要求，向 ServiceLocator 的实例对象请求能提供相关业务逻辑的 BusinessService 对象；ServiceLocator 根据 UserAction 子类对象指定的服务内容，查找并创建对应的 BusinessService 服务对象，同时创建与 BusinessService 有依赖关系的其他服务对象，并处理这些对象之间的依赖关系，最后将装配好的 BusinessService 对象实例返回给 UserAction 子类对象；UserAction 子类对象在得到 BusinessService 对象实例后，调用接口约束好的服务方法，在服务层对数据进行进一步处理；BusinessService 对象实例是根据业务逻辑处理数据的核心，在处理数据的同时 BusinessService 对象还有可能会调用其他相关服务的对象实例对数据进行同步处理，如 LogService，DAOService 等。在 BusinessService 对象处理完数据并将结果返回给 UserAction 之后，UserAction 对结果数据进行包装处理，并直接返回给用户客户端。

与此工作流程相对应的协作图如下：

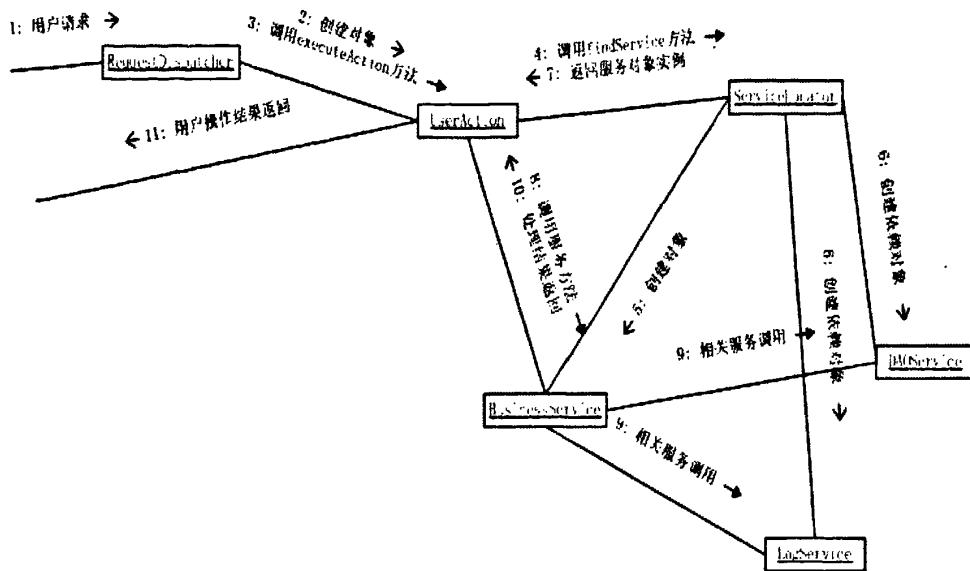


图 4-3 后台管理平台协作图

为进一步说明系统中对象与对象之间的交互关系，下面以添加项目状态简报的操作为例，使用时序图进行进一步说明：

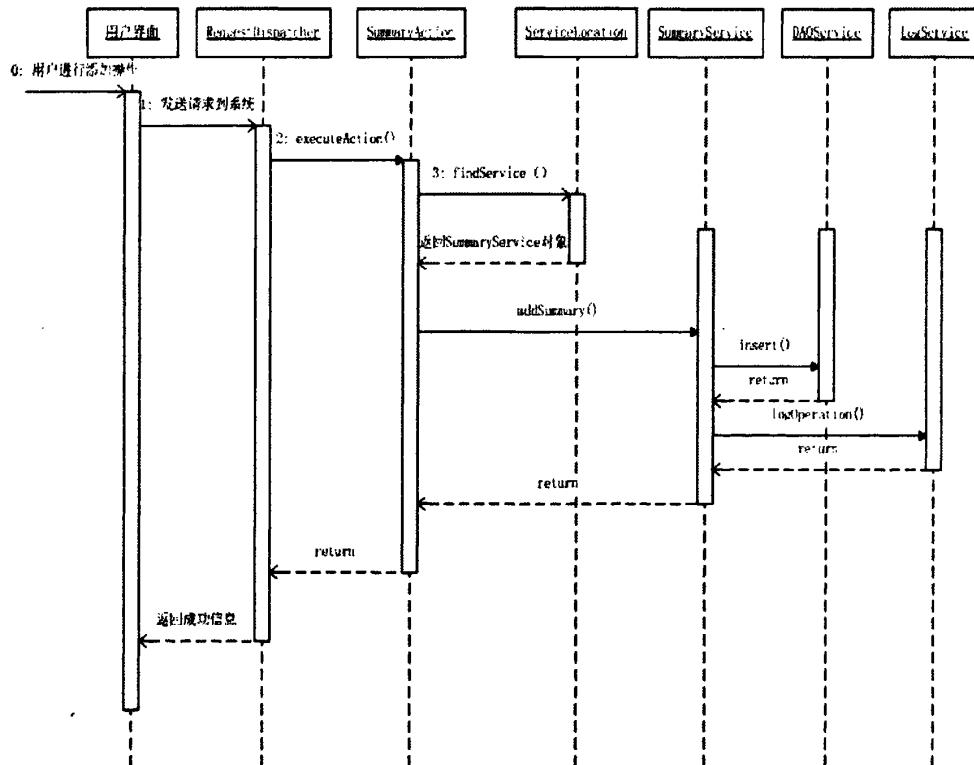


图 4-4 项目状态简报添加场景时序图

根据统一建模语言对协作图和时序图的定义，两者同属交互图，都是用于表

示系统对象之间的动态交互，并且两种图可以相互转换，所以图 4-3 和图 4-4 可以结合理解。

4.3 前端展示平台详细设计

4.3.1 前端展示平台静态模型

前端展示平台的设计类图如图 4-5 所示：

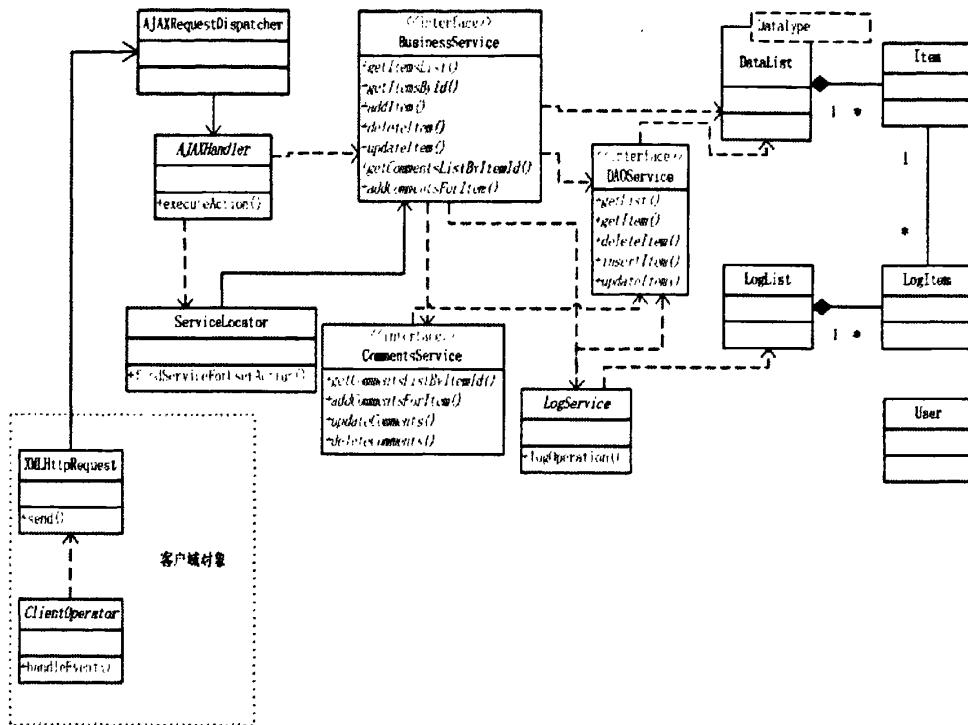


图 4-5 前端展示平台抽象类图

比较图 4-5 与图 4-2 可以看出，前端展示平台的类关系结构与后台管理平台的类关系结构非常相似，只是在原有类结构的基础上，对 **RequestDispatcher** 和 **UserAction** 根据 **AJAX** 类型请求的特殊要求做了修改，分别用 **AJAXRequestDispatcher** 和 **AJAXHandler** 进行代替。同时加入了客户域类。

由于前端展示平台对 **AJAX** 技术的引入，客户端用户界面就不再仅仅具有信息展示的能力，而被赋予更多的功能要求，如客户数据收集、校验、运算，不依赖于浏览器表单提交的异步数据请求，局部页面刷新和排版等等。所以在实际开

发过程中，需要针对 AJAX 技术需要，加入对客户端领域对象和类的设计，用于处理位于客户端操作和计算^[11]。

客户端领域的类主要有两个，一个是自定义的抽象类 ClientOperator，用于捕获并处理用户在客户端的操作和相关数据，另一个是 XMLHttpRequest 类。这个对象是有 Web 浏览器的标准定义并由客户使用的浏览器来实现的，用于在客户端和服务器之间建立一个异步地请求通道传输数据。这两个类都依赖于客户所使用的浏览器，在当前技术条件下，大多数都是用浏览器标准定义的 JavaScript 语言平台实现^[12]。

