

论混合软件架构的设计

摘要:

2007 年 3 月,我所在的公司组织开发了一套完整的变电综合信息管理系统,在这个项目中,我担任系统架构设计师职务,主要负责软件架构和网络安全体系架构设计的工作。该系统包括变电运行所需的运行记录、图形开票、安全生产管理、生产技术管理、行政管理、总体信息管理、技术台帐管理、班组建设、学习培训、系统维护等各个业务层次模块。

本文首先简单地分析了 C/S 架构和 B/S 架构各自的优缺点,然后说明了混合 C/S 架构和 B/S 架构的必要性,分析了“内外有别”和“查改有别”两种混合模型,并以变电综合信息管理系统为例,结合实际情况,讨论了 C/S 和 B/S 混合架构的应用。实践证明,在软件项目的开发中,使用 C/S 与 B/S 混合软件架构,能节省开发和维护成本,使系统具有良好的开放性,易扩展性,便于移植等优点。

正文:

典型的软件架构风格有很多。例如,设计图形用户界面常用的事件驱动风格、设计操作系统常用的层次化设计风格、设计编译程序常用的管道与过滤器风格、设计分布式应用程序常用的客户机/服务器风格等。一个实用的软件系统通常是几种典型架构风格的组合。

1. 项目概述

当前,我国电力系统正在进行精简机构的改革,变电站也在朝无人、少人和一点带面的方向发展(如:一个有人值班 220KV 变电站带若干个无人值班 220KV 和 110KV 变电站),“减人增效”是必然的趋势,而要很好地达到这个目的,使用一套完善的变电综合信息管理系统(TSMIS)显得很有必要。2007 年 3 月,笔者所在的公司组织有关力量,针对电力系统变电运行管理工作的需要,结合变电站运行工作经验,开发了一套完整的变电综合信息管理系统,在这个项目中,我担任系统架构设计师职务,主要负责软件架构和网络安全体系架构设计的工作。

TSMIS 系统包括变电运行所需的运行记录、图形开票、安全生产管理、生产技术管理、行政管理、总体信息管理、技术台帐管理、班组建设、学习培训、系统维护等各个业务层次模块。实际使用时,用户可以根据实际情况的需要选择模块进行自由组合,以达到充分利用变电站资源和充分发挥系统作用的目的。限于篇幅,在此,我们不详细介绍各模块的功能。

系统的实现采用 Visual C++、Visual Basic、Visual InterDev 和 Java 语言和开发平台进行混合编程。服务器操作系统使用 Windows 2003 Advanced Server,后台数据库采用 SQL Server 2005。系统的实现充分考虑到我国变电站所电压等级的分布,可以适用于大、中、小电压等级的变电站所。

2. C/S 与 B/S 的比较分析

C/S 架构具有强大的数据操作和事务处理能力,模型思想简单,易于人们理解和接受。但随着企业规模的日益扩大,软件的复杂程度不断提高,C/S 架构逐渐暴露了以下缺点:

(1) 开发成本较高。C/S 架构对客户端软硬件配置要求较高,增加了整个系统的成本。

(2) 客户端程序设计复杂。采用 C/S 架构进行软件开发,大部分工作量放在客户端的程序设计上,客户端显得十分庞大。

(3) 软件移植困难。采用不同开发工具或平台开发的软件,一般互不兼容,不能或很难移植到其它平台上运行。

(4) 软件维护和升级困难。采用 C/S 架构的软件要升级,开发人员必须到现场为客户机升级,每个客户机上的软件都需维护。对软件的一个小小改动(例如,只改动一个变量),每一个客户端都必须更新。

B/S 架构主要是利用不断成熟的 WWW 浏览器技术,结合浏览器的多种脚本语言,用通用浏览器就实现了原来需要复杂的专用软件才能实现的强大功能,并节约了开发成本,是一种全新的软件架构。基于 B/S 架构的软件,系统安装、修改和维护全在服务器端解决。用户在使用系统时,仅仅需要一个浏览器就可运行全部的模块,真正达到了“零客户端”的功能,很容易在运行时自动升级。B/S 架构还提供了异种机、异种网、异种应用服务的联机、联网、统一服务的最现实的开放性基础。

与 C/S 架构相比,B/S 架构也有许多不足之处,例如:

(1) B/S 架构缺乏对动态页面的支持能力,没有集成有效的数据库处理功能。

(2) B/S 架构的系统扩展能力差,安全性难以控制。

(3) 采用 B/S 架构的应用系统,在数据查询等响应速度上,要远远地低于 C/S 架构。

(4) B/S 架构的数据提交一般以页面为单位,数据的动态交互性不强,不利于在线事务处理应用。

3. C/S 与 B/S 混合软件架构

传统的 C/S 架构并非一无是处,而新兴的 B/S 架构也并非十全十美。由于 C/S 架构根深蒂固,技术成熟,原来的很多软件系统都是建立在 C/S 架构基础上的,因此,B/S 架构要想在软件开发中起主导作用,要走的路还很长。我们认为,C/S 架构与 B/S 架构还将长期共存,其结合方式主要有两种。下面,我们分别介绍 C/S 与 B/S 混合架构的两个模型。

(1) “内外有别”模型

在 C/S 与 B/S 混合架构的“内外有别”模型中,企业内部用户通过局域网直接访问数据库服务器,软件系统采用 C/S 架构;企业外部用户通过 Internet 访问 Web 服务器,通过 Web 服务器再访问数据库服务器,软件系统采用 B/S 架构。

“内外有别”模型的优点是外部用户不直接访问数据库服务器,能保证企业数据库的相对安全。企业内部用户的交互性较强,数据查询和修改的响应速度较快。

“内外有别”模型的缺点是外部用户修改和维护数据时,速度较慢,较烦琐,数据的动态交互性不强。

(2) “查改有别”模型

在 C/S 与 B/S 混合软件架构的“查改有别”模型中,不管用户是通过什么方式(局域网或 Internet)连接到系统,凡是需执行维护和修改数据操作的,就使用 C/S 架构;如果只是执行一般的查询和浏览操作,则使用 B/S 架构。

“查改有别”模型体现了 B/S 架构和 C/S 架构的共同优点。但因为外部用户能直接通过 Internet 连接到数据库服务器,企业数据容易暴露给外部用户,给数据安全造成了一定的威胁。

4. 应用实例

在设计 TSMIS 系统时,我们充分考虑到变电站分布管理的需要,采用 C/S 与 B/S 混合架构的“内外有别”模型,如下图所示。

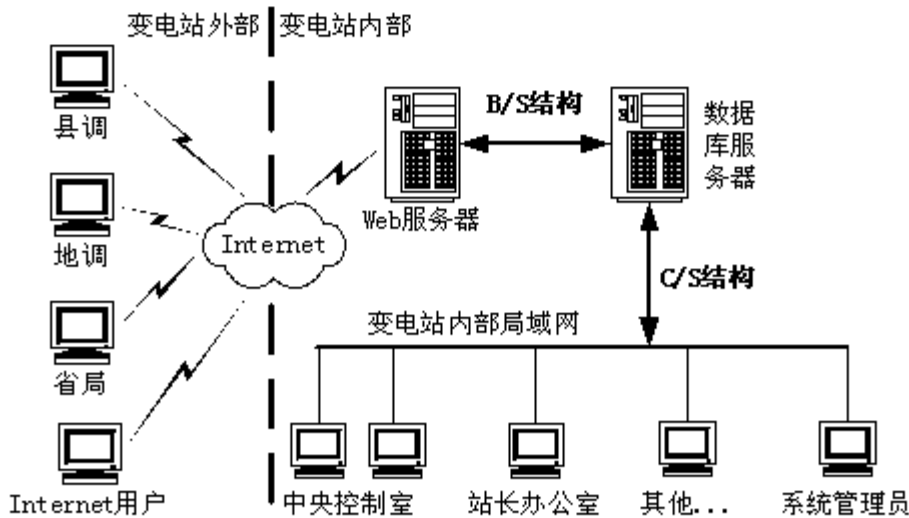


图 TSMIS 系统软件架构

在 TSMIS 系统中,变电站内部用户通过局域网直接访问数据库服务器,外部用户(包括县调、地调和省局的用户及普通 Internet 用户)通过 Internet 访问 Web 服务器,再通过 Web 服务器访问数据库服务器。外部用户只需一台接入 Internet 的计算机,就可以通过 Internet 查询运行生产管理情况,无须做太大的投入和复杂的设置。这样也方便所属电业局及时了解各变电站的运行生产情况,对各变电站的运行生产进行宏观调控。此设计能很好地满足用户的需求,符合可持续发展的原则,使系统有较好的开放性和易扩展性。

该系统已经在全国的多个变电站使用,用户反映良好。真正满足了变电管理朝无人、少人和一点带面发展趋势的需要,提高工作效率、增强准确性,对工作过程中的各种记录都能详实、准确地记载,减少大量手工重复录入,达到变电站无人、少人值班的目的。

实践证明,在软件项目的开发中,使用 C/S 与 B/S 混合架构,能节省开发和维护成本,使系统具有良好的开放性,易扩展性,便于移植等优点。

论信息系统的安全风险评估

摘要:

2005 年 3 月,我参加了某石化公司的实验室信息管理系统项目的开发工作,该系统作为该石化公司产品质量信息管理平台,将实验室的自动化分析仪器与计算机网络进行联结,实现自动采集样品分析数据,对样品检验过程、实验室资源进行严格管理,实现从原料进厂、生产、中间控制直至成品出厂的全过程质量数据管理,以及全公司范围内质量数据的快速传递与共享。我作为项目负责人,负责项目实施中的项目管理工作和系统投运后的运行维护工作。

为了做好系统的开发和应用,必须对系统将面临的安全风险进行评估。我在系统的安全风险评估方面采取了如下措施:分析现有业务流程和新系统信息流的安全因素,做好安全风险分析;建立安全风险评估标准,对安全风险评估分级、分类;在信息系统的各个阶段,反复对安全风险进行评估。系统在 12 月底通过验收,在两年多的运行期间,没有发生重大安全问题,系统建设、运行中的安全风险评估起了很大作用。

正文:

2005 年 3 月,我参加了某石化公司的实验室信息管理系统(以下简称 LIMS, Laboratory Information Management System)项目的开发工作, LIMS 主要实现四个方面的功能:实验室数据管理;实验室资源管理;实验室自动化仪器联结;检验数据 WEB 发布。LIMS 项目实施完成后, LIMS 将作为该石化公司产品质量信息管理平台,通过 LIMS 将实验室的自动化分析仪器与计算机网络进行联结,实现自动采集样品分析数据,按照 ISO/IEC 17025 实验室管理体系对样品检验过程、实验室资源进行严格管理,实现从原料进厂、生产、中间控制直至成品出厂的全过程质量数据管理,以及全公司范围内质量数据的快速传递与共享。

我从系统的可行性分析阶段就参与系统的调研工作,项目立项后作为 LIMS 项目的项目负责人,主要负责项目管理,同时负责项目的需求分析、系统集成、系统测试和系统投运后的运行维护工作。

系统拟采用 C/S 和 B/S 混合架构方式,后台数据库采用 Oracle 9i,前端客户端采用 Visual C++ 6.0 开发,WEB 端采用 ASP.NET 技术开发,B/S 和 C/S 模式均要支持三层结构。

不仅要从开发技术上要保证系统的信息安全,也要从管理上预知系统开发过程和运行过程中的信息安全风险。LIMS 本身服务于石化公司产品质量和生产安全,其系统本身更要有好的质量,包括完善的软件功能和投运后的较高的稳定性、可靠性。在网络环境下运行的信息系统,复杂性更高,在开发系统之前、开发过程中、系统运行时,都要注意信息系统的的风险。

信息系统的的风险,是指由于系统存在的脆弱性,人为或自然的威胁导致安全事件发生的可能性及其造成的影响。信息安全风险评估就是从风险管理角度,运用科学的分析方法和手段,系统地分析信息化业务和信息系统所面临的人为和自然的威胁及其存在的脆弱性,评估安全事件一旦发生可能造成的危害程度,提出有针对性的抵御威胁的防护对策和整改措施,以防范和化解风险,或者将残余风险控制在可接受的水平,从而最大限度地保障网络与信息安全。

信息安全风险评估是实现信息系统安全必要的步骤,通过信息安全风险评估,可以清楚业务信息系统包含的重要资产、面临的主要威胁、本身的弱点;哪些威胁出现的可能性较大,造成的影响也较大,提出的安全方案需要多少技术和费用的支持;分析出信息系统的风险是如何随时间变化的,将来应如何面对这些风险。

信息系统安全风险评估有三种形式,即自我评估、委托评估和检查评估。在 LIMS 系统中采用了自我评估方式。

在 LIMS 的实施与运行维护中,我首先确立 LIMS 的安全目标与策略,然后在风险分析中进行风险评估,在方案设计中风险接受度进行评估,在安全计划实施中进行系统测评,在系统运行与维护中进行日常检查和定期评估,安全风险评估贯穿于信息系统安全管理的全过程,具有极为重要的作用。

在 LIMS 的安全风险评估工作方面,我采取了如下措施:

一、分析现有业务流程和新系统信息流的安全因素,做好安全风险分析。

为了全面了解新建 LIMS 系统的安全需求,我们仔细分析了现有业务流程各环节的安全因素,并通过对新系统信息流的来源、传输、处理和存储等环节的分析,识别 LIMS 系统的安全风险,做好安全风险分析,为下一步的安全风险评估做好准备。

从 LIMS 数据的来源上看,基础数据来自于两种方式:手工录入;自动采集。通过各种化验分析仪器,产生了样品的分析数据,部分数据需要手工录入,部分数据通过采集器(如色谱采集器)进入系统,数据采集器的质量和化验员的操作水平会直接影响最后的分析结果,进而影响石化产品质量的判定,影响产品的出厂。部分化验分析仪器可以直接导出格式文本或 Excel 数据,这部分数据可以直接通过程序读取。数据收集过程的安全因素是必须逐个识别和分析的。

从 LIMS 数据的传输过程上看,网络设备(如 HUB、交换机、网线、光纤等)的质量是重要的安全因素,由于化验分析室中有毒、有害气体对网络设备腐蚀严重,一些重要的设备必须集中到工作环境相对较好的房间,在综合布线时必须考虑。每一个色谱数据采集器都要使用一个 IP 地址,部分化验分析仪器也要使用 IP 地址,IP 地址的管理不仅要考虑办公微机,也要考虑数据采集设备。网络安全是 LIMS 系统的重中之重。

从 LIMS 数据的处理和存储上看,许多分析计算要在 LIMS 客户端上完成,服务器上存放中间计算结果和最终结果(如审核过程和质量合格证),LIMS 查询机要应用在各生产车间和油品罐区操作室,操作员的素质和 LIMS 查询机的安全设置也是不能忽视的安全因素。

新系统投用后,业务流程会发生改变,比如会取消部分纸质台帐,审核方式也不再是人工审核,为了保证系统的正常运行,需要制定事故预案、应急措施,并要有系统运行管理部门。业务流程改变引发的管理方式改变,会造成许多不确定的安全因素。