图 4-5 其他类和类关系与图 4-2 非常相似，可以参照理解。

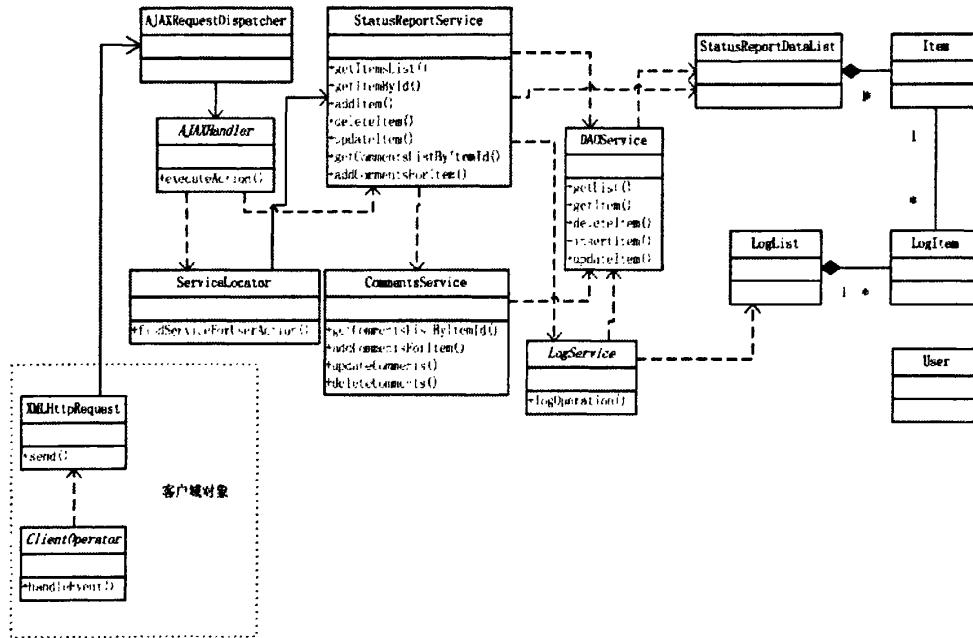


图 4-6 前端展示平台项目简报浏览功能类图

图 4-6 是对前端展示平台项目简报部分的具体类图，是图 4-5 抽象类图在项目简报部分的具体实现。其中 StatusReportService、CommentsService、DAOService、StatusReportDataList 分别是针对图 4-5 中 BusinessService、CommentsService、DAOService、DataList 等接口和泛型类的具体实现。

4.3.2 前端展示平台动态模型

由于版面限制，前端展示平台的动态模型采用协作图进行展示。见图 4-7：

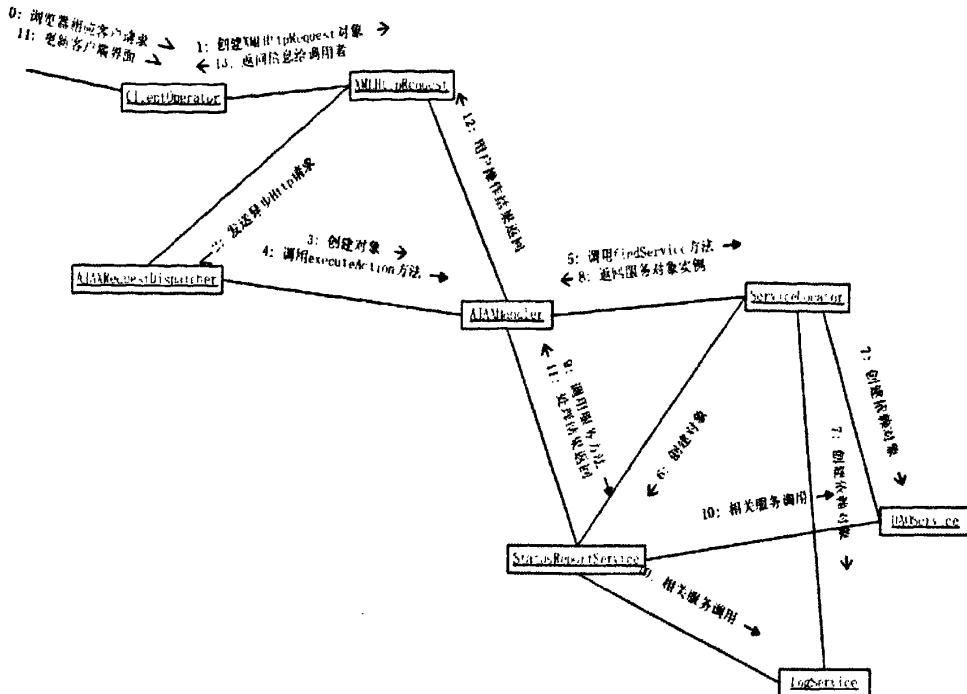


图 4-7 前端展示平台项目简报浏览功能协作图

在图 4-7 中, ClientOperator 和 XMLHttpRequest 都是客户端对象, 运行于用户浏览器的环境下, 而其他对象都是运行于服务器端的对象。ClientOperator 和 XMLHttpRequest 大多数都被实现为 JavaScript 对象。其中 ClientOperator 对象是在客户端具备交互功能的事件被触发的时候由浏览器自动创建并提供服务。而 XMLHttpRequest 对象是 ClientOperator 根据浏览器的具体实现方式不同, 而动态创建的浏览器内置支持的对象。

其他位于服务器端的对象所具有的职责和相互关系, 与图 4-3 中的表示基本一致, 可参照理解。其中 StatusReportService 类是对图 4-3 中 BusinessService 接口在项目简报浏览功能的具体实现类。

4.4 开放服务接口设计

如第 3 章系统数据架构部分所叙述, 本次开发的数据结构略微复杂, 既有新开发系统模块与原有系统模块之间的信息共享, 也有物理上分离的管理平台和展示平台的信息共享, 还有富客户端与整个系统服务间得系统共享。为实现这些跨应用、跨平台之间的信息共享, 系统设计了三种远程信息共享的方式来对外开放

服务接口。

HTTP 资源请求。根据 HTTP 协议的特性，客户端通过实现 HTTP 协议的工具或类库对已知服务器地址的 HTTP 资源进行请求，服务器返回富文本结构的数据。通过这种特性，完全可以通过暴露服务器资源地址以及定义返回数据格式的方式使两个系统通过网络进行交互。这种方式在本系统中主要用在富客户端和系统之间的数据交互。调用方式主要是 JavaScript 对象通过 XMLHttpRequest 向已知的服务器资源 URL 发送 POST。请求返回信息格式以 XML 和 JSON(JavaScript Object Notation, JavaScript 对象注解)为主。

Web Service。Web Service 是为了统一各种平台的远程数据交换和远程方法调用而定义的以 XML 为载体的网络服务标准。它定义了一套客户端请求封装、信息传输、服务器端处理和相应的标准。Web Service 的信息传输完全以 XML 为基础。在本系统中，这种方式主要用户系统向外部潜在的数据消费端提供服务。调用方式是通过 JAVA Web Service 的工具类库对请求进行包装，然后发送到已知的 Web Service 地址，最后接收 XML 格式的回应数据并解析^[13]。

远程方法调用。远程方法调用时比较原始的远程数据共享方式，但也是最直观的共享方式。其主要实现原理是在本地创建一个能满足服务票根对象，通过远程方法调用框架的封装，与服务器端的真实提供服务的对象进行匹配，并获取真实服务对象的服务能力。中间的数据交互通过系统平台提供的序列化和反序列化实现。在本系统中，这种方式被普遍用于物理分离的后台管理平台和前端展示平台之间，新开发模块和原因模块之间的数据共享。

由于第一、第二种方式比较普遍，而第三种方式的实现方式更具代表性，所以本文将在详细设计和系统实现部分对第三种方式进行详细说明。其他两种则不再做说明。

4.4.1 开放服务接口静态模型

远程方法调用的类图如下：

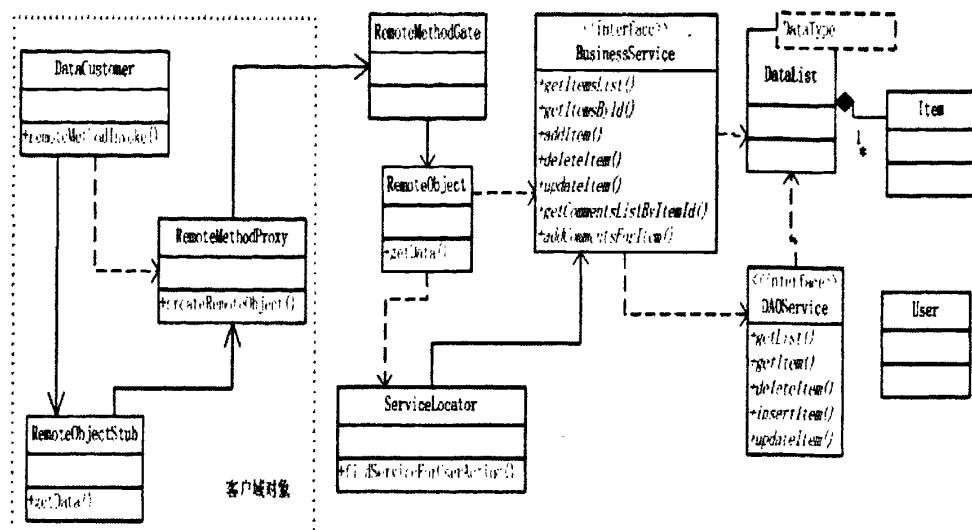


图 4-8 远程方法调用示例类图

图 4-8 中，`DataCustomer` 是指远程端需要远程方法调用的类。`RemoteMethodProxy` 和 `RemoteMethodGate` 是实现远程方法调用的关键类，一般是由远程方法调用框架实现的。`RemoteMethodProxy` 用于创建本地票根对象类 `RemoteObjectStub` 的实例，并向 `RemoteMethodGate` 发出服务请求。`RemoteMethodGate` 捕获到远程请求，创建真正的服务类 `RemoteObject` 的对象实例。`RemoteObjectStub` 对象和 `RemoteObject` 对象一一对应的，分别是远程端和服务器端的服务代表，两者应该是同一个服务接口的实例。

图中的 `ServiceLocator` 等类与图 4-2 中的类含义相同，可以参照理解。

该部分功能主要应用于新系统与已有系统交互，下图是新系统读取已有系统中项目信息内容功能的具体类图。

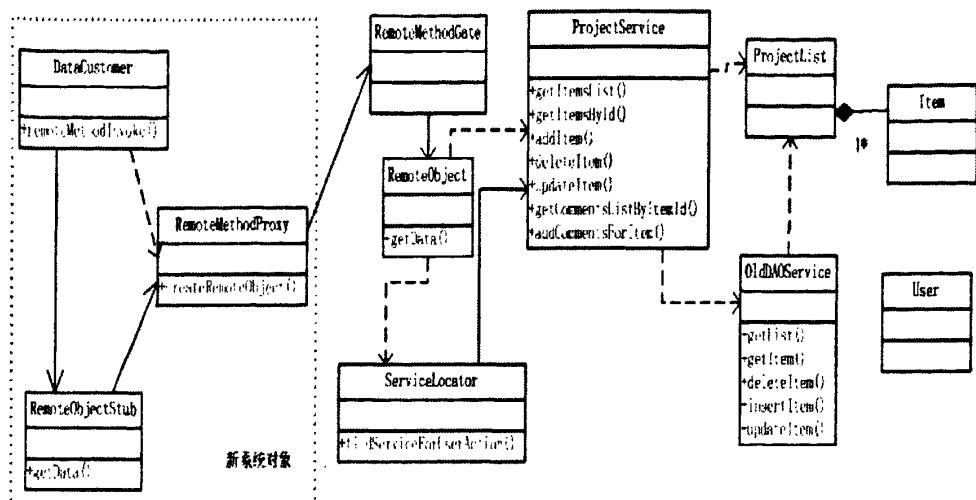


图 4-9 新系统读取已有系统功能类图

图 4-9 中 ProjectService、OldDAOService、ProjectList 分别是对图 4-8 中 BusinessService、DAOService、DataList 等接口和泛型类的具体实现，这些类的职能以及和其他类的关系可以与图 4-8 参照理解。

4.2.2 开放服务接口动态模型

以读取已有系统项目信息为例，远程方法调用的示例时序图如下：

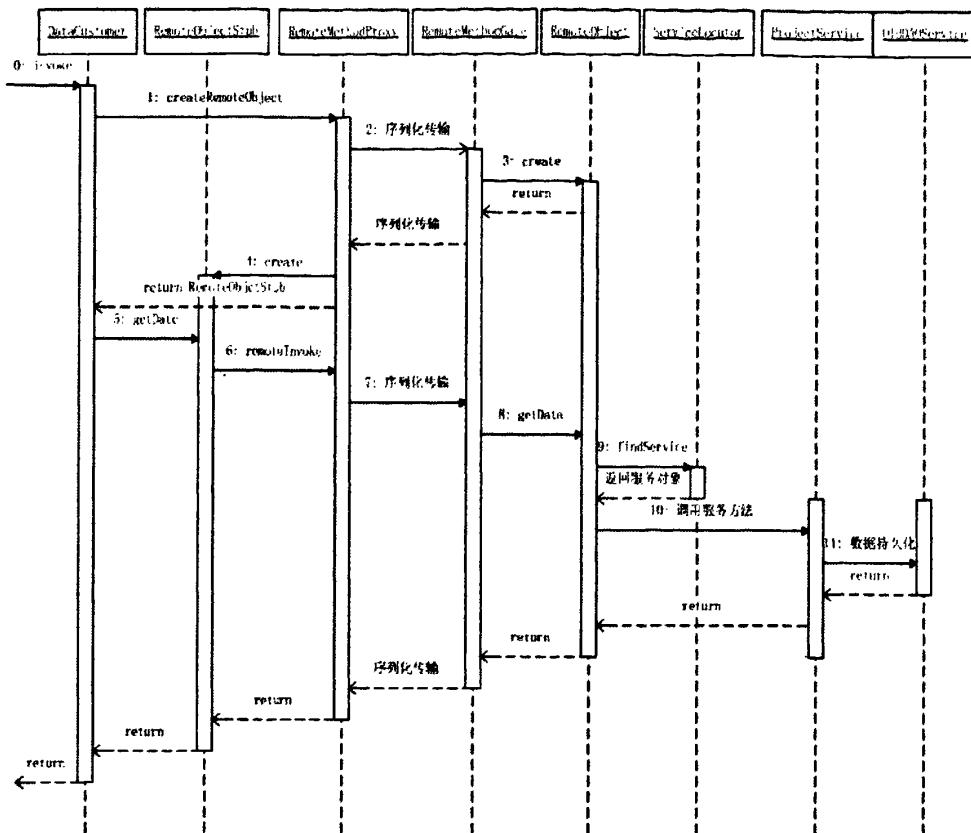


图 4-10 远程方法调用时序图

4.5 数据库设计

通过系统逻辑设计、数据架构设计、系统各模块的领域模型设计等活动，系统的每一个具体类已经有了大体信息。下面所要做的就是要完成数据持久层的设计。

本系统使用关系数据库作为持久层的实现。本节的工作主要是对各领域模型的持久化对象进行详细设计，明确这些对象的字段、字段类型等信息，并将其映射到关系数据库中，用数据库表的形式表现出来。

根据对各持久化对象的分类，本系统的数据库表大体可以分为如下几类：

业务数据表。与业务需求中的各种实际数据相匹配，保存各种数据实体的信息，如项目状态简报、里程碑、风险等

系统配置表。保存系统的配置信息

辅助数据表。保存一些辅助功能的数据，如操作日志等

统计信息中间表。用于暂存统计功能所需要的中间数据，如某月统计的项目工时消耗等，这些表中的数据会随着新统计数据的产生或期限过期而被覆盖或删除

其中系统配置表和统计信息中间表与原有系统关系密切，所以在本文中不再做专门叙述。

系统主要数据库表结构如下：

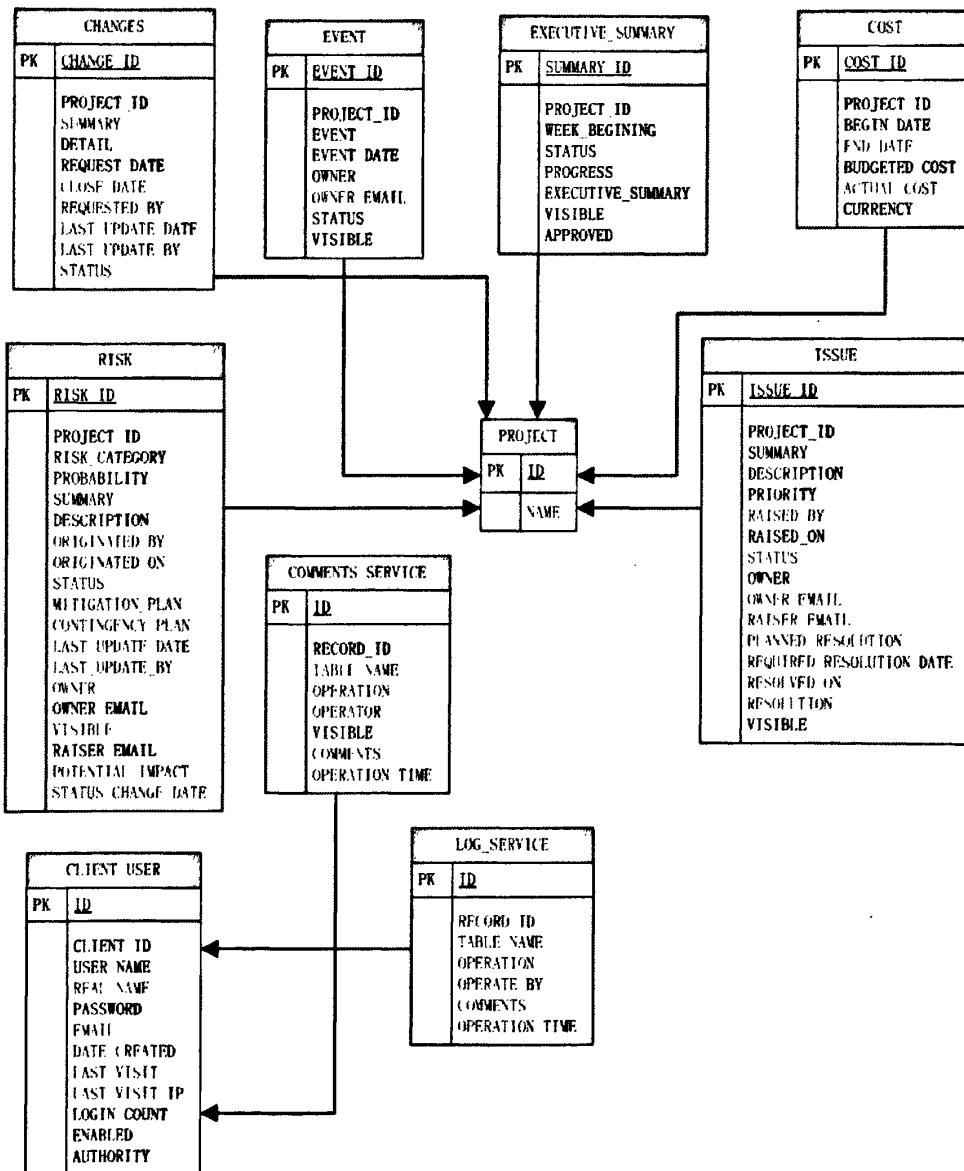


图 4-11 数据库主要表结构示意图

针对图中较为重要的表对其具体信息进行详细描述,说明表的名称、字段名、类型、长度以及意义等信息。

表 4-1 用于记录需求变更的表 CHANGES 结构

字段	说明	数据类型	长度	备注
CHANGE_ID	需求变更 ID	INTEGER		主键
PROJECT_ID	相关项目的 ID 号	INTEGER		外连接到 PROJECT 表 ID 字段
SUMMARY	需求变更简介	TEXT		
DETAIL	需求变更细节	TEXT		
REQUEST_DATE	变更请求日期	DATE		
CLOSE_DATE	变更关闭日期	DATE		
REQUESTED_BY	变更请求人姓名	VARCHAR	64	
LAST_UPDATE_DATE	最后更新日期	DATE		
LAST_UPDATE_BY	最后更新人姓名	VARCHAR	64	
STATUS	当前状态	INTEGER		外连接到 STATUS_DICT 表 ID 字段