二、建立安全风险评估标准,对安全风险评估分级、分类。

识别安全风险之后,就要进行风险评估。在风险评估过程中全面评估资产、威胁、脆弱性以及现有的安全措施,分析安全事件发生的可能性以及可能的损失,从而确定信息系统的风险,并判断风险的优先级,建议处理风险的措施。

风险评估是分级防护和突出重点的具体体现,有些风险可以接受,有些风险则要消除。每个风险的解决成本是不一样的,必须综合考虑。不存在绝对的安全,也不可能做到绝对安全。安全是风险与成本的综合平衡。盲目追求安全和完全回避风险是不现实的,也不是分级防护原则所要求的。

我们参照《信息安全风险评估指南》,结合 LIMS 项目的实际情况,建立安全风险评估标准,用定性和定量的方法为涉及到的安全风险进行评估,可以分级、分类,确定风险等级和优先风险控制顺序,在评估结论中指出所有安全隐患,并给出解决方案建议。在对 LIMS 查询机的安全风险分析中,我认为操作员有可能通过 WEB 上的恶意访问和在 LIMS 查询机上的非法操作影响 LIMS 整个系统的运行,对几个解决方案进行了分析,认为在普通微机上进行一些安全设置,屏蔽掉一些多余的功能,比较可行,成本也低,而采用无盘工作站方式虽然安全效果更好,但成本太高,参照评估标准,选择最优的解决方案。

数据库服务器的安全风险级别较高,我们采用了双机热备软件,通过镜像引擎将数据由专用的直联线进行实时复制,当一台服务器发生硬件或软件故障时,自动启用另一台服务器,保证数据存储的安全和 LIMS 的长周期运行。

优先解决级别高的风险,是安全风险评估分级的目的。

三、在信息系统的各个阶段,反复对安全风险进行评估。

信息系统的安全是一个动态的复杂过程,它贯穿于信息系统的整个生命周期,对信息系统进行不断的安全风险评估是十分必要的。在 LIMS 系统规划设计阶段,通过风险评估明确系统建设的安全需求和安全目标;在验收阶段,通过风险评估验证信息系统安全措施能否实现安全目标;在信息系统运行维护阶段,定期进行风险评估,检验安全措施的有效性以及对安全环境变化的适应性,以保障安全目标的实现。

LIMS 系统在 12 月底通过验收,正式投入运行,在两年多的运行期间,没有发生重大安全问题,LIMS 系统的建设、运行中的安全风险评估起了很大作用,明确的安全需求,合理的安全体系,严格的安全评估,灵活的安全措施,是系统安全运行的保障。

风险评估在信息安全保障工作中具有不可替代的地位和重要作用,我们在 LIMS 安全风险评估中采用的评估标准来源于多年信息系统实施和运行中积累的行业经验,并不是国家标准,还需要进一步完善,还存在着一些问题需要加以解决,如建立完善的工作机制、提高评估能力和技术水平等。信息安全风险评估是一项长期的工作,还有很多问题需要在实践中不断摸索,在理论中深入研究加以解决。

论信息系统的安全体系

摘要:

2005 年 2 月,我参加了某水库管理信息系统项目的实施。通过系统的实施和运行,实现防汛、供水、发电、闸门监控、水文等各种数据的采集、分析、存储,并通过网络及时地向有关部门汇报,以便相关领导进行调度指挥,为领导决策提供大力支持,为业务人员办公提供服务。系统的应用将有效提高某市政府水库管理所的工作效率。

我作为该项目的项目负责人,主要负责项目管理,同时负责项目的需求分析、系统集成、系统测试和系统验收的工作,并负责系统投用后的运行维护。为了确保项目顺利的实施和安全的投入运行,我从软件、硬件、运行环境、用户四个角度对系统的脆弱性进行了分析,从实体安全、操作系统平台的安全、数据安全、访问控制的安全性、管理制度的安全保护五个方面对水库管理信息系统的安全体系进行了分析,最后探讨了安全体系设计中的方法和原则,参照现行信息系统安全架构体系,分析了本系统中安全体系的不足之处。

正文:

2005 年 2 月,我参加了某水库管理信息系统项目的实施。以某市政府某水库管理所为核心建立中心管理局域网,通过水库管理信息系统全面地进行雨水情等信息的收集、输入、修改、查询,并实时动态显示有关信息,通过人机交互进行洪水预报、水库安全运行分析、调度方案制订、调度方案评价与优选、调度成果管理、防洪、发电、灌溉工程信息管理、调度控制、系统管理等。根据实时和定期水、雨、工情准确分析防洪、水库安全及兴利形势,形成调度预案,供各级调度指挥人员参考,并通过专家经验加以修正,最终形成调度方案,供水库管理机构决策,供执行人员实施。上级主管部门可通过 internet 实现与水库管理所的信息共享和信息传输,其他单位(在授权许可下)以同样的方式实现对水库管理所的信息访问。系统的应用将有效提高水库管理所的工作效率。我作为该项目的项目负责人,主要负责项目管理,同时负责项目的需求分析、系统集成、系统测试和系统验收的工作,并负责系统投用后的运行维护。

系统采用 C/S 和 B/S 混合架构,其中 C/S 部分实现各部门管理功能子系统,以数据采集和数据处理为主;B/S 部分则实现存在交叉业务的部门管理子系统、部分共享数据的网上发布和数据的网上查询、浏览。所有的数据都存放在后台的 Oracle 数据库服务器上,进行统一管理和维护。

水库管理信息系统采集和处理数据的过程均在局域网内,内部用户可以查询中间结果和发布审核过的信息,经过授权的外部用户只能通过 internet 查询已经发布的信息。所以本系统是一个相对独立的、封闭的专用信息系统,绝大多数的恶意攻击是来自内部的,水库管理的工作性质决定了本系统信息的机密性显得尤其重要,因此,除了防止来自 internet 外部的攻击外,如何制定内部安全防范体系的问题也必须解决。

信息安全是指防止信息被故意的或偶然的非法授权泄露、更改、破坏或使信息被非法辨识、控制。信息安全包括操作系统安全、数据库安全、网络安全、病毒防护、访问控制、加密与鉴别等七个方面。如果采集的水库管理数据不正确,或者在上传和发布的过程中被篡改、破坏,将影响到防汛、供水、发电等多项决策,信息安全工作做不好,信息系统将无法正常使用和应用。如果系统分析时不充分考虑信息安全因素,将为系统将来的运行留下隐患。

一个完整的安全体系包含的内容有:风险管理;行为管理;信息管理;安全边界;系统安全;身份认证与授权;应用安全;数据库安全;链路安全;桌面系统安全;病毒防治;灾难恢复与备份;集中安全管理等。作为一个政府部门的信息系统,水库管理信息系统应建立一个以内部控制机制为核心的安全防护体系,以降低其带来的风险。从系统的开发到运行和完善,在各个不同阶段,都应把安全问题作为重中之重,并贯穿于整个过程之中。要确保水库管理信息系统的安全可靠,离不开信息安全技术的应用,我对系统的脆弱性和安全体系进行了如下分析。

一、系统的脆弱性分析

水库管理信息系统是一个包括计算机软硬件、通信设施并与监控设施和组织高度集成的人机系统,不可避免面临着来自各方面的潜在威胁。软件、硬件、运行环境、用户各方面都可以形成系统的脆弱性。

软件方面:水库管理信息系统软件的开发人员的专业背景、业务能力、对水库管理信息系统的理解深度等都会对系统开发的成败产生重大的影响,系统开发过程主要以手工为主,没有成熟的软件产品,是造成水库信息软件缺乏可靠性和安全性的重要因素。

硬件方面:除了计算机设备和网络设备,系统用到的水位探测仪、监控设备、数据采集设备,均由大量的电子元件和芯片组成,而芯片和电子元件的老化和一些安插件之间的接触不良都会造成系统的失灵,系统的采样点和监控点分布在野外,受自然环境的影响很大。

运行环境方面:水库管理信息系统中使用传感器技术和数据采集技术,将野外遥测站点的数据传到 10 多个数据接收工作站,数据经过处理后再进入调度室,要经过多个部门和多个环节。即便环境能完全达到信息系统的安全技术要求,但因整个系统分散在各个部门,也就很难避免因为环境因素而导致意外事件的发生。

用户方面:系统的用户数量多,层次也多,能否充分发挥信息系统的作用,并能安全正常的运行系统,这与使用人员对计算机的基本知识了解程度、在系统开发过程中与开发人员配合程度、投入使用后是否按规范操作、对系统运行所需的数据收集是否及时和充分,是否有假的、错误的数据采集输入都密切相关。系统在为水库管理所的上级和各部门提供数据共享时,也可能为入侵者、信息偷窥者提供了方便之门。

二、系统的安全体系分析

设计安全体系时要根据实际情况,综合考虑各种因素。

实体安全方面:水库所处的地理位置在夏季会受到雷电灾害的影响,在野外的雨水情遥测设备,如摄像头、云台、信号采集器必须有防雷设施,办公楼内外的网络设备(交换机、HUB 等)也要有防雷功能,对信息系统实体的破坏,不仅可以造成巨大的经济损失,也会导致系统中的机密信息数据丢失和破坏

操作系统平台的安全方面:作为水库管理信息系统的支撑部分,操作系统安全的内容包括保密性、可靠性和抗干扰性等几方面内容。在选择操作系统时,既要考虑操作系统的实用性与可靠性,又要考虑到操作系统的安全性。操作系统还应该安装防火墙及病毒查杀程序,以提高整个信息系统的可靠性与安全性。

数据安全方面:水库管理信息系统的数据安全性,除了一般由存和取的控制来保证外,还要加强对数据库的管理和对数据采取必要的加密等手段,以防止信息泄漏。

访问控制的安全性方面:通过对用户进行系统功能授权,即系统每一功能,只有被授权的用户才能使用,未被授权的用户无法使用。系统中的每一个用户在完成其工作时,只应拥有最小的必要的系统功能,使出错或蓄意破坏造成的危害的概率降到最低。水库信息系统的功能只对授权用户是可视的,对非授权用户是不可视的,这样可预防蓄意破坏的发生。

管理制度的安全保护方面:在开发人员和使用者中建立完善的安全制度,并使其充分认识计算机系统安全的重要性,自觉执行安全制度,从而形成对水库管理信息系统的一个管理保护层。

水库管理信息系统安全体系的分析应贯穿系统的开发、实施、运行各个阶段。作为一个政府部门的信息系统,安全性永远是第一位的。水库管理信息系统在 2006 年 3 月正式投入使用后,安全平稳的运行了 1 年多,在政府部门的水库管理工作发挥着作用,这与安全体系设计时充分的分析是分不开的。安全体系的质量决定着安全工程的质量,我们在安全体系设计之后,聘请了第三方专家对体系的质量进行评审,对评审结果进行论证,对发现的不妥之处及时修正,做好安全体系的评估工作,保证安全体系的质量,从而保证信息系统开发和运行的各阶段的质量。

安全体系的设计和实施要有一定的灵活性、适应性,还要遵循风险和成本的平衡性,安全体系服务于信息系统的应用,当安全体系的代价大于信息系统应用的价值时,就不再具有可行性。安全体系的分析要具有前瞻性,在系统分析时就要考虑系统运行、维护时的各种安全因素,如果系统投用后才发现安全体系不能保障系统的正常运行,会造成维护成本的剧增甚至信息系统的失败。在系统建设之初就考虑系统安全对策,比等系统建设好后再考虑,不但容易,而且花费也少得多。

现行信息系统安全架构体系以信息安全程度从低到高可分为: MIS+S; S-MIS; S2-MIS。水库管理信息系统的安全体系属于 MIS+S,是一个低级的信息安全保障系统,没有对水库信息系统作任何改变,为防止自然灾害、病毒、黑客等增加一些安全措施和安全防范设备,提高了水库信息系统的**安全强度**,基本满足了安全需要,实施成本较低,但并没有从根本上解决信息系统的安全问题,这也是目前受安全成本和应用效益约束的实际情况,当以后水库信息系统二期工程开展时,我们会考虑往较高层次的安全体系上发展。

论行业应用软件系统的开发规划

【摘要】

本人在一所高校信息技术中心工作，2005 年应多个学校和校领导的要求，对以前开发的较独立的十多个学校信息化软件系统进行升级改造，使得各系统之间能够相互进行数据共享，配合工作。所进行的软件项目，与高等院校的学生招生、收费、教学、管理和服务密切相关，具有很强的行业特征，需要使用系统的师生员工（用户单位）、高校管理专家和项目开发人员间良好的分工协作；与常规的通用软件开发过程相比，具有明显的差异，主要表现在系统的分析、设计、实现、测试和维护均带有明显的行业特点，行业应用软件的成功开发必须遵循行业规范，正确使用专业术语，及时与专家和用户进行交流。

本人在项目中主要负责需求分析和软件系统设计，本文主要从资源互补、角色定位、相互协作的角度讨论行业软件系统的开发规划工作，我在实践中发现，行业应用软件系统的成功开发是从其规划到诞生起就应从行业中汲取各种营养，具有行业特征、行业味道，才能满足行业应用的需求。

【正文】

作为一所高校信息技术中心的技术骨干和技术负责人，高校内部信息化建设受到我们时常的关注。十多年来，我们与兄弟院校、知名软件企业合作进行了多项与高校信息化建设相关的软件系统开发工作，有学生收费管理系统、学生学籍管理系统、学生成绩管理系统、学生选课系统、学校图书借阅系统、学生奖惩管理系统、学生应聘就业管理系统、学校后勤服务系统、教材管理系统、教师业绩考核系统、学校办公自动化系统等十多个软件项目，覆盖了学校招生、入学收费、教学管理、学生管理、后勤服务和学生毕业的整个过程，这些项目，是我们与不同兄弟院校技术部门、不同企业合作的结果，被省内 30 余家高校使用，它有力地促进了我们与兄弟院校的信息化建设，提高办学的效益和办事效率。但随着系统的增多，也暴露出了一些矛盾问题，制约了相关学校的发展和信息化进程：

1、这些软件项目是我们与不同兄弟院校、不同企业合作开发的，也有些是独立开发的，信息难于相互共享。

2、这些软件项目开发于不同时期，期间技术中心人员变更较大，不同阶段、不同时期采用的技术不同，各个系统形式多样，有单机的、C/S 网络结构的、B/S 网络结构的和混合结构的。使用极不方便。

3、经过近年来的高校大发展进程，各校在管理上有了较大变化，原来的一些软件已经不适应学校的发展。

因此，在 2005 年我校领导参加省教育厅关于学校信息化建设工作会时，省内 20 余家高校提出让我们牵头进行系统的升级改造工作，并承诺在资金和人力方面给予支持，引起了省教育厅领导高度重视。我校领导将这一艰巨任务以政治任务的形式布置到我们信息技术中心，限时完成。

项目于 2005 年 5 月启动，成立了由七个人组成的项目组，作为信息技术中心的技术骨干和技术负责人，我参与了该项目的总体架构组，主要负责需求分析和软件系统设计，并完全负责该项目的管理。

虽然我们团队的大部分人长期工作在高校技术部门，但对于高校教学、管理和服务中的专业知识掌握并不多，高等教育行业的专业性特强，由开发人员现学领域知识是不现实的，这就需要用户和行业专家的协助。从相关部门聘请行业专家是明智之举。