表 4-2 用于记录客户账户和信息的 CLIENT_USER 表结构

字段	说明	数据类型	长度	备注
ID	客户 ID 号	INTEGER		主键
CLIENT_ID	相关客户信息记录的 ID 号	INTEGER		外连接到 CLIENT 表 ID 字段
USER_NAME	用于登录的客户名称	VARCHAR	64	
REAL_NAME	客户真实名词	VARCHAR	64	
PASSWORD	帐户密码	VARCHAR	64	
EMAIL	客户电子邮件	VARCHAR	128	
DATE_CREATED	账户创建日期	DATETIME		
LAST_VISIT	最后访问日期	DATETIME		
LAST_VISIT_IP	最后访问的 IP 地址	VARCHAR	15	
LOGIN_COUNT	访问次数总计	INTEGER		
ENABLED	是否启用	BIT		0=FALSE, 1=TRUE
AUTHORITY	权限标识	VARCHAR	11	

表 4-3 用于记录财务预算的 COST 表

字段	说明	数据类型	长度	备注
COST_ID	记录 ID	INTEGER		主键
PROJECT_ID	相关项目的 ID 号	INTEGER		外连接到 PROJECT 表的 ID 字段
BEGIN_DATE	预算开始日期	DATE		
END_DATE	预算截至日期	DATE		
BUDGETED_COST	预算金额	INTEGER		
ACTUAL_COST	实际花费	INTEGER		
CURRENCY	预算币种			

表 4-4 用于记录用户回馈信息的 COMMENTS_SERVICE 表

字段	说明	数据类型	长度	备注
ID	记录 ID 号	INTEGER		主键
RECORD_ID	回馈针对的记录号	INTEGER		
TABLE_NAME	回馈针对的记录类型	VARCHAR	32	如 EVENT、CHANGES 等
OPERATION	回馈操作类型	VARCHAR	256	
OPERATOR	添加回馈人的 ID	INTEGER		外连接到 CLIENT_USER 表 ID 字段
VISIBLE	回馈信息是否可见	BIT		0=FALSE, 1=TRUE
COMMENTS	回馈信息内容	TEXT		
OPERATION_TIME	回馈添加时间	DATETIME		

表 4-5 用于保存里程碑信息的 EVENT 表

字段	说明	数据类型	长度	备注
EVENT_ID	记录 ID 号	INTEGER		主键
PROJECT_ID	相关项目的 ID 号	INTEGER		外连接到 PROJECT 表 ID 字段
EVENT	里程碑信息	TEXT		
EVENT_DATE	里程碑事件	DATE		
OWNER	里程碑负责人姓名	VARCHAR	64	
STATUS	里程碑状态	INTEGER		外连接到 STATUS_DICT 表 ID 字段
VISIBLE	内容是否外部可见	BIT		0=FALSE, 1=TRUE
OWNER_EMAIL	负责人电子邮件	VARCHAR	128	

表 4-6 用于保存项目状态简报的 EXECUTIVE_SUMMARY 表

字段	说明	数据类型	长度	备注
SUMMARY_ID	记录 ID	INTEGER		主键
PROJECT_ID	相关项目的 ID 号	INTEGER		外连接到 PROJECT 表 ID 字段
WEEK_BEGINING	简报对应的星期第一天	DATE		
STATUS	简报状态	INTEGER		外连接到 STATUS_DICT 表 ID 字段
PROGRESS	简报汇报的项目进度	INTEGER		代表进度的百分比
EXECUTIVE_SUMMARY	项目简报内容	TEXT		
VISIBLE	是否外部可见	BIT		0=FALSE, 1=TRUE
APPROVED	是否已经过审阅	BIT		0=FALSE, 1=TRUE

表 4-7 用于保存项目新需求的 ISSUE 表

字段	说明	数据类型	长度	备注
ISSUE_ID	记录 ID	INTEGER		主键
PROJECT_ID	相关项目的 ID 号	INTEGER		外连接到 PROJECT 表 ID 字段
SUMMARY	新需求简介	TEXT		
DESCRIPTION	新需求详细描述	TEXT		
PRIORITY	优先级	INTEGER		
RAISED_BY	提出者姓名	VARCHAR	64	
RAISED_ON	提出时间	DATE		
STATUS	当前状态	INTEGER		外连接到 STATUS_DICT 表 ID 字段
OWNER	负责人姓名	VARCHAR	64	
OWNER_EMAIL	负责人邮件	VARCHAR	128	
RAISER_EMAIL	提出者邮件	VARCHAR	128	
PLANNED_RESOLUTION	计划解决方案	TEXT		
REQUIRED_RESOLUTION_DATE	要求完成日期	DATE		
RESOLVED_ON	实际完成日期	DATE		
RESOLUTION	实际解决方案	TEXT		
VISIBLE	外部是否可见	BIT		0=FALSE, 1=TRUE

表 4-8 记录项目风险问题的 RISK 表

字段	说明	数据类型	长度	备注
RISK_ID	记录 ID	INTEGER		主键
PROJECT_ID	相关项目的 ID 号	INTEGER		外连接到 PROJECT 表 ID 字段
RISK_CATEGORY	风险类别	INTEGER		外连接到 STATUS_DICT 表 ID 字段
PROBABILITY	出现概率	INTEGER		
SUMMARY	问题概述	TEXT		
DESCRIPTION	问题详细描述	TEXT		
ORIGINATED_BY	发现人姓名	VARCHAR	64	
ORIGINATED_ON	发现时间	DATE		
STATUS	问题状态	INTEGER		外连接到 STATUS_DICT 表 ID 字段
MITIGATION_PLAN	缓解计划	TEXT		
CONTINGENCY_PLAN	应变安排	TEXT		
LAST_UPDATE_DATE	最后更新时间	DATETIME		
LAST_UPDATE_BY	最后更新人姓名	VARCHAR	64	
OWNER	负责人姓名	VARCHAR	64	
OWNER_EMAIL	负责人邮件地址	VARCHAR	128	
VISIBLE	外部是否可见	BIT		0=FALSE, 1=TRUE

表 4-8 记录项目风险问题的 RISK 表

续表

字段	说明	数据类型	长度	备注
RAISER_EMAIL	发现人邮件	VARCHAR	128	
POTENTIAL_IMPACT	潜在影响	TEXT		
STATUS_CHANGE_DATE	状态改变时间	DATETIME		

表 4-8 用于记录操作日志的 LOG_SERVICE 表

字段	说明	数据类型	长度	备注
ID	记录 ID	INTEGER		主键
RECORD_ID	操作针对的数据 ID 号	INTEGER		
TABLE_NAME	操作针对的数据类型	VARCHAR	32	如 EVENT、RISK 等
OPERATION	操作内容	VARCHAR	256	
OPERATE_BY	操作人	INTEGER		外连接到 CLIENT_USER 表 ID 字段
COMMENTS	操作说明	TEXT		
OPERATION_TIME	操作时间	DATETIME		

第 5 章 系统实现

5.1 用户界面实现

JAVA EE 中支持表现层的是 Servlet 和 JSP。Servlet 是 Java 平台上的 CGI 技术。Servlet 在服务器端运行，动态地生成 Web 页面。与传统的 CGI 和许多其它类似 CGI 的技术相比，Java Servlet 具有更高的效率并更容易使用。对于 Servlet，重复的请求不会导致同一程序的多次转载，它是依靠线程的方式来支持并发访问的。JSP(Java Server Page)是一种实现普通静态 HTML 和动态页面输出混合编码的技术。借助形式上的内容和外观表现的分离，Web 页面制作的任务可以比较方便地划分给页面设计人员和程序员，并方便地通过 JSP 来合成。在运行时态，JSP 将会被首先转换成 Servlet，并以 Servlet 的形态编译运行，因此它的效率和功能与 Servlet 相比没有差别，一样具有很高的效率^[14]。图 5-1、5-2 就是使用 JSP 页面实现的后台管理平台的用户界面实例：

The screenshot shows a web-based project management application interface. On the left, there is a vertical sidebar with a list of navigation links: Status Summary, Key Events, Change Request, Risk Management, Issue Management, Cost Management, Resource Planning, Client Users, Change Password, Exchange Rate, Nantics Configuration, Dictionary, Parameters, and Time Report. The main content area has a header "Status Summary" and a sub-header "Add New Status Summary". Below this are several input fields and dropdown menus: "Client" (dropdown menu), "Project" (dropdown menu), "Sub-Project" (dropdown menu), "From" (text input: Jan 01, 2008), "To" (text input: Feb 29, 2008), a "Search" button, and a "Show Progress Chart" button.

图 5-1 项目状态后台管理模块实例界面

图 5-2 项目状态简报添加实例界面

根据项目需求约束，前端管理系统采用富客户端的界面形式来进行数据展示。本系统的富客户端技术是建立在 JavaScript 页面组件的基础上的。浏览器在对 HTML 页面进行展示的时候，会根据 HTML 源文件中的 JavaScript 和 CSS（Cascade Style Sheets，层叠式样表）对页面元素进行渲染修饰。利用这种特性，加以复杂的 CSS 和 JavaScript 代码辅助，可以是传统的 HTML 页面更具表现力，达到甚至超越传统桌面系统给用户的使用体验。本系统在数据展示部分还使用 FLASH 组件实现动态效果。图 5-3、图 5-4、图 5-5 就是使用富客户端技术的页面实例：

Client Logo

Logout

This screenshot shows a static web page for a project titled "Client For Demo". The page has a left sidebar with a tree view of projects, and a main content area with several tabs: Overview, Executive Summary, Issues, Risks, Key Events, Change Requests, and Graphs. The Executive Summary tab is active, displaying a week beginning on Oct 28, 2007, with overall status "Good" and progress "65%". It lists four items under "Change Requests" and two under "Risks". Below these tabs is a table titled "Event Summary" with columns for Event Summary, Target Completion Date, Owner, and Status. The table contains eight rows of event data. At the bottom left is an "Export" button.