作为行业性很强的应用软件，它和通用应用软件系统开发有很多不同，我觉得主要有以下几个方面：

一是需求分析阶段专业业务难把握、难摸透，管理目标和专业目标难一致。

二是设计阶段，设计规范难统一，难细化。

三是开发阶段，开发阶段的测试往往由于专业背景各系统联系紧密，难确认测试。

四是实施阶段，实施阶段的业务人员对系统认同差，沟通复杂。

五是整个过程涉及到过多的行业习惯，许多方面不和其他开发雷同。

聘请行业专家做项目组的顾问是应用软件开发过程中较好的策略。行业专家主要是作为领域知识源，提供技术文件和已有应用。在高校信息化改造建设这个项目中，我们从有关部门和其他高校聘请了各分系统的资深专家作为项目的顾问，获取到《高等学校数据规范标准》等技术文件和部分以前编写的源代码，该源代码是多个独立运行的、凌乱的、较小的系统，经过整理后可以用到新系统中，这为领域模型和设计模型的建立打下了良好的基础。

用户是应用软件目前和未来的需求提供者，其对项目成败的重要性是显而易见的。OOA 过程是从对系统将被使用的方式的理解开始的，在需求诱导时需要从用户的观点对系统建模，需要项目组和用户共同确定使用场景和定义系统的功能和运行需求；需要用户提供清楚而无二义性的终端用户和系统如何相互交互的描述；需要用户提供确认测试的基础。在这外项目中，我们与用户充分合作，请用户给我们讲解高等学校有关教学、管理和服务的常识、过程、功能等，将用户作为项目组成员，作为每个阶段的技术评审的主要成员。

开发者是应用软件系统的建设者，需要从用户方进行需求诱导，建立领域分析模型；需要将分析模型变换为设计模型，建立软件的构造蓝图，需要完成编码和测试。由于应用软件具有行业背景，在建立分析模型是需要从各种渠道了解行业知识；在建立设计模型是需要使用逆向工程技术从现有的系统中得到关键的算法。在进行测试时需要使用用户提供的具有行业特点的真实的测试数据。开发人员需要随时在涉及到行业知识时向用户或行业专家请教，必须克服在开发一般应用软件时的问题讨论在相对封闭的范围内进行的特点。

与常规的通用软件开发进行比较，专业背景的应用软件开发过程及各个环节的主要差异有：

（1）在计划阶段，具有行业背景的应用软件系统与普遍应用软件的目标与动机一般是不同的，普遍应用软件以给本公司或部门带来经济上的收益为目的，而行业应用软件以更好的完成某项职能为目的。高等教育行业的应用系统开发目的是为了对高校内教育教学、管理和后勤服务进行有效管理，使师生方便、快捷、充分地享受高校资源，提高办学水平和能力。这就要求系统方案的制定必须符合行业的特征。基于高等教育行业的信息化建设改造项目，因其牵扯高校内部教务、人事、财务、学生管理、招生、就业、团委、后勤等众多部门，针对不同院校，所管理学生的学历层次、学习类型、办学形式等五花八门，管理方式也有差别，这就增加了应用软件开发的复杂性。

（2）在需求分析阶段，行业背景的应用软件与普通应用软件相比一般需求比较明确、相对固定和有章可寻。必须获得行业的各种技术规范、数据共享资料及现行系统的运行数据。在此基础上划出应用软件的作用范围和与周围环境进行数据交换的接口，在类的划分协作模型和对象-关系模型。

（3）在设计阶段，行业背景的应用软件系统与普通应用软件相比，从用户界面设计、数据管理、任务管理到控制机制均要表现出行业特征。我们在该项目开发时，邀请有专业知识的用户参加到项目组实施的关键环节中，将设计模型在该行业专家作用户中进行解释。确保设计模型与领域模型保持一致。数据和任务管理的设计策略需要参考与《高等学校数据规范标准》相类似的行业标准；通过创建每个操作的过程表示类属性的数据结构而完成对象设计，数据结构和算法必须符合行业需要，使用对象间的协作和对象关系完成消息设计。在评审设计模型时需要行业专家和用户的参与，从行业的角度提出修改意见。

（4）在集成测试阶段，行业背景的应用软件系统与应用软件相比，一般更强调进行 β 测试和进行新旧系统对比，需要使用行业的数据作为测试数据，测试案例需要由用户和专家参与制订，行业数据需要由用户和专家提供，测试结果需要由用户和专家确认和解释。由于针对高校信息化系统建设牵扯到众多的部门、管理方式和类别多、数据共享关系复杂、数据量非常大，因此系统测试需要在真实的数据环境中进行。我们在测试高校信息化系统建设（改造）应用软件时使用的是真实的数据，从数据的采集、传输、储存、处理和显示等的各个环节均使用的是高校各子系统的真实数据，根据测试结果和用户意见进行修改。

从上面的几个主要差别可以看出，具有行业背景的应用软件的开发需要行业中的各类标准、专业知识、现有相关系统合人员参与到开发过程中，要使行业特点与信息技术有机结合，在项目管理上还需要包含行业内的相关的人和物。

在该项目的开发过程中，人员、设备、资料等的方方面面的资源均是由项目组与高校信息化系统共享的，在知识和思想上充分融合，经过双方的共同努力不仅开发出了满足高等教育行业需要的应用软件，还丰富了双方的知识面，拓展了眼界。

在实践中我发现，行业应用软件的开发必须遵循行业规范，正确使用专业术语，及时与专家和用户进行交流，要从其规划到诞生起就从行业中汲取各种营养，使其具有行业特征、行业味道。只有这样，行业应用软件的开发才能取得成功，才能被该行业人员认可、应用，才能满足行业应用的需求。

论基于 UML 的需求分析

【摘要】

UML 是集多种面向对象方法的优点于一身的统一建模语言, 通过 UML 可以解决开发过程中存在的一些问题。包括解决人员交流的障碍, 响应需求的变化, 利于构件的复用, 保证软件项目开发周期等。采用 UML 进行需求分析, 主要是通过用例模型来捕获和组织用户的需求, 通过用例建模, 描述对系统感兴趣的外部角色及其对系统 (用例) 的功能要求。

2006 年 5 月, 我参与了某区贸工局的电子政务系统的开发。在需求分析过程中采用了基于用例的需求分析方法, 取得了良好的效果。在用例建模过程中, 通过识别系统参与者, 合并需求获得用例并绘制用例图, 进行用例分解及细化用例描述等步骤, 及各步骤间的循环反复, 成功完成了需求分析, 需求描述也得到用户的认可。当然, 由于使用该方法还不很成熟, 各种方法及工具的集成度还不高, 未能充分发挥其作用。在项目中, 我担任系统分析员, 主要负责系统分析和系统设计工作。

【正文】

2006 年 5 月, 我参与了某区贸工局的电子政务系统的开发, 项目历时七个月, 于 2007 年 1 月正式上线。项目组成员共 7 人, 在项目中, 我担任系统分析员, 主要负责系统分析和系统设计工作。

区贸工局已有近十年的信息系统使用经验, 在本系统开发时, 该局除一套采用 VB+SQL Server2000 开发的二层 C/S 结构的核心业务管理系统外, 还有多套业务系统和数据交换系统, 主要有: 外资审批管理系统、加工贸易电子数据交换平台、加工贸易联网监管电子数据交换系统以及电子公文交换等。上述各系统基本是相互独立的, 只在数据库端实现初步的数据共享, 但应用的集成性很差。

区贸工局的电子政务系统是一个基于知识管理的全新的集成的管理系统, 其应用范围涉及办公自动化、审批业务管理、档案管理、数据交换、互联网站等各个方面。该系统由门户网站、办公自动化和业务管理三个子系统构成。与原有的业务系统相比, 区别主要体现在三个方面: 一是全新的体系结构; 二是集成性, 全面集成原有的各业务系统及数据交换系统; 三是以知识管理为主要特征的应用层次上的全面提升, 对业务审批的全过程进行监督管理, 引入审批要点对相关业务进行智能辅助审批。

在核心的业务管理子系统的开发过程中, 考虑到传统结构化开发方法的局限性和软件本身的易复用性、易扩展性和可维护性, 以及可能面对的需求变化, 我们在开发时采用了面向对象的方法。UML 是集多种面向对象方法的优点于一身的统一建模语言, 它使用统一的表示法, 呈现一致的风格, 通过 UML 可以解决开发过程中存在的一些问题。首先, UML 解决了人员交流的障碍。它提供了一套通用的思维方式和交流的语言, 既有助于分析人员与用户的交流, 又有助于分析人员与设计人员的交流。其次, 利于响应变化。分析人员可以将对象作为构筑系统的基本单位, 将容易发生变化的属性和操作封装在对象之内, 对象之间通过接口联系, 使得需求变化的影响尽可能限制在对象的内部。再次, 便于构件的复用。集成 UML 思想构建的系统模型能很好地支持软件复用, 类可以派生子类, 类又可以产生实例对象, 而对象具有封装性和信息隐蔽性, 这就实现了对对象类的数据结构和操作代码的软构件的复用。最后, 因开发人员的方法、工具及经验的差异, 往往造成较大型或是较复杂的软件项目开发周期得不到保证。若运用 UML 进行系统的分析设计, 利用规范化的表达方式及优秀的 CASE 工具, 问题可以得到较好的解决。

采用 UML 进行需求分析, 主要是通过用例模型来捕获和组织用户的需求, 通过用例建模, 描述对系统感兴趣的外部角色及其对系统 (用例) 的功能要求。在用例建模过程中, 首先是识别系统参与者, 然后合并需求获得用例, 并绘制用例图, 最后进行用例分解及细化用例描述。

1. 识别参与者。我们从三个方面分析系统的参与者, 一是系统的直接使用者, 二是系统的管理维护人员, 三是外部相关的软硬件系统或人员。

系统直接使用者即是使用业务系统的相关业务人员。根据业务岗位, 分别有收文、录入、打印、窗口 (一级) 审批、科长 (二级) 审批、局长 (三级) 审批等角色, 另外还有派单 (针对个别不由系统自动分派任务的情况)、审批检查、查询统计和科室管理 (执行科室内部人中角色的分派、业务智能审批定义等) 角色。

系统管理维护人员, 主要进行科室管理员的角色分派以及系统相关初始数据的设定等工作。

外部相关的软硬件系统, 即与业务系统进行交互的一些外部系统。包括了外资审批管理系统、加工贸易电子数据交换平台、加工贸易联网监管电子数据交换系统、电子监察系统、门户网站、办公自动化系统以及系统时钟等, 其中门户网站和办公自动化系统虽是整个电子政务系统的一部分, 但相对于业务子系统来说属于外部系统。外部相关人员即申办各类业务的企业办事人员。

2. 合并需求获得用例, 绘制用例图。

在确定了参与者后, 分析每一个参与者希望系统提供什么样的功能, 为参与者确定用例。之后画出用例图, 通过用例图描述系统包含的参与者、用例以及两者之间的对应关系。

3. 用例分解及细化用例描述。

绘制初步的用例图后, 还需对用例予以细化, 编写用例规约, 或是根据情况进行用例分解。用例规约规定了系统需要完成哪些步骤才能实现用例的功能, 主要包括前置条件、后置条件和事件流。前置条件指明了执行用例之前系统须处于的状态; 后置条件指明用例执行完毕后系统可能处于的一组状态。事件流描述用例执行的步骤, 它又包括基流、分支流和替代流。基流描述用例执行的基本步骤, 没有分支和选择; 分支流描述用例在执行中根据不同条件或不同选择而可能执行的步骤; 替代流描述用例在执行中因异常或偶尔发生的一些情况而执行的相应替代步骤。其中基流是必需的, 分支流和替代流是可选的。

整个用例的分析及用例图的绘制, 需循环执行上述各步。通过参与者及其需求识别用例, 通过用例 (图) 反过来印证识别出的参与者; 对用例进行细化描述, 如该描述较复杂则对用例进行分解。这样识别参与者和识别用例交替进行, 并结合用例的细化与分解, 直到识别出的用例能够涵盖用户需求, 且用例的细化粒度以在用户较容易理解的范围。下面以较典型的“外商投资企业设立”模块的用例图加以说明。

首先根据参与者及相关业务需求, 识别出“业务收文”、“业务审批”、“业务指派”、“信息录入”、“查询”、“数据写监察系统”、“外资审批管理系统数据读写”等七个用例。

接着又对照系统参与者检查每一个用例, 我们发现在“业务收文”漏了与门户网站子系统的交互, 即企业先在网站预录入资料的收文受理没有考虑, 因此加入了“门户网站子系统”这一参与者, 并添加相关事件 (分支) 流。

然后, 根据用户需求, 对部分用例进行细化与分解, 从“业务收文”中又进一步分出“补交资料受理” (即首次收文时资料不全或不全要求时企业补交资料的受理), “信息录入”中进一步分出“企业资料存档” (即企业负责人签名等若干重要原始资料的扫描存档), “业务审批”细分为“一级审批”、“二级审批”、“三级审批”, “一级审批”当中再分解出“补交告知”。“数据写监察系统”用例的事件流则最复杂, 分别与“业务收文” (发送“受理”状态及可选的“补交受理”状态)、 “一级审批” (发送“承办”状态及可选的“补交告知”状态)、 “外资审批管理系统数据读写” (在外资系统打印批文后, 发送“审核”、“批准”、“办结”状态) 等三个用例相关。

我们使用基于 UML 的需求分析方法, 取得了比较好的效果, 特别是相对于传统的需求分析与描述方法其优点是明显的。但由于我们使用该开发方法还不很成熟, 在开发过程中也出现了一些问题。一是 UML 各图形的组合使用问题。用例分析的方法和用例图无疑是需求分析的有力工具, 但光是进行用例分析还不足够, 如能很好地结合类图和活动图等图形, 则会对需求的分析和描述起到很好的辅助作用。另外就是 UML 与相关工具及开发方法的组合使用问题。UML 并不是一套独立的方法或工具, 要充分发挥 UML 的效用, 还须结合统一开发过程以及 ROSE 等相关 CASE 工具, 而在此方面我们还有明显的不足。由于开发大型项目较少, 因此还很少使用统一开发过程, CASE 工具的使用也还未达到系统化, 这都限制了 UML 方法的效果。另外, 由于传统的结构化开发方法的思维习惯, 我们在用例识别过程中有时还会按照 DFD 图的思维进行思考。

针对在项目开发过程中暴露出的问题, 我们以渐进的方式改进我们的开发过程, 向统一开发过程靠拢; 改善开发环境, 逐步购置、逐步消化使用 CASE 工具; 提高个人的开发水平, 深入学习掌握面向对象思想和方法及 UML 等各类相关 CASE 工具。我们不能指望通过一两个项目就完全掌握相关工具和方法, 不论组织还是开发者个人, 使用先进方法和工具的过程都是一个循序渐进的过程。

论信息系统架构的设计

摘要

我在一个软件企业从事软件系统架构设计工作，2005 年 4 月，我公司承担了某高校的应用集成项目，该校领导决定投资建立一个可扩展的统一集成平台，以解决学校信息系统中复杂、分散、异构的数据信息之间的交换、相互转换、共享等问题。

为了集成已有的系统，保护用户投资，同时，又要使已有的系统之间能够通信，使已有的系统与新开发系统之间也能够通信。在该项目中，我们采用中心辐射型消息代理技术，将中心辐射型集成模型引入到高校应用集成，结合相关标准，建立了一个适应于 IT 技术发展的教育应用的可扩展集成架构。

在中心辐射集成架构中，消息系统具有高度可扩展性，容易与其他系统进行集成，对于异构系统的集成效果显著。该项目完成至今已接近 1 年，从运行的效果来看，达到了项目的预期目标。项目验收时，得到了同行专家和该大学领导及有关人员的好评。

正文

经过 20 年的信息化建设，我国高校的信息化工作取得了很大的发展，一些高校建立了多个应用系统。例如学籍管理系统、档案管理系统、办公自动化系统、财务管理系统、设备物资管理系统、劳资管理系统、图书馆管理系统等。由于这些系统在不同的时期开发，运行于不同的系统平台，采用了不同的技术和不同的标准规范，导致这些系统都是一些“信息孤岛”，系统之间的数据共享和交换存在问题。同时，高校对信息化的需求又是不断扩展和发生变化的，这决定了任何一家厂商都不可能提供一所高校所需要的所有产品，高校需要采购不同厂商的软件产品。因此，在同一高校环境下，集成不同厂商的应用软件是高校信息化实施过程中必然要面临和解决的问题。

我在一个软件企业工作，2005 年 4 月，我所在的公司承担了某高校的应用集成项目，该校领导决定投资建立一个可扩展的统一集成平台，以解决学校信息系统中复杂、分散、异构的数据信息之间的交换、相互转换、共享等问题。该校已有的系统主要有办公自动化系统、财务管理系统、设备物资管理系统、图书馆管理系统和教务管理系统，要求新开发招生管理系统、就业管理系统、劳资管理系统、人事管理系统和科研管理系统。我有幸参加了该项目的开发，并担任架构师职务，主要负责系统架构设计工作。

通过系统分析，我们发现该高校信息化建设存在以下几方面的问题：

(1) 目前，各系统厂商各自为政，致使学校缺乏一套通用的数据标准。

(2) 现有各系统之间主要依靠数据的直接共享达到集成的目的。在系统之间进行信息交换的过程中，被交换信息的安全性没有保障。

(3) 部分系统之间已经进行了紧密集成，增加了扩展教育信息系统的难度，某一个系统的调整或维护会影响整个学校其它系统的正常工作。

(4) 现有系统的集成接口数量众多，难以维护。随着系统数量的增加，接口数量成比例增加。多家厂商所开发的系统之间的协调，异构平台之间的信息共享比较困难。

基于该校信息化存在的上述问题，在我的组织下，我们召开了架构设计会议，会议由我公司核心技术人员和该高校有关计算机专家参加。在这次会议上，确定了一些基本原则。会议认为，应用集成的架构必须体现遵循以下原则：

(1) 可扩展性：使得高校应用集成可以快速进行，有较强的通用性，各系统之间属于松散耦合，保证教育信息系统的可持续性发展。

(2) 标准化：采用教育行业的信息交换标准进行信息集成。

(3) 安全性：充分考虑高校应用集成过程面临的安全问题，各应用系统之间不直接进行数据共享或交换，采用基于消息的方式进行集成。

(4) 保护现有投资：高校应用集成必须考虑已有系统，充分发挥现有系统的作用，保护现有投资。

为了实现可扩展性强、安全性高的应用集成，经过研究和实践，我们设计了一个基于国家教育部标准的中心辐射型（hub-and-spoke）集成架构，该架构既包括数据集成，也包括应用集成。

我们采用中心辐射型消息代理技术，将中心辐射型集成模型引入到高校应用集成，结合相关标准，建立一个适应于 IT 技术发展的教育应用的可扩展集成架构。该架构使各异构系统之间的信息交换采用消息传递的方式，系统之间的通信只取决于消息发布和消息订阅，并不需要知道系统的位置，以及具体做什么和怎么做。每个系统只关心各自订阅的消息和需要发布的消息。中心辐射型集成架构如图 1 所示。

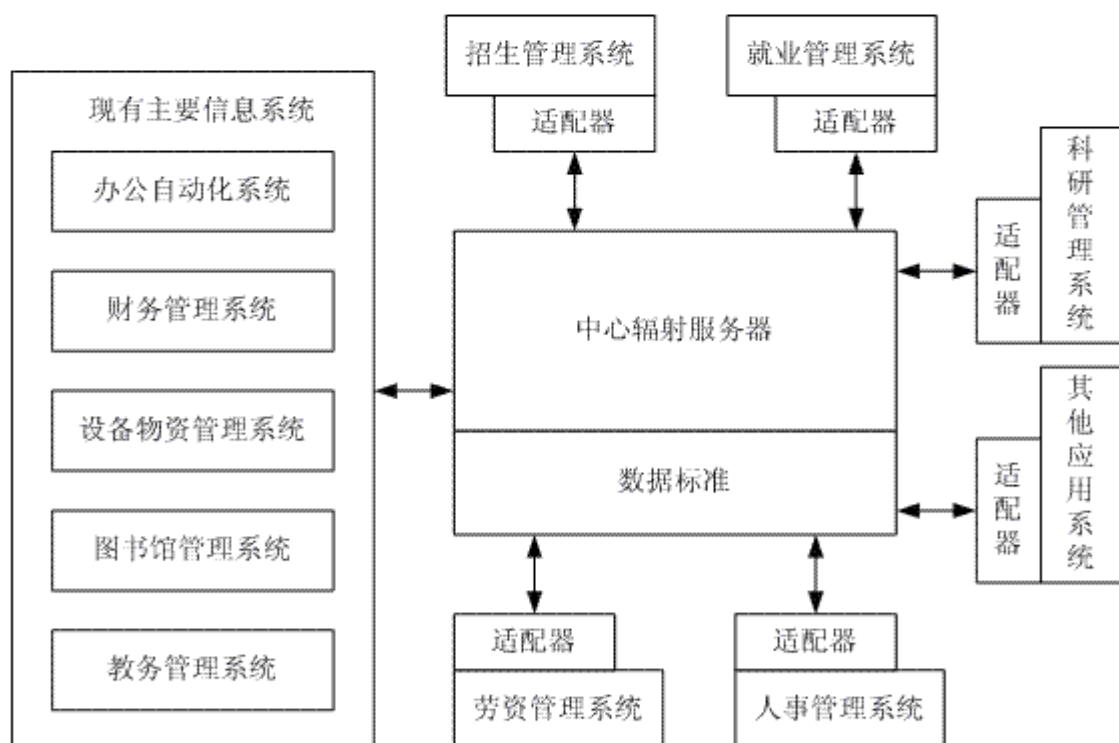


图1 中心辐射型集成架构

模型中各辐射与中心服务器之间的消息传递直接采用标准的消息格式，任何需要接入的应用产品（一个辐射）只要将自己的接口充分暴露给中心服务器即可。消息在中心服务器内既可以采用消息队列方式进行调度，也可以采用消息代理的方式进行调度。各应用系统之间基于发布与订阅方式进行消息交互，各应用系统以广播的方式发布标准消息，消息进入中心服务器，中心服务器再根据相应的业务流程组织，将消息发送给订阅该消息的应用系统，被请求系统进而做出回应。

该集成架构具有以下优势：

(1) 在中心辐射型集成架构中，消息系统具有高度可扩展性，容易与其他系统进行集成。

(2) 中心辐射型集成架构为建立动态、可靠和灵活的系统提供了基础。在中心辐射型集成架构中，消息的生产者和使用者之间是一种松散耦合的关系，这种关系基于对消息的异步处理。对于使用者来说，它并不在乎是谁产生了消息、产生者是否仍在网络上，以及消息是什么时候产生的。对于生产者来说，它也并不在乎谁将接收这个消息。生产者和使用者只需将接收消息的格式达成一致，就可以达到目的。这样，一个子系统的修改不会影响其他子系统的正常运行。

(3) 消息系统具有高度的可靠性，消息服务为消息传递提供保证，可以指定给消息不同的优先级别，从而保证关键任务消息比常规消息的传送有更高的吞吐量。

在选择集成服务器平台时，我们对 Microsoft 的 BizTalk Server、IBM 的 WBI、BEA 的 Weblogic Intergration 和 Oracle 的 HTB 进行比较，这些产品的设计思想基本相同，仅仅是实现方法有所区别。由于该大学的基础数据库建立在 Oracle 数据库上，因此，我们选用了 Oracle 的 HTB 作为集成服务器平台。

该项目完成至今已接近 1 年，从运行的效果来看，达到了项目的预期目标。项目验收时，得到了同行专家和该大学领导及有关人员的好评。

论虚拟现实技术的应用与发展

【摘要】

本文介绍了虚拟现实技术的主要技术和应用，并结合作者本人参加的国家重点实验室的自然科学基金项目，重点讨论了基于图像的建模与绘制在虚拟现实中的重要地位与应用，详细分析了近15年来国际上基于图像的建模与绘制技术的发展状况与当今世界上最先进的研究热点，并详细阐述了作者在国家重点实验室精心基于图像绘制技术科研工作中所参与的系统开发。文章最后，说明了作者所参加的科研项目的开发成果，并简单介绍了作者在试验项目中所发表的论文，对未来几年中虚拟现实技术的发展趋势作出了较为科学的预测并提出了作者本人独到的见解。

【正文】

虚拟现实技术是当前科学研究的热点，同时也是国家大力支持发展的核心技术之一。所谓虚拟现实，就是通过计算机等虚拟手段来构造一个非常接近真实世界的虚拟环境，它主要体现在视觉、触觉、嗅觉等方面的逼真模拟来达到高度的沉浸性，也就是如我们通常所说的“身临其境”。它涉及的领域非常广泛，包括仿真技术、多媒体技术、传感器技术以及计算机图形学等多个科学领域。同时，随着科技的发展特别是网络、计算机硬件等领域技术的完善，虚拟现实也得到了越来越多的应用。包括军事、医疗、航空航天、旅游业、建筑业、娱乐业等众多行业中，虚拟现实技术都具有广阔的应用前景。

我作为浙江大学CAD/CG国家重点实验室的一名科研人员，有幸参加了多个关于虚拟现实方面的科研项目，其中国家自然科学基金项目《虚拟现实中的基于图形的建模与绘制技术》中，我投入了大量的精力，进行了广泛的学习与探索，参读了国外很多相关领域的文献，特别是近15年来SIGGRAPH大会中的精采论文，并与实验室的其他人员进行了大量的讨论，和国内外领域中的专家学者进行了多方面的交流，取得了很多实质性的进展。同时，我结合该课题在理论的探索中进行了大量的编程模拟实验，开发了一些虚拟现实系统。

我们知道，虚拟现实的一个最主要的方面就是视觉效果上的真实性，例如虚拟漫游系统。可以说，视觉上的问题解决了才能进行考虑触觉、嗅觉等方面，否则就是本末倒置，因而目前世界上大部分的科研工作都是基于视觉上的真实性进行研究的。同时，随着网络技术的发展，网络的带宽不断增加，其于Web方式的虚拟现实系统越来越成为主流，而且在美国很多先进的企业中已经得到了广泛的应用。而网络上虚拟现实的一个关键性问题就是速度，因为网络带宽相对于本地运行来说速度还是很小的，那么如何解决虚拟现实中国形的绘制速度，以达到网络显示的实时性就是一个非常重要的课题。我们也了解到，传统的建模是基于几何的，这导致当场景较为复杂时计算量过大，特别是要处理物体表面材质时，几何贴图一来真实性不高，二来速度过慢，几乎没有达到网络实时性的可能。对于大部分PC机客户，更是不能在网欣赏到逼真的虚拟现实场景。因此，世界上一些科研人员就提出了使用基于图像的建模与绘制的方法。因为图像是真实世界的拷贝，逼真度高，而计算量又很小，可以满足普通PC机的用户以及当今网络带宽的需要，很好地达到虚拟现实场景的实时绘制。

基于图像的绘制技术(image-based rendering, IBR)最早由Apple Company的Eric Chen提出，他早在80年代初就在SIGGRAPH上发表论文介绍了他的图像拼接理论，次年又基于该理论开发出了Apple著名的Quick Time VR系统，并在当年的SIGGRAPH上发表论文详细地介绍了该系统，在当时的计算机领域引起了极大的轰动。该系统的主要手段是用多点拍摄的8张图片，进行无缝拼接算法整合成一张360°的环视图像。我在实验室进行相关的科研时，参与了一个类似的系统开发，当然，时隔多年，我们又在原来的Eric Chen的理论基础上，结合最新的科研成果，使用更好的算法开发我们的系统。关于该系统的开发方法，本文后面将有详述。在Eric Chen之后，又有很多世界顶级的科研人员不断在SIGGRAPH上发表论文，以向世人告知他们的最新研究成果，现在IBR已经成为每年SIGGRAPH中一个重要议题，每年都有4~6篇论文发表。例如基于特殊点之间进行转换的Morphing技术，基于全光场理论的光场建模绘制技术，基于图像拼图的拼接技术，基于人机交互的辅助技术等，都在一段时间内取得了巨大的成功。特别是拼图技术和全光场理论带来的各种基于光场的技术，在最近的3年内主导了国际上IBR领域甚至虚拟现实领域的主要发展方向。

我在实验室从事相关领域的科研工作之时，主要是进行了大量的理论学习与研究，当然也开发了一些虚拟现实系统。例如本文前面提到的类似于Quick Time VR的系统。在此系统的开发中，基础的理论已经比较成熟，而且已经有产品化的实物供我们参考。我们的主要工作就是结合理论，使用逆向工程的方法，导出软件系统的设计思路，并融合后来几年中国际上对Quick Time VR系统提出的改进论，开发出一套自主知识产权的虚拟现实系统。

我们首先以理论为基础，结合逆向工程工具导出原始的设计模型，加以改进，采用合理的分层结构重新设计了系统。在讨论完系统高层模型后，我们的人员需要分成3个组来进行开发。一组开发算法，一组开发系统界面，一组开发底层控制部分。我被分在系统界面开发组。虽然不是核心算法组，我们在对待界面交互设计时仍然进行了大量的分析与策划。我们抛弃了传统的MVC模型来工作，而是采用了国际上最先进的TaskMaster模型来作为我们的界面部分的驱动模型。TaskMaster模型是MVC模型的改进，它使用有限自动机来控制状态驱动，将原来的MVC模型作了更加彻底的分离，使得软件系统的各个模块之间独立性更强，增加了内聚，减少了耦合。它将系统界面对用户的响应分成多个Task，每一个Task使用一个有限自动机来驱动，有限自动机接收消息，执行Implementation，并完成状态转换。同时，TaskMaster模型不但结构清晰，还具有代码自动生成工具，使用该工具，可以快速自动地实现自动机模型。总之，使用TaskMaster模型，我们解决了传统MVC模型的不足，取代了MFC消息响应机制的不完善性，大大加速了系统的开发。后来的开发成果也证明了我们使用较少的人力物力开发出了优质、美观、高效、高可靠性及高移植性的系统(TaskMaster模型很容易移植到Linux系统下，并且Linux系统下也有TaskMaster自动代码生成工具。)