图 5-3 使用 JavaScript 页面组件的页面整体效果

Client Logo

Logout

This screenshot shows a dynamic web page for the same project. The "Executive Summary" tab is active, displaying the same week beginning on Oct 28, 2007, with overall status "Good" and progress "65%". A modal dialog box is open over the table, showing a summary of the selected row. The table below has columns for Client, Project, Sub-Project, Summary ID, Week Beginning, Progress, Overall Status, and Status. The table contains 20 rows of project data. At the bottom right is a "View All..." link.

图 5-4 使用 JavaScript 页面组件的页面动态布局和表列拖动效果

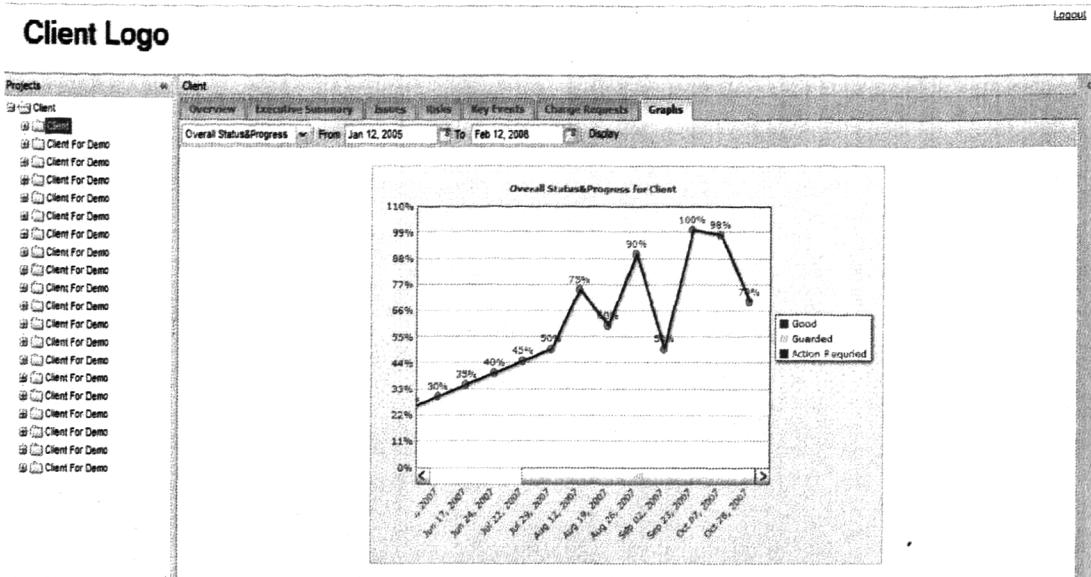


图 5-5 使用 FLASH 组件实现的页面动态图表效果

5.2 表现层关键技术实现

如前文所述 JAVA EE 平台中，表现层的关键在于用户请求的捕获、包装与分发，调用相关的业务服务并回送相应数据，以及用户界面的生成。将这些实际问题映射到具体开发过程中来，相对应的开发问题分别是 Request Dispatcher、Model、Action、View。为解决这些问题，我们表现层使用了 struts 框架来简化开发过程。

5.2.1 struts 开发框架简介

在基于 JAVA EE 平台的 Web 应用开发中，MVC 模式是被推荐并广泛利用的设计模式。其核心思想是把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离，把一个应用被分成三个逻辑上相互关联的三个模块——模型、视图、控制。当有事件导致 Controller 改变 Model 或 View，或者同时改变两者时，只要 Controller 改变了 Models 的数据或者属性，所有依赖的 View 都会自动更新。类似的，只要 Controller 改变了 View，View 会从潜在的 Model 中获取数据来刷新自己^[15]。

在传统的 JAVA EE 平台上，Model、View、Controller 分别是由 JavaBean、JSP、Servlet 实现的。但这种实现方式随着系统规模的增大、功能处理的负责，

会越来越显得笨拙和难以维护^[16]。

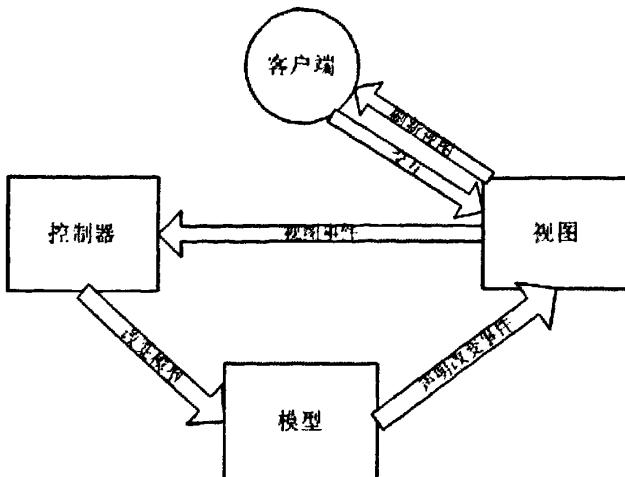


图 5-6 JAVA EE 经典 MVC 模式示意图

struts 就是为了简化和优化 JAVA EE 平台上的 MVC 模式处理过程的一套 MVC 框架。这个框架使用一个默认的 servlet 做为前端控制器，同时也是系统的核心，实现用户请求的获取、包装和分发。分别使用 FormBean、JSP、Action 作为 Model、View、Controller 的实现^[17]。

5.2 struts 开发应用与配置说明

struts 内置的核心类 org.apache.struts.action.ActionServlet 已经实现了请求获取、包装和分发的能力，可以直接将它作为 RequestDispatcher 使用。为了在应用中利用 struts 的功能，需要通知应用服务器在初始化应用时装入 ActionServlet 并对它进行初始化，需要在 Web 应用的核心配置文件 web.xml 中加入如下内容：

```

<servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
        <param-name>config</param-name>
        <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
    <init-param>
        <param-name>debug</param-name>
        <param-value>1</param-value>
    </init-param>

```

```

<init-param>
    <param-name>detail</param-name>
    <param-value>1</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.dhtml</url-pattern>
</servlet-mapping>

```

通过上述配置，所有对系统中以 dhtml 为扩展名的资源访问的请求都会有 ActionServlet 捕获到，并按照在 struts-config.xml 文件中配置约束的进行包装和分发。

在处理过程中 ActionServlet 会将请求包装成用户定义的 FormBean 并交给与之匹配的用户自定义的 Action 进行处理。用户定义的 FormBean 是一个继承自 org.apache.struts.action.ActionForm 的 Java 类。类中包含与请求属性名称一一对应的属性，同时还可以包含一些验证方法。用户自定义的 Action 是继承自抽象类 org.apache.struts.action.Action 的 Java 类。为了满足处理需要，用户在自定义 Action 类的时候需要实现超类中的 execute 方法^[18]。下面分别是用于登录的 FormBean 和用于登出的 Action 代码：

```

//Login Action Form 源代码
package com.company.pm.admin.logon;
public class LogonActionForm extends ActionForm {
    String userName="";
    String password="";
    public String getPassword() {
        return password.trim();
    }
    public void setPassword(String password) {
        this.password = password.trim();
    }
    public String getUserName() {
        return userName.toLowerCase().trim();
    }
    public void setUserName(String userName) {
        this.userName = userName.trim();
    }
}

```

```
//Logout Action 源代码
package com.company.pm.admin.logon;

public class Logout extends Action{
    public ActionForward execute(ActionMapping mapping, ActionForm actionForm,
HttpServletResponse request, HttpServletRequest response) throws Exception {
        HttpSession session = request.getSession(false);

        //remove the attributes of the session
        if(session != null) {
            session.removeAttribute("userName");
            session.removeAttribute("userId");
            session.removeAttribute("roleId");
            session.removeAttribute("requiredURL");
            // make the session invalid
            session.invalidate();
        }
        response.sendRedirect(mapping.getInput());
        return null;
    }
}
```

struts 的自动处理机制是以 XML 配置文件为核心的，为实现用户请求、FormBean 和 User Action 的匹配处理，需要将相关信息配置到 struts-config.xml 文件中，示例配置如下：

```
<struts-config>
<form-beans>
    <form-bean name="logonActionForm"
        type="com.company.pm.admin.logon.LogonActionForm" />
</form-beans>
<action-mappings>
    <!--for login -->
    <action path="/logonAction"
        type="com.company.pm.admin.logon.LogonAction"
        name="logonActionForm" scope="request" validate="true"
        input="/logon.jsp">
        <forward name="success" path="/MyHomeEntry.do" />
    </action>
</action-mappings>
</struts-config>
```

5.3 AJAX 关键技术实现

为实现良好的客户端界面和丰富的用户使用体验，在前端展示平台中引入和 AJAX 技术。在开发过程中 AJAX 技术的开发关键主要有两点，一是异步地 HTTP 请求，二是动态的用户界面布局。