另外，在开发过程中，我们的测试文档也做得相当成功。坚持使用了单元测试、集成测试，确认测试每一个步骤都到位，特别是单元测试和集成测试，我们小组做了大量的工作。先是个人测试，然后进行组内交换测试，最后全组讨论，再进行集成测试。文档也和测试一样，是每一天都跟着来的。

在项目管理上，我采用了Microsoft Source Soft来进行控制。因为小组有6人，大家分别在不同的机器上开发，进度不同，加上我们小组内模块相关性高，传统分开开发最后集成的方法肯定不行，而使用了Source Soft之后，管理开发就轻松许多，并且大大减少了最后进行模块集成的工作量。

总之，在本次科研活动中，我们不但顺利、高效地完成了模拟系统的开发任务，而且大量实践了软件工程方法，引进了国际最优先的TaskMaster模型等，得到了很好的效果。更重要的是，我们出色地完成了实验室的国家自然科学基金项目，为实验室争得荣誉，为国家填补了科技空白。我个人而言，不但在组织本小组进行系统开发的过程中，实践了项目管理、进度管理、质量管理、人员分配、合作能力、文档管理以及测试管理等软件工程方法，并引进了国际先进的开发模式，取得了良好的实践效果；同时，本人还有一篇关于基于图像的绘制技术的论文被第八届联合国际计算机大会录取。

当然，我在科研工作中以及相关系统开发中也遇到一定的困难，比如模型选择、人员分配等都走了一些弯路。而且TaskMaster模型虽然清晰高效，但代码量比传统MVC要大得多，文件数也多，给管理造成了一定的困难。

再回到虚拟现实领域，我觉得目前在视觉方面已经较为成熟，例如IBR领域，未来的扩展需要依靠硬件辅助计算、网络分布式、人工智能技术的结合。而整个虚拟现实领域将更多地开始从事触觉和嗅觉等方面的研究，而且为了达到更真实的效果和更高的实时性，我预计将会有更多的整合人机界面学、人工智能、分布式计算、数据库等技术，共同来创造一个完美的虚拟现实世界。

论中间件在 SIM 卡应用开发中的作用

【摘要】

我曾于近期参与过一个基于SIM卡应用的开发项目，并在项目中担任系统分析的工作。

在分析过程中，我们依据面向对象方法对系统进行了划分，其中许多的模块已有成熟的中间件产品可供使用。我们选择中间件产品的标准是：功能、性能、封闭性、独立性、可扩充性、是否标准化等（或是否是主流产品）、跨平台性、话语言性，以及中间件的大小和价位，并且面向对象的优于面向功能的。

由于复用了大批的中间件，使得项目工作量大大减少，开发周期明显缩短，并且在项目的编码部分，我们也尽量制作自己的中间件，以便在以后复用，或以恰当的形式销售，增加公司的利润。

自己设计中间件，首先要做领域分析，以确定其价值，并依据上述原则开发。产品形成后，要由管理系统负责中间件产品的存储、描述、查询、发布以及动态维护、版本控制等。

【正文】

中间件技术是近几年随着软件工程技术的发展，网络应用的普及而迅速发展起来的，其在软件开发中最突出的贡献在于资源复用。随着分散的、作坊式的、独立的软件开发方式向集中地、大型的、协作式的软件开发方式的转换，软件的划分向着高内聚、低耦合的方向前进。类似的，重复性的、独立的软件部分被抽取出来，被做成模块、对象以及中间件。我们利用中间件拼装产品，共享着人力资源和时间资源，减少了重复劳动，减少浪费，增加了价值。由于中间件的出现，软件开发可以变得十分专业化。例如通信中间件、数据库中间件，……，他们屏蔽了许多繁琐的细节，使得应用程序开发人员质量与优化业务逻辑。各种不同的劳动量得以在自己熟悉的领域得到发展，不用分散精力，这就是分工带来的优势，专业化带来的好处。如今，在计算机软件科学中，这就是中间件带来的好处。

我曾经参与过的某个项目在这方面带给我深深地感触。这是一个关于SIM卡的应用，其功能要求是：在SIM卡上存储信息并与计算机内的信息同步；利用SIM卡的信息实现网络安全控制。

我作为系统分析人员，采用面向对象方法进行分析，这里只作简要概述。

在该项目中，我抽象出的一个核心对象是“信息对象”。通过读取SIM卡/计算机内信息，可构造出该对象实例，通过将该对象实例更新入计算机/SIM卡完成信息同步的功能。而实现网络安全控制的功能则通过验证对象属性来完成。

其中读出SIM卡、写入SIM卡的这部分，已经存在大量的中间件产品，我们选择了微软公司的系列产品，带给我们极大的便利。因为我们不再花费精力去考虑如何编写代码以及与SIM卡读写设备驱动程序进行通讯，如何将信号转换为二进制数，或转化为字符。它使我们摆脱了对设备物理细节的考虑。

此外，在读写计算机内数据时，利用了数据库中间件，此处选择的是微软的ODBC、ADO系列。在进行用户界面设计时，也应用了许许多多各具体特色的图形控件。

而在实现SIM卡网络安全控制时，选择了SUM公司的基于JAVA的SIM应用套件、SSL网络安全协议的辅助构造我们的项目。

在这样的选择之后，遗留给我们的仅是一些信息管理以及在各中间件进行协调、调度的工作，整个项目因此而显得异常轻松。

如果在没有中间件可供使用，一切需要从零开始的境遇中，这样一个小项目将会变得多么可怕！除了劳动力资源的严重消耗，还带来了许多隐患，例如与别人的产品的协作性问题。因为如果对硬件设备的处理不同，网络传输的各自构件的协议不同，根本不存在互通性。

因此中间件在软件模块标准化、软件复用性、跨异构系统协作方面也功不可没！

在我们选择中间件产品的时候，最先考虑的是功能性。因为只有项目缺少的功能，才有被使用的价值。而且中间件产品性能也是可以影响项目的因素。此外中间件是否采用标准和通用的接口形式，其封闭性、独立性、扩充性、跨平台跨语言的能力，也是影响我们项目的稳定性、可扩充性的关键因素。另外产品的大小、运行环境以及价位也是我们考虑的因素，这影响到项目的成本和效益。

如今的主流开发技术都支持面向对象和中间件（组件、控件、服务）。大家都意识到了复用的重要性。如果由我来决策，我会选择在开发项目同时开发中间件产品。虽然这样的开发模式在单个项目中所耗资源较大，因为要考虑中间件的各种特性，如独立性、通用性等。但就长远目光来看其复用带来的成本节约以及项目周期短、开发模式标准化、通用化等利益，将极大地提升企业利润。

为了能够开发出有价值的中间件并同时进行高效地利用，我们注意以下方面：

一、在决策开发中间件时，一定要做好领域分析。在相关领域内通用的中间件才有可复用性，才有价值。

二、对于中间件的策划，要注意其独立性，封闭性（接口清晰、标准），最好是以对象的形式开发，提供继承功能，可以提高中间件的可扩充性。

三、最好做到跨平台、跨语言，以增强通用性。

除了这些原则，对于中间件产品有也应有相应的管理工具，应建立系统对中间件产品进行存储以及描述、查询、定位该产品，对于产品的升级也应用cus等版本控制工具来管理。此外，发布产品也应遵循业界的标准。

当前的Web Service其实也是一种广义的中间件，公司应尽早投入到该领域，现有产品有微软的.NET，IBM的WebSphere等。

这些方案的益处是能够提高产品的可复用性，应用Web Service技术则提供了更远大的网络应用发展空间。

改进 Web 服务器性能的有关技术

【摘要】

一个大中型的图书馆信息系统涉及到许多方面的技术与方案，本文着重讨论与 Web 服务器性能有关的一些内容。

本人有幸作为项目负责人之一参与了某大型图书馆数字化信息系统的设计和基于 Web 应用软件的开发工作。由于在数字化图书馆信息系统中流通着的大多是数字化的索引、文摘、全文、图像或音频视频等多媒体信息，对 Web 服务器性能有着较高的要求。

结合实际工程经验，本文将从硬件实现手段（缓存服务器、均衡负载设备、Web 双机镜像、CPU 和网卡的提升、网络带宽扩充）和软件实现手段（三层 C/S 软件结构设计、应用程序部署）等两个大方面论述如何提高 Web 服务大路的性能，以便使用户能够更快捷、高效、安全地使用应用系统。

【正文】

随着 Intranet 信息技术的发展，图书馆为了更好地发挥其图书流通、资料检索和学术交流的职能，图书馆的数字信息化工程也势在必行。某图书馆为了尽快地步入世界先进图书馆的行列，已经启动了一部分的数字图书馆工程。

该数字图书馆工程主要包括对外信息 Web 发布系统，交互式检索网，后台馆藏信息管理系统、多媒体资料采集制作以及外 VOD 点播系统等。本人有幸作为项目负责人之一，参与了整个数字化信息系统的总体设计，并参与了基于 Web 的一些应用（如对外信息发布系统、图像/全文混合系统、VOD 点播系统的）开发。

某图书馆数字化信息系统从网络环境上讲，主要划分为多个网段：（一）Intranet 接入部分，采用 2M 的 DDN 专线；（二）公网段（非军事区），主要包括前台发布数据库服务器、Web 服务器、E-mail/FTP/DNS 服务器、检索服务器及 SAN 网络区域存储设备；（三）是内部局域网，包括内网 Web 服务器、后台馆藏数据库服务器、OA 服务器等。（四）是 VOD 点播专用网，包括音频视频点播服务器等。由于制定了严格的网络级和应用级访问权限，通过具有三层交换能力的高性能交换机和安全授权认证系统等，有效地控制了访问权限，确保了数据的安全性和完整性。考虑到经费和人员素质及今后的维护管理运营等方面，操作系统采用 WindowsNT 平台，服务器选用 DELL 高端的系列，数据库采用 IBM 的 DB2。主干网为千兆快速交换式以太网，局域网百兆到桌面，VOD 点播网十兆到桌面。

在该网络环境下应用主要分为三大部分：（一）对外 Web 发布系统、对外图书辅助检索系统；（二）后台馆藏信息管理系统和图像/全文混合检索系统；（三）VOD 点播系统。由于绝大部分应用采用 Browser/Server 方式结构，最终用户在本机只需安装 IE 或者 Netscape Web 浏览器，在后台数据库服务器的支持下通过网页方式请求和访问各类应用服务。另外，由于在图书馆信息系统中流通的多为索引、摘要、全文或音频视频等多媒体信息，对 Web 服务器性能与网络带宽等有更高的要求。

通过不断地试验和实践，我们发现从以下几个方面可以相对有效地提升 Web 服务器性能：

（1）缓存服务器和均衡负载设备使用可以缓解访问瓶颈，提高网络带宽、实现均衡负载。

缓存服务器也称为 cache 服务器，可以存储 cache 静态的内容如网页、多媒体点播资源和会议实况（已压缩的、有一定格式要求的）等。此外，目前美国 cashflow 缓存服务器，已经可以存储 cache 数据库、ASP 等动态内容。cache 服务器通常放到防火墙之外，外网 Web 服务器之前，因此 Internet 用户点击网页不再直接访问网站 Web 服务器，而是访问 cache 服务器。

由于 cache 服务器具有多个 CPU 和高速大容量 I/O 通道，独立的 OS，因此能大大缓解 Internet 访问瓶颈，而且也具有一定的抗黑客攻击的能力。

目前某图书馆采用这种方式，把大数据量的静态图片、点播资源、虚拟三维应用等都事先置放在 cache 服务器中，即使现今只有 2M internet 的接入带宽，以上应用的播放速度和效果仍能让用户满意。

另外一种方式采用均衡负载设备或 Web 双机镜像。这种方式通过负载均衡的方法达到 Web 访问性能最优。Web 双机镜像较早以前流行的方式，虽能使系统可靠性提升，但由于双机总是在互相询问对方状态，将会影响一定的访问性能。均衡负载设备是独立于 Web 服务器的硬件，它和 Web 服务器及网站中其他服务器接在同一交换机上，通过负载调度程序为各个服务器分配工作量，从而，能达到充分利用资源，提高访问性能的目的。只是由于某图书馆目前对外发布资源相对仍较少，只有用了三台 Web 服务器，因此目前的均衡负载设备作用还不显著。

（2）从 Web 服务器的配置来看，Web 服务器自身 CPU 个数及速度、网卡数量、Web 服务器与防火墙的位置关系等，都会影响到 Web 服务器的性能。

从 Web 服务器硬件本身来讲，CPU 个数的增加、网卡个数的增加、I/O 信道的扩展无疑可以直接地提高 Web 服务器性能。此外，由于千兆口的防火墙目前较少且费用较高，如果把 Web 服务器放置防火墙之后，一定会大大影响 Internet 访问性能。某图书馆采用 IDS（入侵侦测）+Web 服务器（服务器防火墙，较低端，不会影响流量）+应用服务器+数据库服务器（防火墙，高端），分层次的安全模式，既保证了系统的安全模式，既保证了系统的安全性，又提升了网络访问性能。

另外，某图书馆还采用了 SAN 网络区域存储来提高服务器访问速度。

（3）三层 C/S 软件结构设计和应用程序的适当部署也会提高 Web 服务器的性能。

将业务逻辑、通用访问接口与数据等相互分离、分别置放于 Web 服务器、应用服务、数据库服务器上，通过过程功能逻辑的合理部署，也能大大改进 Web 服务器性能。

一般的原则是，Web 服务器只需接受 Internet http 访问请求，使 Web 只有最少的任务，把实际处理交给各个应用服务器处理，然后返回结果给 Browser。某图书馆采用这种方式专门开发了搜索引擎应用服务器和混合检索应用服务器等，达到了良好的应用效果。

事实上，Web 服务器的性能提升还存在很多手段和方法，比如 CPU 与存储之间关系，Web 交换机等等，有待于我们进一步的实践、分析和讨论。

论改进 Web 服务器性能的有关技术

【摘要】

基于 Web 技术的数据库应用是当前应用的一个热点,在用户数目与通信负荷很大的场合,提高 Web 服务器性能是一个迫切的课题。本文从笔者参与某个银行系统项目开发的经历出发,阐述了提高 Web 服务器的性能应渗入到项目论证、选型、开发、运行和管理的各个环节,只有各个环节都能充分考虑到性能与质量的需要,系统的性能才是真正可保证的和可扩充的。

文章从系统的实际运行与相应的经验出发,阐述了性能改进方面的一些具体措施。

比如:在本文中讨论了 Web 服务器平台的选型考虑;Web 服务器的配置管理;应用系统本身的优化与预先设计系统时可扩性的性能保障等具体内容。

通过技术上的分析与改进,综合性地运用多类措施与手段,在实际系统中,Web 服务器运行的性能得到了一定程序的保证。

【正文】

我所在的单位是把目标定位于金融领域开发 IT 应用的一家信息技术公司。随着金融电子化建设和发展商业银行业务之间市场竞争的加剧,各主要商业银行不断通过信息技术提供新的金融产品,并且希望整合市场渠道。比如主要的商业银行不断推出形形色色的网上银行服务。在这种背景下,本人参与了开发新一代网上银行产品,涉及提供网上个人理财服务、网上外汇买卖服务、网上企业服务等具有市场竞争力的产品。作为项目开发的组织者之一和主要的技术骨干,在整个项目开发过程中始终要处于第一线,从而在改进 Web 服务器性能、提高整个网上银行系统性能方面收获良多,在本文中简要讨论如下,希望与读者们共享经验。在 Web 服务器配置与优化方面,我有如下几方面主要的体会:

第一方面是 Web 服务器选型考虑。

在 Web 服务器选型及网上银行搭建这初,我们就已充分考虑整个网上银行的性能及可扩展性问题,这一考虑为系统的稳定性及扩展能力方面打下了坚实的基础。

某银行原有的一些网上产品由于开发较早,故而采用的是老式的 HTTP Server + CGI 程序调用的方式。这时,每一客户请求需要对应于后端系统的系统进程来运行 CGI 程序来处理,系统的开销相当大,系统的扩展能力也很差,性能已不能满足业务处理的需要,故而在为此银行系统具体选型的时候,我们一开始就否决了这种方案。

通过市场上同类产品的比较选择,我们选择了国际商业机器有限公司 IBM 的 Web Sphere 产品系列作为该行网上银行系统的建立平台。作出这样选择是因为 Web Sphere 基于使 HTTP Server 和应用服务器相分离的整体架构,同时支持 JSP、Servlet 和企业级 JavaBean 等轻量级线程规范,所有的请求对应于应用服务器上的处理线程,系统的开销低,效率非常高,同时 Web Sphere 整个体系结构相当的灵活,为适应扩展需要可以作不同的横向和纵向扩展,从而可以满足各银行未来的扩展需要。

正是因为在一开始选型的时候我们就已考虑到未来的扩展需要,整个系统在接下来的几次性能改进方面,我们大体上都能相对顺利地达到了预期目标。

第二方面是 Web 服务器的性能配置。

在一开始系统上线的时候,由于系统的负荷不是很大,为了节省系统总拥有成本 TCO 投资,我们在一台较低配置的 IBM RS6000 上投产了该系统。整个系统的 HTTP 服务器、应用服务器、通信服务器等均位于该台机器上,由于初始投产时用户不多,所以系统的性能基本上能令人接受。

但随着业务的发展和用户访问量的增大,我们发现该服务器的响应变慢,系统的 CPU 利用率和内外存交换显著增大。经过跟踪,我们发现关键原因之一是系统的内存不足的缘故。由于网上银行服务器把大量用户的会话信息保存在内存中供给应用系统使用,当内存不足时,大量 Session 信息被迫交换至硬盘,大量 CPU 时间消耗在等候内外存的交换上,系统效率迅速下降。

鉴于这种情况,我们把该服务器的内存由 2GB 扩充为 4GB,同时相应调整用户会话信息的保存时间,这样整个系统的效率又回到较为理想的状况。

由于新应用的不断投产及数据库操作的日益增加,我们后来逐渐监控到系统的数据库处于繁忙状态,系统的错误日志也记录下了供应用服务器使用的数据库连接处出现资源不足的情况。在这种背景下,我们认为整个系统由于硬件配置所限,应该进行横向扩展,因此我们把数据库服务器分离出来,配置到另一较高性能的服务器上,相应定义的数据库资源也大幅增加,这样整个系统的性能又处于较为理想的状况。

第三方面是对应用系统进行相应的优化以提高性能。

Web 服务器配置及相应的硬件扩展不失为解决系统性能问题的一条捷径,但应用系统的优化也是应该重点加以考虑的,毕竟它能够在投入较少的情况下提高系统的运用效率。

在开发的初期,我们就已经十分注意系统的利用效率,比如提醒程序员尽量不要利用用户会话信息(Session)来传递大的对象,对于内存要注意回收等。同时,通过内部的交流会推广与介绍一些小的,有用的编程技巧来提高开发人员的水平,通过代码的抽查,希望能在早期就发现问题等。

在系统运行期间,我们通过监控发现,应用服务器所基于的 Java 虚拟机,其内存堆的空闲空间有不断下降的趋势,每隔若干天导致空间消耗殆尽,无法分配新对象空间,从而导致系统重启。在排除了系统本身问题的原因外,我们确定为应用系统的开发有问题。通过从网上下载 IBM 公司检测 Java 虚拟机的相关工具对 JVM 进行监控后终于发现系统内部存在着不能回收内存的对象,再通过查找相应的程序发现在该程序中有“环状”的对象引用,从而导致对象使用后不能被垃圾收集器所回收。这个问题的解决过程虽然十分艰苦,但由于该问题不能通过升级硬件或增加资源配置而得到根本解决,会给系统带来很大的隐患。所以,整个过程的分析与解决是完全值得的,更何况通过查找故障原因的过程,给整个项目组上了生动的一堂软件质量保证课,对项目组的质量意识起了很大的促进作用。

所以说改进 Web 服务器的性能并不单纯是系统管理方面的工作,它渗透到开发以及系统运行第一系列环节中。

第四方面预先考虑未来的扩展与性能需要。

随着系统的发展及成熟,考虑到用户访问量的不断上升,为了预留系统的发展空间,我们最近又对整个系统作了一个系统性的升级。通过引入多台 HTTP 服务器及应用服务器并行工作提高整个系统吞吐量及单点故障克服能力。由于在一开始选型的时候就已经充分考虑到动态负载均衡及横向扩展方面的需要,这一项的升级无需对整个系统的体系结构作根本的变革,对应用程序来说,更是没有造成任何影响。

整个项目历时近两年,从这两年的系统情况来看,整个系统是成功的。根据我亲身的经历,系统性能并不单纯是系统运行与管理阶段的问题,而是渗透在项目论证、开发以及运行的各个阶段。只有各个阶段都能充分考虑到性能方面的需要,在实际运行时,整个系统的性能才可能真正有保障。在技术方面来看,可以综合利用选型评估、硬件扩展、应用优化和系统配置优化等一系列的手段;比如在硬件扩展方面,又可以分为主要部件扩容,纵向升级、横向升级等方面。在我们的项目实践中,曾综合地利用了上述的各种手段。比如某银行的整个系统日访问量不足 1 万至现在的每日超过 10 万次以上的点击的发展情况来看,整个系统的性能保障及提高方案是比较成功的。

基于构件的软件开发

摘要:

2007 年上半年,我参加了**市信息办技术服务平台的项目开发,担任系统平台的设计和开发工作,该项目主要为了支持**市党政机关专网的二期建设所开发设计的,它是一个在线式的远程服务平台。在项目的开发过程中,我充分的进行基于构件的软件开发,考虑到软件复用和以后的扩展等方面,文中介绍了构件平台的选择,几种 COM 构件的开发,平台的实现过程。基于构件的开发大大提高了软件的质量,缩短了开发周期。该项目现在以完满完成,得到用户的一致好评。但现在看来,在开发过程中也出现了一些不足,文中就构件的划分和使用方面谈了自己的一些看法。

正文:

2007 年上半年,我有幸参加了**市信息办技术服务平台的项目开发,并在其中担任系统平台设计和部分开发工作,本项目隶属于**市党政机关专网(党政内网),是电子政务建设的一部分,市党政机关专网是政务信息化建设的基石,在市政务信息化建设和发展中具有举足轻重的地位,随着**市信息化建设的要求不断提高,要在下半年开展市党政机关专网的二期工程建设,二期工程覆盖的范围更广,为了配合二期工程的建设,对此项目有较高的要求。

由于此平台是一个综合性的在线式基于 WEB 的远程技术支持平台,存储着核心信息数据,提供:任务发布和查询,拓扑图查询,机关代码查询,软件支持下载,BBS,VOD 视频教程等服务,功能的多样化,必然会在软件开发中出现重复开发的现象,所以在开发初期我从该系统的需求分析入手,着重考虑了软件复用,在开发过程中采用了基于构建的软件开发方法。我将软件设计为 B/S 结构,三层体系架构:用户界面层,应用逻辑层,数据库层。用户界面层考虑到用户的易用等方面,用的是浏览器(如 IE 等),通过 ASP 语言来实现同应用逻辑层构件交互;应用逻辑层来处理各种应用功能的实现,负责事务处理,主要通过使用 COM 组件方式实现,以构建形式进行设计开发,考虑到以后的扩展和软件的复用性,也可大大的提高了开发效率;数据库层用 SQLSERVER 来实现。

结合我部门的实际情况,我部门现有的各级软件系统都是基于 WINDOWS 系列平台,且开发人员对 COM 组件技术也比较熟悉,对开发语言 VB6 也很熟悉,在这里我选择 Microsoft 应用服务器的解决方案,用 windows server2000 作为应用服务器(它集成了 2MTS,ASP,IIS,MSMQ,COM,DCOM),确定使用 Microsoft 的 COM 组件技术来开发该平台。用 SQL SERVER2000 为后台数据库,用 ASP+IIS5.0 来架构网站。

由于 COM 组件既可以被嵌入动态 WEB 页面,还可以在 LAN 或桌面环境的 VB,VC 等应用中使用。另外该组件之间是彼此独立的。当应用需求发生变更时,可能需要更换中间层的个别 COM 组件,但并不影响其他组件的继续使用。组件具有若干对外接口(属性和方法),可以根据不同的应用需求,由选择的使用不同接口。即使不再使用某些接口时,COM 组件本身仍然可以继续使用。同一 COM 组件可以在不同的应用环境中重复使用。

由于该系统以在线服务支持为主,主要包括了用户交互操作,资料输入,数值处理,数据存储等几个方面,我们依据平台的主要功能,为了节省开发时间和提高维护效率,我决定把公用的代码模块都作成了组件,例如把记录操作(如记录的删除,增加,修改等),数据库操作,查询做成用户管理组件,把用户身份证和用户类型识别作用户管理组件,把所有实现与数据库的连接作成连接组件,把用户的错误操作,与系统的交互出错等作成错误处理组件。对于各组件采用 VB6 语言进行编写,并写成 DLL 文件,同过注册成为 COM 程序,供各个组件调用。在数据库连接方面,我们采用了 ADO 技术。由于 ADO 采用了 OLE-DB 技术,使能访问各式各样的数据并提高了访问性能。

在该平台开发过程中,主要设计和实现了以下一些 COM 组件:

1 用户管理组件,包括身份认证功能

我们主要定制 COM 组件用户管理组件 USERCHECK.DLL 进行用户管理处理。该组件主要完成两个功能:一是身份认证功能,主要是提供用户登录时验明身份,保证应用的安全性。二是根据用户所输入的帐户名确定该用户的类别。

因此,该组件具有两个接口,每个接口代表组件的某个属性或方法。对用户的登录请求作出相应得处理:如果是普通用户登录则转入普通用户平台,如果是管理员登录就转入管理员平台。

2 查询和提交信息组件

我们主要制定 COM 组件 Querysys.dll 进行查询和提交信息处理。该组件主要完成两个功能:一是提供查询相关信息;二是提交相关信息。

该组件有两个接口,每个接口代表组件的某个属性或方法。如果用户的请求是查询信息功能,则将查询信息请求作出相应的处理,并将查询结果集返回给用户。如果用户的请求是提交注册信息,则将提交信息请求做出相应处理,并将信息提交替式返回给用户。

3 连接组件

我们主要定制 COM 组件 conector.Dll,该组件主要完成与数据库的连接。该组件有一个接口,那就是确定数据源,一边自动连接后台数据库。

4 错误处理组件

我们主要定制 COM 组件 Cerror.dll,该组件主要确定错误类集,该组件具有一个接口,主要是输出错误信息,方便用户排错。

我们把编译好的组件,将其在 MST 中注册,并将其分布在服务器上,这样就可以在设计平台过程中进行调用这些组件了。在本系统中,我们通过以下几种方式把组件集成到系统中来:

1 是连接集成,即我们将组件直接嵌入 ASP 主页中,即在 ASP 脚本中通过 set 对象名=server.createObject(“类名”)来引用,使此二进制组件可以运行于服务器端。

2 是容器集成,即如果一个组件需要调用另一个组件时,就在需调用的组件中引用另一个组件的方法。例如在使用查询和提交信息组件时就需要先调用连接组件。

我们结合连接集成和容器集成两种方式来组装系统,以登录界面为例,在客户端我们只提供两个输入项和一个提交信息的功能按钮,主要通过 ASP 来实现。在服务器端,主要根据用户输入的信息来进行相应得处理,这就要调用各种组件。如果用户用错误的帐号和用户名来登录,则系统调用用户管理组件,错误处理组件和连接组件,返回非法用户的信息。如果用户以合法的身份登录进入用户平台,这就要调用用户管理组件,连接组件。如果管理员以合法的身份登录进入管理员平台,也要调用用户管理组件,连接组件。

目前,该平台运行收到良好的效果,我们采用 COM 组件技术进行开发,减少了重复输入代码的工作,缩短软件的开发周期。同时,在尽行系统维护时,我们只关心组件的接口参数,而不用再考虑组件内部的具体实现,提高了系统的可维护性。在以后的工作中,如果我们要扩展某些功能时,也可以重复利用这些组件,提高了系统得可复用性。当然在开发过程中也总结了一些应该注意的问题,比如这次软件开发过程中,对构件开发过程的控制不够理想,究其原因是在划分构件时有所不当,划分时应该将版本变化相对同步且一致的部分划分到同一个构件中去,这样每变更一次,就可以从构件上确定不同阶段软件的改动状况,也就是所谓的版本,这样可以在不同的版本上明确软件的整个开发进度,有效的控制了软件开发的过程,有益于系统的管理。

论基于构件的软件开发

摘要

去年初，单位承担了新立的“测井资料处理与解释集成软件”项目，目的是集成目前国内零散的测井解释方法，我有幸参加该项目，并负责软件系统平台设计和部分开发工作，在项目的实施过程中，我充分进行基于构件的软件开发，复用成熟的商业构件和本单位的构件资源库，同时考虑了本项目开发资源的进一步复用，形成了绘制组件包，数据交换组件和数值计算组件包等。基于构件开发，大大提高了软件的质量，缩短软件的开发周期。开发的软件目前在石油测井几个油田现场使用，并得到用户的好评。本文就在本项目中如何进行基于构件开发进行描述，并在复用构件的使用和丰富方面谈一些自己看法。

正文

2004年3月，我有幸参加了单位软件开发项目——“测井资料处理与解释集成软件”，并负责系统平台设计和部分开发工作，本项目是由中国石油天然气集团公司科技部的测井软件类项目，在项目立项之前，国内测井行业为了提高石油解释的分析和决策能力，在解释方法方面广泛与各大高校进行合作，经过长时间的积累，形成的非常多而且实用的方法和处理模型，并开发了简单的使用系统，但这些应用非常零散，不统一，无法协调工作，处理效率低，多个解释系统之间没有必然联系，而且数据格式多种多样，不利于测井的精细解释和综合应用。本项目的目的是为了综合决策能力，提高石油解释的准确率，考察了解国内流行的测井资料处理解释方法，将成形和半成形的解释方法进行筛选和整理，集成到统一的系统平台上，增强一体化处理功能和商业化能力，并有偿反馈给石油测井现场用户使用。