异步地 HTTP 请求需要借助于浏览器内置的 XMLHttpRequest 对象向服务器发出请求。在本系统中，对 XMLHttpRequest 的调用方式使用 JavaScript 做了简单封装，封装方式如下：

```

var xmlhttp;

function createXMLHttpRequest() {
    if(window.ActiveXObject){
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
    } else if(window.XMLHttpRequest) {
        xmlhttp = new XMLHttpRequest();
    }
}

/**
 * send a ajax request to server and parse response text using call back method
 * .@param url String : the target url
 * .@param parameterString String : if you are using POST method , it will be used as parameters
 * string
 * .@param method String : GET or POST
 * .@param asynch Boolean : true or false whether the request is asynch
 * .@param callBack Function : the call back method to resolve response string. function
callBack( String )
*/
function ajaxRequest(url , parameterString , method , asynch , callBack){
    createXMLHttpRequest();
    callBackMethod = callBack;
    if(method=="GET"){
        xmlhttp.onreadystatechange = handleStateChange( callBack );
        xmlhttp.open("GET" , url , asynch );
        xmlhttp.send(null);
    }else if ( method == "POST"){
        xmlhttp.onreadystatechange = function(){
            if(xmlHttp.readyState==4){
                if(xmlHttp.status==200){
                    if(callBack)
                }
            }
        }
    }
}

```

```

        callBack(xmlHttp.responseText);
    }
}
};

xmlHttp.open("POST" , url ,asynch );
xmlHttp.setRequestHeader("Content-Type" , "application/x-www-form-urlencoded;
charset=UTF-8");
xmlHttp.send(parameterString);
}

}

```

在实际开发过程中，只需要调用 ajaxRequest 函数，并传入相关参数，就能实现对服务器的异步请求。

动态的页面布局方式是通过 Ext-JS，Javascript 页面组件库来实现的。Ext-JS 是一个开源的 JavaScript 页面组件库，其原理是通过在传统的 HTML 页面的基础上加入浏览器可以自动调用的 JavaScript 代码来对 HTML 中的页面元素来进行装饰和渲染，最终显示给用户具有良好界面和丰富功能浏览器页面^[10]。如前端展示系统中左侧项目树的实现：

在 HTML 代码中定义用于显示项目树的容器 div：

```
<div id="leftDiv" class="x-layout-inactive-content" style="overflow:auto;"> </div>
```

然后通过 Ext-JS 提高的 Tree 组件进行装饰和填充：

```

//initial left tree menu
var tree = new Ext.tree.TreePanel('leftDiv', {
    animate:true,
    loader: new Ext.tree.TreeLoader({dataUrl:'showClientTree.dhtml'}),
    enableDD:false,
    containerScroll: true,
    ddScroll: true,
    rootVisible:true,
    dropConfig: {appendOnly:true}
});
// set the root node
var root = new Ext.tree.AsyncTreeNode({
    text: 'E5 Systems Client',
    draggable:false, // disable root node dragging
    id:'rootNode'
});
tree.setRootNode(root);
// render the tree

```

```
tree.render();
root.expand(false, /*no anim*/ false);
```

将上述代码加入到页面 Body 元素的 onload 事件中，当页面读取完毕的时候，名字叫 leftdiv 的 div 组件就会被当作左侧的项目树被装饰出来。

5.4 业务逻辑层关键技术实现

在本系统中，为简化开发过程、提高开发效率，使用轻量级容器框架 Spring 作为逻辑层的基础框架。Spring 是一个完整的应用框架，它可以在很多应用层面发挥作用。它的目标是提供一种贯穿始终的解决方案，讲课中专用框架整合成一个连贯的整体架构，消除 JAVA EE 应用中众多自制的工厂和 Singleton。它的关键性技术是提供了一个控制反转（IoC, Inversion of Control）容器。通过这个容器，Spring 允许使用 bean 工厂的方式统一去装配所有应用对象，而所有的应用对象都是 POJO (Plain Old Java Object) 对象，被 bean 工厂管理的对象根本不需要知道 Spring 的存在。所以 Spring 也是一个真正的无侵入性的应用框架^[20]。

为支持 Web 应用，Spring 提供了“Web 应用上下文”(WebApplicationContext) 的概念，将强大的轻量级 IoC 容器无缝结合到 web 环境中，各种 web 层程序都可以使用这个容器。通过这个 web 应用上下文提供的 bean 工厂方法，可以非常容易的找到系统提供的独立的 POJO 服务对象。这个 WebApplicationContext 可以做的应用中的 ServiceLocator 来定位服务对象^[21]。

为在 web 应用中使用 Spring 的 web 应用上下文，需要在 web.xml 中定义一个监听器（ContextLoaderListener），用于加载根应用上下文。这个监听器在 web.xml 中的定义方式如下：

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/conf/applicationContext.xml</param-value>
</context-param>

<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

下面选取通过 Spring 对本系统数据源和数据使用对象的组装配置来说明 Spring 使用依赖注入方式对各对象及其关系的处理：

```
<beans>

    <!-- ===== GENERAL DEFINITIONS
          -->

    <!-- (in this case, mail and JDBC related properties) -->
    <bean id="dataSource"
          class="org.springframework.jndi.JndiObjectFactoryBean">
        <property name="jndiName">
            <value>java:jdbc/humanResourceDS</value>
        </property>
    </bean>
    <bean id="transactionManager"
          class="org.springframework.transaction.jta.JtaTransactionManager" />
    <bean id="baseTransactionProxy"
          class="org.springframework.transaction.interceptor.TransactionProxyFactoryBean"
          abstract="true">
        <property name="transactionManager">
            <ref bean="transactionManager" />
        </property>
        <property name="transactionAttributes">
            <props>
                <prop key="insert*>">PROPAGATION_REQUIRED</prop>
                <prop key="update*>">PROPAGATION_REQUIRED</prop>
                <prop key="*>">PROPAGATION_REQUIRED,readonly</prop>
            </props>
        </property>
    </bean>
    <bean id="sqlMapClient"
          class="org.springframework.orm.ibatis.SqlMapClientFactoryBean">
        <property name="configLocation">
            <value>/WEB-INF/conf/sql-map-config.xml</value>
        </property>
        <property name="dataSource">
            <ref bean="dataSource" />
        </property>
    </bean>
    <!-- General Dao -->

    <bean id="commonDao"
          class="com.company.pm.common.dao.CommonDAO">
```

```

<property name="sqlMapClient">
    <ref local="sqlMapClient" />
</property>
</bean>
</beans>

```

5.5 持久层关键技术实现

本系统使用 iBATIS 作为持久层的实现框架。

iBATIS 是一个简单实用的持久层框架，它的创意非常简单：SQL 语句在 XML 文件中定义并预留参数占位符（可以指定参数的 SQL 类型和其他细节）。在执行时，占位符将被指定的参数值所取代，参数可能来自参数映射表（parameter map）、JavaBean 属性（bean property）或是简单的参数对象（simple parameter object）。在执行 SQL 查询时，结果字段将被映射到对象，映射的方式与参数映射是一样的^[22]。

iBATIS 实际上是一个半对象关系映射（ORM，Object Relation Mapping）的框架，与 Hibernate 等完整的 ORM 框架并不完全相同。本系统选取它作为持久层框架而不是 Hibernate 等框架的原因是本系统有大量的数据统计工作，对 SQL 的灵活程度和自定义要求比较高，而 iBATIS 对 SQL 语句有开放性的支持。而选取 iBATIS 作为持久层底层支持的缺陷就是在一定程度上放弃了数据库底层操作的可移植性。

以下用登录模块的数据库操作模块来对 iBATIS 的使用方法做实例说明。

登录模块的 SQL 配置文件：

```

<sqlMap namespace="UserLogon">
    <resultMap id="userLogon" class="com.company.pm.admin.logon.UserLogon">
        <result property="userId" column="ID"/>
        <result property="userName" column="USER_NAME"/>
        <result property="password" column="PASSWORD"/>
        <result property="fullName" column="fullName"/>
        <result property="roleId" column="ROLE_ID"/>
        <result property="status" column="STATUS"/>
        <result property="timeZone" column="TIME_ZONE"/>
        <result property="locationId" column="LOCATION_ID"/>
    </resultMap>

```

```

<statement id="checkUser" resultMap="userLogon" >
    SELECT ID, USER_NAME, PASSWORD, CONCAT(FIRST_NAME,
    ",LAST_NAME) AS fullName, ROLE_ID, STATUS, TIME_ZONE, LOCATION_ID FROM
    USER_BASIC WHERE USER_NAME=#userName# AND
    PASSWORD=PASSWORD(#password#)
</statement>
</sqlMap>

```

对 checkUser 这条 SQL 语句的使用方式：

```

package com.company.pm.admin.logon;

import org.springframework.orm.ibatis.support.SqlMapClientDaoSupport;

public class UserLogonDaoImpl extends SqlMapClientDaoSupport implements IUserLogonDao {

    public Object logonCheck(UserLogon user){

        return getSqlMapClientTemplate().queryForObject("checkUser", user);
    }

}

```

5.6 远程调用框架的实现

本系统为了让新系统享用老系统已有的业务服务方法，而省去相同功能的重复构建冗余工作，在系统中引入了远程调用框架使新老系统的服务调用能有机地融合在一起。在传统的 C/S 系统远程方法调用方法中，系统对底层远程方法调用的框架依赖性较强，而且移植能力、跨平台能力都很差；而新的基于 Web Service 的远程数据访问方式又依赖于很多平台以外的外部定义语言，如 XML、WSDL (Web Service Definition Language, web 服务调用语言)。

本系统选取了轻量级的远程服务调用框架 Hessian 作为远程服务调用的实现框架。Hessian 和传统的远程方法调用一样，通过二进制来建立客户端与服务器端的消息交流，但 Hessian 的实现方式更简单，提供了跨多种平台的远程方法调用的实现，而且在 JAVA 平台上，还可以 POJO 对象实现远程方法调用。所以 Hessian 是 JAVA 平台最佳的远程方法调用选择方案^[23]。

下面选取前端展示平台利用原有系统的服务读取系统中已有项目列表的方式对 Hessian 的使用方法进行说明。

首先定义新系统和原有系统公用的服务接口：

```
package com.company.rmi;

import java.util.List;

public interface ProjectRmiService {
    public List getProjectList();
}
```

在原有系统添加新的基于 HessianServlet 的用户类，并实现这个服务接口用于对外提供服务：

```
package com.company.rmi.serviceImpl;

public class ProjectRmiServiceImpl extends HessianServlet implements ProjectRmiService {

    public List getProjectList() {

        WebApplicationContext wac =
        WebApplicationContextUtils.getWebApplicationContext(this.getServletContext());
        ProjectService ps = (ProjectService) wac.getBean("ProjectServiceBean"); //get service
        object from spring container
        return ps.getProjectList();
    }

}
```

通过在 web.xml 中对这个实现服务接口的 Servlet 进行定义，将服务开放给外部程序：

```
<servlet servlet-name="projectRMI"
servlet-class="com.company.rmi.serviceImpl.ProjectRmiServiceImpl"/>
<servlet-mapping url-pattern="/project.rmi" servlet-name="projectRMI"/>
```

最后在新系统中创建对这个远程服务的调用：

```
package com.company.rmi;

import java.net.MalformedURLException;
import java.util.List;

import com.caucho.hessian.client.HessianProxyFactory;
```

```

public class ProjectRmiInvoke {
    public List getProjectListFromRmi() throws MalformedURLException{
        List result = null;

        String rmiUrl = "http://e5systems.url/project.rmi";
        HessianProxyFactory factory = new HessianProxyFactory();

        ProjectRmiService service = (ProjectRmiService)
factory.create(ProjectRmiService.class, rmiUrl);
        result = service.getProjectList();
        return result;
    }
}

```

5.7 系统部署

本系统最终的部署使用的应用服务器是 JBOSS Application Server 4.0.2。JBOSS Application Server 是一个开源的 JAVA EE 应用服务器，其核心是在 Tomcat (一个被广泛使用的 Java 轻量级 Web 服务器) 的基础上加入了对 EJB、JMX 等高级 JAVA EE 标准的实现。JBOSS Application Server 可以为应用提供更高的标准化和安全性方面的服务。

JBOSS 采用的部署方式是包部署方式，整个应用需要被打成一个.war 的压缩包或者放到一个以.war 为结尾的应用目录下才可被服务器识别并部署。JBOSS 所使用的数据源 JNDI 定义方式与其他服务器，也有所区别^[24]。它是通过在部署目录下定义相关数据源的 xml 文件来实现的，如本应用所用的 MySQL 的数据源 JNDI 文件 mysql-ds.xml 就应该被放在相关的/deploy 目录下，文件定义如下：

```

<?xml version="1.0" encoding="UTF-8"?>

<!--
-->
<!--
-->
<!-- JBoss Server Configuration
-->
-->
-->
-->

```

```
-->

<datasources>
    <local-tx-datasource>
        <jndi-name>jdbc/managementDS</jndi-name>

        <connection-url>jdbc:mysql://localhost:3306/HRMANAGEMENT?useUnicode=true&characterEncoding=GBK</connection-url>
            <driver-class>com.mysql.jdbc.Driver</driver-class>
            <user-name>root</user-name>
            <password>Garriot</password>
        </local-tx-datasource>
        <local-tx-datasource>
            <jndi-name>jdbc/presentationDS</jndi-name>

            <connection-url>jdbc:mysql://localhost:3306/PRESENTATION?useUnicode=true&characterEncoding=GBK</connection-url>
                <driver-class>com.mysql.jdbc.Driver</driver-class>
                <user-name>root</user-name>
                <password>Garriot</password>
            </local-tx-datasource>
        </datasources>
```

本系统使用的数据库是 MySQL 5.0，所使用的服务器操作系统是 RedHat Linux 7.3。为使应用正常运行，应用服务器及数据库服务器需要每一个月进行一次维护和备份。

第6章 总 结

本系统的开发方法使用了敏捷开发方法，充分强调客户在开发过程中的参与，采用小步迭代、持续集成、增量设计等新的开发实践理论为指导，成功地开发了一个可用于指导敏捷软件开发项目管理与实践的软件产品。

本系统建立了一套比较完整的项目管理系统，包括项目需求管理、需求变更管理、里程碑管理等等，照顾到了软件项目管理的各个方面，建立了一套自动化的管理体系。通过本系统可以有效规范软件开发过程管理中的各个环节，降低软件开发过程中的风险。

同时软件工程管理与开发实践方法论仍是一个在不断丰富发展的学科，所以本系统也是一个期待在实际实践过程中进行不断修正与丰富的系统。能使本系统跟得上方法论的发展，更符合新思想的实践方式是本系统下一步开发的目标。

参考文献

- [1] Kathy Schwalbe 王金玉 时郴 译《IT 项目管理》：机械工业出版社
- [2] Kent Beck、Cynthia Andres 雷剑文、陈振冲、李明树译 《解析极限编程》：电子工业出版社 2006.5
- [3] 赖宇阳 《中文 Microsoft Project 2000 教程》：北京希望电子出版社 2001.1
- [4] Carig Larman 姚淑珍 译 《UML 和模式应用》：机械工业出版社 2002.1
- [5] Dean Leffingwell、Don Widrig 著 《软件需求管理——统一化方法》：高等教育出版社 2002
- [6] Sue Spielman、Meeraj Kunnumpurath 著 《J2EE 1.4 编程指南》：电子工业出版社 2005.4
- [7] 金灵 等译 《AJAX 基础教程》：人民邮电出版社 2006.2
- [8] Deepak Alur、John Crupi、Dan Malks 著 刘天北、熊节译 《J2EE 核心模式》 机械工业出版社 2005.11
- [9] Chris Richardson 著 李林骁、王海龙译 《POJOs IN ACTION》 电子工业出版社 2007.4
- [10] Jim Arlow、Ila Neustadt 方贵宾、胡辉良译《UML 2.0 和统一过程（原书第 2 版）》：机械工业出版社 2006.6
- [11] Michael Mahemoff 著 杨仁和译 《AJAX 设计模式》：电子工业出版社 2007.5
- [12] Dave Crane、Eric Pascarello、Darren James《AJAX In Action》：Manning Publications 2006
- [13] Lai R 著 周斌 等译《J2EE 平台 Web Services》：电子工业出版社 2005.6
- [14] 林上杰，林康司 编著 《JSP 2.0 技术手册》：电子工业出版社 2004.5
- [15] Erich Gamma、Richard Helm Ralph、Johnson John Vlissides 李英军、马晓星、蔡敏、刘建中译《设计模式：可复用面向对象软件的基础》：机械工业出版社 2005.6

- [16] 计磊、李里、周伟 《精通 J2EE--Eclipse、Struts、Hibernate 及 Spring 整合应用案例》：人民邮电出版社 2006.8
- [17] Husted Ted、Franciscus George、Dumoulin Cedric、Winterfeldt David 著 《Struts in Action》: O'reilly & Associates Inc 2005.7
- [18] JAMES HOLMES、孙燕等译《Struts 程序员查询辞典》：中国铁道出版社 2005.6
- [19] Ext-JS Documents <http://extjs.com/deploy/dev/docs/>
- [20] Rod Johnson、Juergen Hoeller : 《Expert One on One J2EE Development Without EJB》 : Wiley Computer Publishing, 2005 143-175
- [21] Craig Walls、Ryan Breidenbach 《Spring in Action》: Manning Publications 2005
- [22] Clinton Begin、Brandon Goodin、Larry Meadors 《Ibatis in Action》: Manning Publications 2007.1
- [23] 李刚著 《Spring 2.0 宝典》: 电子工业出版社 2006.10
- [24] Marc Fleury, Scott Stark 等著 罗时飞译《Jboss 管理与开发核心技术》: 电子工业出版社 2004.1
- [25] Abraham Silberschatz, Henry F.Korth , S. Sudarshan 杨东青等 译 《数据库系统概念》: 2000.2
- [26] Thimas M. Connolly, Carolyn E. Egg 何主洁 黄婷儿译《数据库设计教程 (第 2 版)》 : 机械工业出版社 2005.1
- [27] Robert C. Martin 著 邓辉译 《敏捷软件开发——原则、模式与实践》: 清华大学出版社 2003 年 9 月
- [28] James Rumbaugh, Ivar Jacobson, Grady Booch 著 《UML 参考手册》, 机械工业出版社, 2004 年 5 月版
- [29] Bruce Eckel 著 陈吴鹏、饶若楠等译, Thinking in Java, 机械工业出版社
- [30] Three-Layered Services Application (三层服务应用程序) MSDN 2004.4

致 谢

首先要感谢我的导师史清华老师，在整个论文的撰写过程中，他那一丝不苟的作风，严谨的治学态度深深教育了我，对史老师的敬佩之情是无法用语言表达的。

其次，我要感谢山东亿帆环球科技公司给了我这个实习的机会，让我在学校中学到的理论能够结合实践，让我的学识得到升华。

我还要感谢项目组的同事和领导，在项目实施过程中，我们互相帮助，互相学习，精诚合作，最终成功交付了项目。

最后，我要感谢我的家人，父母，他们给了我无私的关怀，让我能够安心于学业当中。

作者： 张晓良
学位授予单位： 山东大学

相似文献(10条)