在本项目立项前，单位承担了较多的软件开发项目，软件开发方法一直在改进更新。单位早期的软件以数值计算为主，主要采用面向过程开发方法，但随着软件的复杂度增加，可视化要求和开发效率，在面向过程方法不能满足要求的情况下，开发方法转为面向对象方法，实现界面编程，可视化操作交互，开发效率得到大幅度提升，但是这种方法也存在一些不足，无法有效实现资源复用，多个项目之间的资源复用仅限于部分开发过程和偶尔可以复用的模块类，新项目很少能借鉴老项目，即使存在明显类似的项目也无法有效复用，出现了多个项目组软件界面不统一，操作习惯不一致等情况，因为新项目只注重眼前效率和操作适应性，根本没有考虑以后的复用，导致重复开发，浪费人力物力。因此，在面向对象开发方法的基础上进行改进，采用基于构件的开发方法，在每一次项目中，尽可能地考虑以后可重用的部分，以构件形式进行设计开发，虽然在单部分的开发工作可能增加，但是多项目综合考虑，大大提升软件的开发效率。而且，本单位大部分项目为测井类软件项目，在软件的设计思路和最终成品上看，很多部分非常相似甚至雷同，几个连续项目存在联系，基于构件开发的软件方法在本单位更能体现作用。

在“测井资料处理与解释集成软件”项目的设计和实施过程中，我使用了基于构件开发的方法，充分利用公认的构件技术-COM/DCOM 和 ADO 等，使用已有的商业构件-Xtreme 和 Esstentails 等，复用单位的构件资源-数据转换组件，并开发了新的构件，如绘制组件包，数据交换组件和数值计算组件包等。

本软件系统以测井解释分析为主，主要包括图件绘制、用户交互操作、数值处理和数据存储等几个部分，我将软件设计为 C/S 四层体系架构。客户端应用程序采用 Microsoft Visual C++ 开发，本单位已经购买了 Xtreme 的使用权，Xtreme 是基于 VC 的 MFC 框架的界面构件，它是微软的合作伙伴，能经过简单的几步向导就能产生美观的软件界面，它生成的界面具有一致性，对于测井解释集成应用，统一的界面可以降低软件的学习难度，增强软件的易用性，使用 Xtreme，我非常方便地实现了界面设计和开发工作。

图件绘制基于 GDI 和 GDI+ 开发，每个绘制对象采用单独的一个类进行封装，绘制的载体对象派生于 MFC 的窗口类，其他对象派生于同一个基类，基类设计足够的消息响应函数，对象采用重载虚函数或者操作自动定向，实现鼠标响应，属性修改和递归型子对象管理。绘制载体类和基类封装在一起，形成绘制核心库，其他对象单独以 MFC 扩展动态库实现，一个或者几个对象设计成一个扩展对象库，如果要增加和删除某个对象，只需修改相应的扩展动态库。这样，只需要绘制核心库与必要的绘制对象扩展库，就可以提供相应的绘制功能，而且绘制的对象可以定制并进行二次开发。在某些绘制图件的设计中，考虑到算法的难度，直接利用较为成熟的绘制构件-Gigasoft Pro Essentials，如等值线图，区块三维分布图和多井连通图等图件的绘制算法，核心绘制部分直接由 Essentials 完成，用户交互响应由对象封装。

测井解释的核心部分是解释方法，每种测井项目有多种解释方法，而每种解释方法又有多个解释模块，因此解释方法的设计是相对复杂的。在考虑到测井解释的流程化处理和模块动态增减的要求，采用 COM 组件技术进行了模块设计。例如常规测井解释项目，包含 POR、CLASS、SAND 和 CRA 等多种解释方法，而这些解释方法都包含几个解释过程，如 POR 包含泥质含量计算，渗透率计算、孔隙度计算和含水饱和度计算，将上面的每个过程设计为功能完整封装的 COM 组件，当用 POR 解释方法进行资料处理时，依次调用处理过程完成解释。由于采用组件方法设计，方法模块可以动态添加，如果在 POR 的基础上需要添加含油饱和度计算模块，只需要设计相应模块，动态添加到 POR 的管理队列中。在解释方法分解设计的同时，将多种相同的数值计算方法进行提取抽象，形成满足自己需求的数值计算组件包，方便多种解释方法的复用，并为新的解释方法研究提供强大的计算能力。

在软件设计开发中，数据的存储最为繁琐，由于测井解释面对多种数据格式，如国内 Forward 软件使用 wis (Well Logging Information System 测井信息系统) 数据文件，国外 GeoFrame 使用的 Lis (Log Information Standard)、eXpress 使用的 XTF (eXchange Tape Format 可交换带格式) 格式和 DPP 使用的 Clis 格式等，这些测井软件使用相当广泛，如果本项目软件能进入市场，这些格式都需要兼容，但是将这些格式都作为应用对象很难实现或者不可能实现。在数据储存方面，我直接采用 Sql Server 数据库存储，存储结构设计采用国际标准结构，国际石油标准化组织 POSC 制定了一套 EPICENTRY 石油数据存储标准，这种结构被石油系统广泛采用，可以存储石油系统多行业数据，如测井、录井、钻井和开发类数据等，直接采用已有标准可以满足复杂的存储要求，避免重新设计格式的存储能力限制。数据访问采用 ADC 数据访问组件，实现数据通讯和交换。在数据库的数据录入方面，我充分利用了本单位其他项目开发的数据格式转换组件，用于支持多种测井数据的解码和编码，由于以往项目的数据操作是基于文件系统 (CIF 格式)，内部封装了固定 CIF 文件格式转换功能，我将组件进行了适当修改，将基于 CIF 文件格式的操作改为内存操作方法，实现向数据库录入。

在数据库和解释模块之间，我设计了数据交换组件，因为测井解释的特殊性，一次解释需要获得数据非常多，一次解释处理至少需要二三条曲线，测井采样间隔一般为 0.01 米左右，2000 米井段每条曲线传输数据量就有 100 多 K，如果需要处理成像数据，数据量更大，可能有几十兆，甚至上百兆，由解释模块或测井图件直接采用 ADO 进行数据库访问，每次解释会多次访问数据库，势必会降低软件的执行效率和处理的稳定与连续性。数据交换组件将一次解释数据全部读出，打成固定格式的数据包 GDS (Global Data Storage)，GDS 可以以文件形式存放和内存形式交换，应用模块只处理 GDS 数据包，完成各种应用后，按照同样的格式传回数据库，优化处理性能，而且基于 GDS 构件的应用还可以扩展开来，GDS 不一定只与我们设计的数据库进行交换，如果需要增加其他存储方式数据库，只要知道详细的存储结果，开发访问组件，GDS 就能与其相联进行信息处理，新的数据格式也是一样，只要提供了访问组接口，也能够进行测井解释。

开发过程中，我充分利用已有的构件，也充分地考虑了本项目开发资源的进一步复用，开发了绘制组件包、数据交换组件和数值计算组件包等构件，完善单位的可供复用的构件库，积累和固化知识财富。到目前为止，绘制组件包和数据交换组件已经在新项目测井数据采集平台使用，效果非常好。

基于构件的软件开发，可以缩短软件产品的开发周期，提高软件质量，但是如果构件选择不当或使用方法不当，也会带来一些麻烦，在构件的使用上，我也走了一些弯路，如在使用 Xtreme 上，由于购买了它的源代码，曾将某些绘制模块建立在其源代码级别的紧密耦合，当 Xtreme 提供的新的版本后，这些紧密耦合的模块无法适应新版本，如果想继续应用，必须再进行同样的开发工作，最后我放弃了某些紧密耦合的功能，将绘制交互与 Xtreme 完全隔离开，后续的版本升级再没有带来任何问题，类似的教训还有很多，总结分析后，我认为在基于构件的开发中主要注意一些问题：如项目开发不要与已有构件紧密耦合，防止构件升级带来的影响；构件的使用一定要进行严格的测试与审核；不要勉强应用某些构件，如使用巨大构件的其中某个微小部分，虽然功能很适合应用，但可能影响软件效率；新项目必须注意构件库的积累，只有这样才能充分体现基于构件开发的便利；新构件功能一定要完整，文档及接口定义明晰，否则难以复用。

在本项目的开发工作完成后，相比以往的开发项目，开发周期明显缩短，软件的质量明显提高，用户软件维护要求也减少了，而且其扩展性非常强大，目前该解释软件已经在华北油田、长庆油田和土哈地质研究院进行了现场应用，得到了用户的一致认可。通过这次案例，我认为在软件系统的复杂性不断增长的情况下，基于构件的开发能有效提高软件质量、积累和固化知识财富，并有效地缩短软件产品的开发周期，提高了软件生产效率。

论基于构件的软件开发

摘要:

基于构件的软件开发是提高软件生产效率和软件产品质量的有效途径,本文结合我们的实践,以“在线学习支持服务平台”项目为例,讨论了基于构件的软件开发的技术的应用。由于我校现有的各级软件系统都是基于微软 Windows 系列平台,因此我们确定使用微软的 COM 组件技术来开发该平台,并介绍了该平台所使用的几种 COM 组件,主要采用 VB6 语言编写通用模块并生成 DLL 文件及注册成为 COM 程序,客户端用 ASP 语言来实现并通过 ADO 技术来调用 SQL SERVER 2000 和 COM 组件。

在本项目中的开发过程中,我担任了系统设计工作。

正文:

“在线学习支持服务平台”是面向我校开放教育学生进行远程学习教学辅导,经过多年的远程教育模式的探索,确立了成熟完善的远程教育教学模式--利用先进的网络数字信息技术,为广大的学生提供开放的教育平台和最优秀的教育资源,突出个性、学生自主学习的教学。

“在线学习支持服务平台”是一个综合性的在线式基于 WEB 的远程教学平台,存储着核心信息数据,提供网上课程、信息发布、查询、BBS、VOD 视频点播等教学服务,该系统的开发技术主要集软件复用、企业级应用程序开发等技术于一体的“基于构件的软件开发”。系统运行于 WINDOWS SERVER 2000,用 SQL SERVER 2000 为后台数据库,用 ASP+IIS5.0 来架构网站。

由于 COM 组件既可以被嵌入动态 Web 页面,还可以在 LAN 或桌面环境的 VB、VC 等应用中使用。另外该组件之间是彼此独立的。当应用需求发生变更时,可能需要更换中间层的个别 COM 组件,但并不影响其他组件的继续使用。组件具有若干对外接口(属性和方法),可以根据不同的应用需求,有选择地使用不同的接口。即使不再使用某些接口时,COM 组件本身仍然可继续使用。同一 COM 组件可以在不同的应用环境中重复使用。因此,结合我校的实际情况,我校现有的各级软件系统都是基于微软 Windows 系列平台,且开发人员对 COM 组件技术也较熟悉,对开发语言 VB6 也很熟悉,因此我们确定使用微软的 COM 组件技术来开发该平台。

该平台采用 B/S 结构进行设计,把整个系统分为三个层:数据库层,应用逻辑层,用户界面层。用户界面是浏览器(如 IE 等),并通过 ASP 语言来实现同应用逻辑层构件交互。应用逻辑层负责事务处理,应用逻辑层用主要通过使用 COM 组件方式来实现,数据库层用 SQL Server。

我们依据平台的主要功能,在平台开发中,如果采用传统的方法来开发,则每实现一个功能都要编写同样的代码,为了节省开发时间和提高维护效率,我们把共用的代码模块都做成组件,例如我们把记录操作(如记录的删除、增加、修改等)、数据库操作、查询做成用户管理组件,把用户身份认证和用户类型识别做成用户管理组件,把所有实现与数据库的连接做成连接组件,把用户的错误操作、与系统的交互出错等做成错误处理组件。对于各组件我们采用 VB6 语言进行编写并生成 DLL 文件,通过注册成为 COM 程序,供各个组件调用。在数据库连接方面,我们采用了 ADO 技术。由于 ADO 采用了 OLE-DB 技术,使能访问各式各样的数据并提高了访问性能。

在该平台的开发过程中,主要设计和实现了以下一些 COM 组件:

(1) 用户管理组件,包括身份认证功能

我们主要定制 COM 组件用户管理组件 UserCheck.dll 进行用户管理处理。该组件主要完成两个功能:一是身份认证功能,主要是提供用户登录时验明身份,保证应用的安全性。二是根据用户所输入的账户名确定该用户的类别。

因此,该组件具有三个接口,每个接口代表组件的某个属性或方法。对用户的登录请求做出相应的处理:如果是学生登陆则转入学生学习平台,如果是教师登陆则转入教师平台,如果是管理员登陆则转入管理员平台。

(2) 查询和提交信息组件

我们主要定制 COM 组件 QuerySys.dll 进行查询和提交信息处理。该组件主要完成两个功能:一是供学生用于查询学生成绩和查询课程信息;三是提交学生注册信息。

该组件具有两个接口,每个接口代表组件的某个属性或方法。如果学生的请求是查询功能(QueryInfo 方法),则将查询信息请求做出相应的处理,并将查询结果集返回给出学生。如果学生的请求是提交注册信息(Submitinfo 方法),则将提交信息请求做出相应处理,并将信息提交提示返回给用户。

(3) 连接组件

我们主要定制 COM 组件 Conector.dll,该组件主要完成与数据库的连接。该组件具有一个接口,那就是确定数据源,以便自动连接后台数据库。

(4) 错误处理组件。

我们主要定制 COM 组件 Cerror.dll,该组件主要确定错误类集,该组件具有一个接口,主要是输出错误信息,方便用户排错。

我们把编译好的组件,将其在 MST 中注册,并将其分布在服务器上,这样就可以在设计平台过程中进行调用这些组件了。在本系统中,我们通过以下几种方式把组件集成到系统中来:

一是连接集成,即我们将组件直接嵌入 ASP 主页中,即在 ASP 脚本中通过 SET 对象名=Server.CreateObject("类名")来引用,使此二进制组件可以运行于服务器端。

二是容器集成,即如果一个组件需要调用另一个组件时,就在需调用的组件中引用另一个组件的方法。例如在使用查询和提交信息组件时就需要先调用连接组件。

我们结合连接集成和容器集成两种方式来组装系统,以登陆界面为例,在客户端我们只提供两个输入项和一个提交信息的功能按钮,主要通过 ASP 来实现。在服务器端,主要根据用户输入的信息来进行相应的处理,这就要调用各种组件。如果学生以错误的学号 and 用户名登陆,则系统调用用户管理组件、错误处理组件和连接组件,返回非法用户的信息。如果学生以合法的身份登陆进入学生平台,这就要调用用户管理组件、连接组件。如果教师以合法的身分登陆进入教师平台,也要调用用户管理组件、连接组件。

目前,该平台运行收到良好的效果,我们采用 COM 组件技术进行开发,减少了重复输入代码的工作,缩短软件的开发周期。同时,在进行系统维护时,我们只关心组件的接口参数,而不再考虑组件内部的具体实现,提高了系统的可维护性。在以后的工作中,如果我们要扩展某些功能时,也可以重复利用这些组件,提高了系统的可复用性。目前该平台运行存在的缺点是:由于在 ASP 中运行的 COM 组件是二进制代码,当 COM 组件工作出错时,ASP 不能指出 COM 组件发生错误的具体位置,只能简单显示对象创建不成功。这样就给我们在调试该平台过程中增加了难度。