1. 学位论文 宁晓倩 基于系统动力学的软件开发项目管理 2004

“软件危机”一直困扰着软件产业的发展。软件危机产生的原因与业界缺乏对软件开发过程的深入理解有关。每个软件开发项目都是一个动态的复杂的复杂系统。没有对软件开发生命周期和影响软件开发生命周期绩效的因素的深入理解，就不能制订出提高项目绩效的有效策略。该文的目的就是通过建立一个模拟软件开发生命周期模型来帮助管理者增加对软件开发生命周期和软件开发生命周期中一些独特现象的理解。该文的结构安排如下：导论部分讨论了该文的立题背景和目的。第一章回顾了软件开发生命周期的经典理论，和系统动力学理论与方法在项目管理中的应用现状。第二章到第四章是该文的重点。第二章详细描述了该文模型的结构和模型参数的估计依据；第三章通过案例分析验证了模型的有效性；第四章把模型作为实验工具对一系列软件开发生命周期策略对项目绩效的影响进行了分析。第五章对全文进行了总结，并讨论了系统动力学模型作为学习工具在提高组织软件开发生命周期水平方面的优势，最后对模型本身和应用范围的拓展进行了讨论。该文的主要创新点有：1. 建立了一个包括软件开发生命周期、人员管理、计划、控制等方面的综合性的软件开发生命周期模型，对软件开发生命周期中与人有关的软因素进行了充分的考虑，从战略性的视解帮助软件项目管理者分析理解软件开发生命周期和软件开发生命周期中存在的独特现象，从而制定有效的管理策略。2. 数组的运用使模型使用者可以根据需要把开发生命周期细分为不同的开发阶段，并运用模型方便灵活地对多阶段的开发生命周期进行模拟。而现有的关于项目管理的系统动力学模型一般都在模型结构中就规定了模型所能模拟的开发阶段。3. 该文的系统动力学模型可以作为软件开发生命周期后评审工具，对项目行为产生的原因进行分析。4. 该文的系统动力学模型可以对软件开发生命周期策略所产生的后果进行定量的分析和预测。

2. 期刊论文 孟宇桥, MENG Yu-qiao 项目管理在高校软件开发中的应用 -吉林工程技术师范学院学报 2008, 24(5)

随着信息产业的飞速发展，项目管理在软件开发中的应用日益受到重视。而高校软件开发中的项目管理应用却相对落后于企业，本文主要分析了高校软件开发中在项目管理方面存在的问题及解决方案。

3. 学位论文 田力红 现代商业银行软件开发项目管理问题研究 2004

随着中国金融业改革的不断深入，银行业完成了从计划经济体制下的专业银行到市场经济体制下的商业银行的转变，银行业之间的竞争日趋激烈。现代商业银行的竞争主要是通过改进服务手段、增加服务功能、拓展业务品种、提高服务效率来实现的。而实现这一目标的有效途径就是通过实现银行电子化，利用高科技的IT手段，为客户提供方便快捷、丰富多样的金融服务，以增强自身的竞争能力和发展能力，取得更大的经济效益。因此现代商业银行开发高效适用的应用软件系统的能力就直接决定了银行的竞争能力。为满足银行不断增长的金融业务需求和产品创新需求来进行软件开发是一项高度复杂的工作。如何将银行应用软件开发工作以系统高效的方式进行，是该文的研究主题。项目管理是当今软件工程界用于保证软件质量和提高生产效率的主要手段之一。该文运用项目管理理论和软件工程理论，结合现代商业银行的实际情况，对应用软件开发项目的集成管理、范围管理、时间管理、成本管理、质量管理、人力资源管理、沟通管理、风险管理、采购管理九方面的内容进行了研究探讨，给出了商业银行实施软件开发项目管理可操作的方法。将项目管理方法应用于某商业银行的开发实际，取得了很好的效果。尤其是在人力资源、时间、质量等方面效果显著。通过该行的应用实践表明将项目管理方法应用于银行的软件开发切实可行，但在应用水平方面有待提高。

4. 学位论文 史本娜 我国中小软件开发企业的项目管理研究 2008

软件开发行业有不同于传统行业的一些固有特点，而我国的中小软件开发企业的项目管理状况又因为外部宏观环境和自身的一些原因更加不容乐观。本文首先介绍项目管理的基本定义和理论框架，软件开发项目管理的一些基础概念和理论以及其相对于传统行业项目管理的特点。然后结合我国中小软件开发企业的实际情况从成本、人力资源和风险这些在中小软件开发企业的项目管理工作中表现突出影响较大的三个方面分别探讨软件开发项目管理的理论、问题及方法。

在软件开发项目的成本管理、人力资源管理和风险管理中，本文分别介绍了项目管理理论中关于相关领域的知识，对应的软件开发项目管理相应的方法，并针对我国中小软件开发企业存在的问题，提出了相应的一些改进策略。特别探讨了在软件开发项目估算预算方面比较成熟的一个参数模型COCOMOII模型及其22个成本驱动因子。进而，基于这些成本驱动因子简单分析了人力资源管理和风险管理对成本的影响。

最后，本文根据我国中小软件开发企业的实际情况提出一套项目管理改进方案，并且结合COCOMOII模型模拟了改进前后的成本状况。通过本文的分析和模拟可以得出结论，通过对组织结构和业务流程及相应管理规范的重新思考与定义，我国的中小软件开发企业也可以发挥出自身优势降低成本改善项目管理状况。

5. 会议论文 王妍凤, 郑轶 软件开发项目管理方法研究 2007

列举并分析了目前软件开发中存在的问题，提出以项目管理的方式对软件开发过程进行管理。在对项目管理进行简单介绍的基础上，将软件开发作为一个项目，以项目管理技术对软件开发过程中的人力资源等进行全面的管理。提出了应用先进的软件项目管理理论和方法，有效地提高软件开发效率，确保软件质量。

6. 学位论文 胥静 出口软件开发项目管理技术研究及应用 2005

当前，世界经济的发展正在使国际间的产业关联度日益增加，全球软件市场的需求也持续上升，这种趋势为发展中国家的软件业提供了发展机会。在此背景下，软件企业发展外包业务已经成为世界软件产业发展的一种趋势。发达国家的软件外包已经成为软件企业发展的必要手段，国际间的软件外包业务已经日益成熟。因此，中国软件产业要发展，软件出口将是一个非常重要的途径。

作者在一家专门做对日软件外包的企业工作，参与了许多项目的整个开发过程。其间不但学习了关于开发软件的各种技术，对于整个项目的项目管理更是深有体会。所在公司是一家对于对日软件外包已有一定经验的软件企业。本文的内容主要是通过对软件开发项目管理理论包含的三个方面：项目管理技术、软件开发过程管理、人文管理的介绍，结合实际的开发项目引入公司的一套已经比较成熟的针对软件外包的管理框架。这套框架是公司一些比较资深的项目经理通过将近十年的工作经验，以及对于项目管理理论的学习和创新总结出来的，对于项目管理以及软件质量的提高都起到了很有效的作用。

近几年，我国引进了许多项目管理理论，而对于这些理论如何结合各类项目的实际，如何有效地应用于实践，却关注较少。作者在校期间学习了国内外特别是国外经典的软件工程和项目管理理论，认识到了项目管理对于软件开发的重要性。在公司工作期间，将理论知识融入这套框架，得到应用并起到了一定的成效。

本文希望通过介绍实际的项目开发过程及其项目管理经验，总结这套框架，以期望能介绍给更多的项目管理人员，与同行业的人员相互交流和互相学习。通过吸收和总结现有的模式和规范，来提高中国软件开发的项目管理能力，从而提高中国出口软件开发的国际竞争力，打造自己的企业核心竞争力。

7. 学位论文 王辉 项目管理在软件开发中的应用和改进 2005

本文以江苏联通炫铃网关项目为背景，结合软件项目管理的相关理论，从项目范围确定、项目管理组织形式、项目风险管理、项目控制过程几个方面深入细致的阐述了软件项目管理在软件项目中的应用方法。提出了软件项目管理过程中，要充分重视项目定义。对项目开发过程中主要经验进行总结并提炼出项目管理的策略，要明确定义用户需求，明晰项目价值和项目目标；提高计划执行准确率，加强基础数据积累和分析；做好项目人力资源管理，实现动态、灵活调度；加强项目过程控制；提出成功的软件开发项目标准，提出软件项目的目标实现是多重的，各种因素和资源受到多种牵制。提出着眼于提高软件开发企业项目管理整体能力的知识管理，并对ISO质量管理体系和CMM软件成熟度进行了评估和对比。

8. 期刊论文 甘露, 周娟 浅谈软件开发中的项目管理 -金卡工程2009, 13(8)

随着信息产业的飞速发展,项目管理时于以应用开发为主的软件企业是一个行之有效的管理方法,项目管理在软件开发中的应用日益受到重视.本文主要通过对项目管理在软件开发中的应用的成因、存在的问题以及相应的解决方案进行了分析和论述.

9. 学位论文 李知菲 基于WEB的信息系统软件开发与项目管理 2005

本文结合一个WEB工程项目实例即浙江交通技师学院驾驶员培训管理信息系统(文中简写为JPMIS)的软件开发与项目管理过程及相关成果,提出了一些基于WEB工程软件开发与项目管理技术的,应用于我国中小型软件企业WEB工程的具体实施方法。

首先简述了在JPMIS开发中使用的软件工程理论、项目管理方法和具体实现JPMIS开发的相应软件工具。同时也结合WEB工程开发实际,给出了选择上述理论、方法和开发工具的理由。

其次,阐述了JPMIS的WEB工程项目管理过程。例如,根据JPMIS实际需求对CMM2级KPA的裁剪和利用Project制定系统的项目计划等。

最后,论述了JPMIS的具体开发过程。其中包括利用VISIO建立系统的UML模型,给出了“报名管理”、“理论考试”和“超级管理”等业务模块的源代码。另外,还论述了WEB工程测试方法和系统SQA执行过程。

10. 学位论文 潘崴 GDKT公司软件开发项目管理问题研究 2005

本文针对中国中小型软件企业的软件开发项目普遍存在进度拖延、成本超支和质量难以令人满意等问题,对软件开发项目管理问题进行了研究。文章通过对一中小型软件开发企业——GDKT公司在项目管理中存在的问题的深入细致的剖析,从具体流程和制度入手,为其建立起一套简单实用的项目时间管理、成本管理和质量管理流程,并将三者有机结合在一起。通过对这一典型案例的研究,可为中国广大的中小型软件企业提高自身项目管理水平和能力提供借鉴和参考。

本文链接: http://d.g.wanfangdata.com.cn/Thesis_Y1352093.aspx

下载时间: 2010年5月18日