论软件产品线技术

摘要

本人在测井行业的一个国有企业软件开发部工作，从 2002 年初开始，我陆续参加了多个测井软件开发项目，这些项目都是测井行业资料处理解释软件，具有很强的行业特征，其开发方向和应用范围都非常相似，从“测井资料处理集成软件”项目，开始我实施了软件产品线技术，虽然在开始阶段，由于经验不足和管理不善，遇到了一些问题，但是随着逐步实施，都得到的纠正和有效控制，目前这几个软件项目都非常顺利的完成，实施工期明显缩短，极大的提高了产品质量，本文就在这些项目中为什么实施软件产品线？在实施过程中遇到哪些问题？产品线开发支持工具选用情况和产品线实施带来的益处等进行论述，并分析总结在目前本单位产品线技术应用中存在的不足。

正文

本人在测井行业的一个国有企业软件开发部工作，在近几年来内，我陆续参加了几个的软件开发，2002 年初参加“测井资料处理集成软件”项目，此项目的目的是集成本单位重点实验室研究成形的测井资料处理方法，提供一体化的应用，服务于各个测井公司，我负责底层平台设计和开发，2003 年初参加“阵列感应实时分析软件”项目，主要针对单位自行研制测井仪器-阵列感应仪器 MIT 的现场资料实时处理，总体负责软件的设计和实现；2003 年中，参加国家级项目“山地地震和矢量地震勘探研究”的“成像勘探开发处理软件”子项目，本项目的目的是实现国内勘探解释方法和仪器的大整合，“成像勘探开发处理软件”子项目完成成像勘探部分的软件开发，我负责成像勘探部分软件架构设计和分析实现；2004 年初，参与总公司科技局“测井综合应用平台”项目，性质与“测井资料处理集成软件”类似，但集成范围扩大到整个测井行业，我参与了总体架构组，并完全负责底层平台的设计。

这几个项目都是测井行业资料处理解释软件，具有很强的行业特征，其开发方向和应用范围都非常相似。几个项目表现了很多共性特征，如数据录入接口方面，都采用测井行业标准的数据格式 CIF、LIS 和 XIF 等；数据表现方面，采用图件可视化方式；数据管理方面，除了第二个实时解释项目采用文件系统外，其他均采用数据库存储，使用测井国际化标准组织 POSC 的 EPICentry 规范，不仅如此，处理流程也大致相似，首先获取数据、进行数据转换，按照用户指定的方法进行资料处理，然后以形象化的方式反馈给用户一些处理特征，进行处理结果输出，各种资料归档等。除了这些共性外，几个项目也表现了一些个性特征，如“阵列感应实时分析软件”要求的实时性比较高，性能要求较高，采用文件系统，“测井资料处理集成软件”项目用于实验室方法集成，分析功能和研究性质略强于其他三个项目等，虽然存在个性，但是我认为几个项目更多地表现为一致性。

在这几个项目中我都实施了软件产品线技术。主要原因有以下几个：1、本部门长期承接测井行业软件开发，开发范围清晰，且容易界定，在多个连续的项目中，一直存在一些共性开发。在早期项目中，由于构架和开发接口不统一，造成重复开发的弊端，实施产品线可以有效节省人力、物力和财力。2、测井行业目前出现的软件都是层次架构，80%采用采用 C/S 模式，主要应用在测井资料处理上，其余采用 B/S 模式或者两种组合方式进行简单数据查询和处理结果显示，有利于产品线架构设计。3、由于行业特点，开发的软件很难适合所有的测井公司，而多数情况是量身定做，因此同一个项目可能出现多种软件版本，如华北油田专版，重点挂接生产测井解释功能，而长庆油田对斜井校正等方法较为重视，采用软件产品线可以方便解决这些问题。4、同时，测井行业中，方法的更新较快，但是一般的软件使用的方法都相同，如“测井资料处理集成软件”项目和“测井综合应用平台”项目使用同样的软件处理方法和图件绘制方式（测井行业特征，方法永远采用最新的），一旦解释方法和绘制方式改变，所有的软件要求能统一改变。

我在接手“测井资料处理集成软件”项目后，与其他开发人员向单位建议了软件产品线的开发技术，得到同意并开始实施。由于是新项目开始，而且软件规模不算很大，因此，我们直接采用了革命的方式，项目直接实施全新的软件产品线方式，我将团队分为两路，一路总结分析旧项目的软件，一路进行当前项目的分析界定，同时聘请行业专家和技术专家结合本项目从大局进行规划，首先进行行业分析，对行业分析结果与单位的规划进行整合，最后在形成的领域架构基础上进行当前系统的需求分析和设计界定，其中开发重点在核心资源库上，然后当前项目的软件直接复用资源库上的产品线构架和构件。这几个项目完成后，本单位的核心资源库得到了丰富和充实，由这几个项目直接构成的核心资源包括两套产品线架构，分别适用于测井资料处理和测井资料检索管理两种开发类型；数据格式转换构件组，能够进行十几种测井原始数据进行转换和录入；测井图件绘制构件组，数据按照四十多种图形方式显示在通用的桌面系统、浏览器，并能以位图、矢量元文件、SVG 和 CGM 进行输出；还有文件系统和数据库操作标准和一些商业构件库和中间件等。

几个项目完成的非常顺利，除了第一个项目的出现短暂延期外，其他项目大幅度缩短了开发周期，而且软件维护和版本升级非常方便，但在实施过程中我们也遇到了一些问题：1、在第一个项目实施期间，经验不足，人员分工与过程组织不合理，核心构件与应用系统开发混淆，导致了开发设计的反反复复，延迟工期。2、部分核心资源设计不合理，某些构件在老项目方便使用，但没有充分考虑新项目的特点，在新项目无法充分利用，如果直接修改冲突构件满足新项目，老项目的维护和更新又无法与新项目配套，在一段时间内存在同时维护多套多版本构件的情况。3、资源管理不利，因为产品线实施需要非常强大的构件管理和配置管理，我的四个项目针对不同油田的应用每个项目都生成了多个软件产品，维护还包括了核心资源维护和应用系统维护，实施前期经常出现管理不善带来的负面影响。

到目前，我们逐步解决了这些不利情况，主要是加强人才培养、资源管理和组织管理。在资源管理上，我们主要通过自动化工具来解决。目前支持产品线的工具不多，大多是构件的管理和组装工具，在系统分析设计和建模方面一直采用 Rational 的 ROSE，软件配置管理采用 CVS，核心资源管理方面从初始的利用 CVS 转变到北大青鸟的公共软件构件库管理系统，它是采用了 B/S 的多层体系结构，分布式的应用架构，基于 EJB 技术，具备较强的灵活性和扩充性，支持剖面分类等多种分类模式，并提供基于角色的用户管理机制，使系统具有灵活的权限分配和安全的控制方式。在选择这些工具的选择上最重要的是核心资源管理工具选择，因为其他工具相对成熟且熟悉，而核心资源库是产品线最重要的部分，并须能进行统一的管理和识别，如关键词分类、枚举分类和剖面分类方法等，因此我们选择了北大青鸟的公共软件构件库管理系统来进行核心资源库的维护，它不仅满足上述要求，而且完全支持 UDDI V2.0 标准规范，而且支持构件统一、描述、发现和集成。

产品线技术在本单位不断扩充与发展，到目前为止，形成了采集、通讯、解释处理等多方面的产品线架构和构件库，利用这些资源，可以方便快速地“组装”一个测井实用软件系统。通过实施软件产品线，本单位测井软件的开发模式逐渐规范，不同项目复用同一个或多个产品线架构，做到组件复用的最大化，实施工期明显缩短，极大的提高了产品开发过程和产品质量。虽然有了几年的技术和资源积累，我认为在本单位的产品线开发上还存在一些不足，如在领域工程方面，缺乏具备深厚行业经验的领域分析人员，需要在不断的实践培养过程中，辅以本行业外部专家的咨询和辅导；在核心资源方面，缺乏高效的管理，虽然使用了较为方便的构件管理工具和其他辅助工具，但是其智能化欠缺，不是很成熟，同时构件的组装上，还是采用人工代码方式，缺少智能组装工具应用；还有就是管理和过程方面，传统单兵作战思想影响依然深远，国有企业开发人员或多或少有一定技术封闭性，不利于核心资源积累，而且向团队开发和多部门协同开发转变存在一个阶段，需要更好的管理和更多时间的磨合。

论软件产品线技术

摘要:

本人作为某软件公司负责人之一，通过对位于几个省的国家甲级、乙级、丙级设计院的考查和了解，我决定采用软件产品线方式开发系列《设计院信息管理平台》产品。

该产品线开发主要有如下功能：1) 知识、资源管理平台；2) 内部管理平台；3) 基于 WEB 的办公平台。软件产品线最难开发的是核心资产的分析与建模，如何从用户重多不同需求中抽象出共性的东西，如何使得核心资产通过继承、参数化等方式能够组装成用户实际需要的产品，我以及我们公司的系统分析员做了大量工作。概括起来我们用了如下三种方式：

(1) 加强核心资产开发的灵活度，部分产品作成用户能够自定义的功能，彻底免除产品化时的问题，但这样做难度很大，实施周期长；

(2) 说服客户微调管理模式，以使得管理适应我们的产品线功能。这样做无疑是“双赢的”，但是实际不见得会行得通；

(3) 根据流程最长的需求开发所有需要的构件，构件间的接口做成松藕合。演化成产品时进行构件组装。

正文:

本人所在软件公司于 2003 年 6 月承接了某《化工乙级设计院管理平台》工作，主要功能包括：1) 知识、资源管理平台；2) 内部管理平台；3) 基于 WEB 的办公平台。该项目于 2004 年 2 月成功投入运行，得到客户的肯定。该项目的开发成功以及通过与该设计院员工的接触，对我触动很大。我作为公司主要负责人与其他负责人研究后，决定进入“设计院信息工程”这样的行业领域做深度软件产品开发。理由有如下几点：

1) 设计院是一类知识技术密集型单位，员工素质较高，管理比较规范，有完善的硬件设施——性能良好的局域网平台，具备了实施信息化管理的基础；

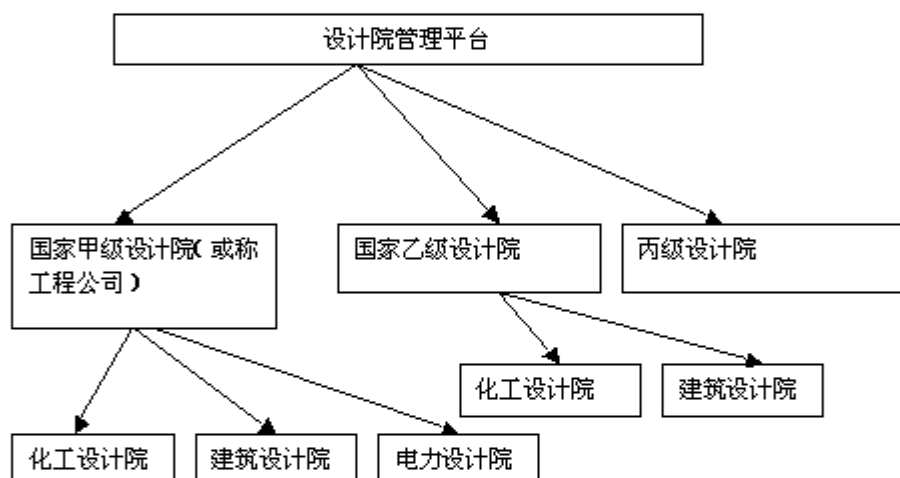
2) 设计院的图形工作站所用的二维、三维软件或其它工程计算软件都用正版软件且价格比较昂贵，高层领导他们有支付软件费用的观念；

3) 设计院开发一个项目，从可行性分析到项目实施、项目交付、到项目后期支持往往需要几年的时间，这期间产生巨量的、非常重要的各类文档（比如可行性研究报告、进度计划、项目条件、变更记录、会议纪要、计算书）和大量工程图，在整个项目开发期间这些资料要提供给项目开发人员查阅或使用、修改，这些资料的安全性、完整性都是至关重要的，这些资料既是知识、技术的载体、又是法律的载体；这些资料保存年限是 30 年或更长。目前设计院大都设立专门的部门，但基本都是采用手工管理。

4) 设计院的客户在项目开发期间以及项目上线以后一段时间都希望得到良好的技术服务。技术支持既包括现场技术服务，也包括远程服务。现在的远程技术支持是很被动的接听客户电话、传真或 EMAIL，没有建立专门的平台也没有进行规范管理。

有了对设计院内部运作的一般了解后，我们将已经开发成功的某《化工乙级设计院管理平台》精简做成演示原型，我带着这个原型以及对《设计院信息管理平台》的设想来到本市各类设计院：省级建筑设计院、市级建筑设计院、省级电力设计院、省级化工设计院、大型企业内部设计院等大大小小 20 家，征求他们的意见和建议，了解他们对该管理平台的认可度。我感到很欣慰的是 80% 的设计院高层领导对我们的产品设想有兴趣，并提出了针对自己单位的需要扩充或压缩的功能。随后的一个多月我带着补充修改后的原型产品，来到另外几个省会的设计院征求意向，结果：1) 大部分的设计院目前没有完善的管理平台；2) 如果我们真能设计出满足他们需要的产品，他们有兴趣购买。

通过持续 2 个月的调查我们公司于 2004 年 5 月决定投入这个行业的软件开发，因为要开发大量类同的软件产品，公司决定采用软件产品线开发。我们的对该系列产品的设想是：



这样划分的理由是：不同级别的设计院规模大小、管理模式、职能部门、待管理的资源区别非常大，国家甲级设计院除了设计以外，还能做技术咨询、工程总承包、建设监理、设备材料供货等工作，承接国际性工程。而丙级设计院一般隶属于相关厂矿企业，只能做本单位的项目开发，所以管理规模、管理资源相对较少，乙级设计院则介于两者之间。在同级别的设计院中，则因为设计院的性质不同，开发所采用的步骤和侧重点不同。该系列产品主要功能：

1) 知识、资源管理平台：管理内容包括工程图纸、各类文档（可行性研究报告、项目计划、条件变更单、项目会议纪要、来往传真等）、标准图库、标准及规范、招投标书、合同等；2) 内部管理平台：包括项目管理、人力资源管理、供应管理、客户关系管理、员工绩效管理；3) 基于 WEB 的办公平台：用于对客户的远程技术支持、保持与关键设备供应商、国家级研究机构的快速联系、快速获取新技术新行业信息、推广公司形象。

确立了产品线开发的基本思路以后，我采用一种动态的组织结构进行开发，初始阶段将公司主要技术力量集中起来进行核心资产开发；计划当我们的核心资产开发已经有几个实际产品之后，逐步将重点放到产品开发上并不断演化我们的核心资产。我决定核心资产的开发先重点考虑“乙级设计院”级别，这样做的好处：将来开发丙级设计院产品，只需做适当的功能裁剪；另外避免了开发甲级设计院管理平台的技术风险和项目风险。

建模我们采用 ROSE 2003，对需求分析的评估则采用 RequisitePro，软件开发平台选用 J2EE，数据库选用 ORACLE，这样做主要考虑 J2EE 的平台无关性。在开发平台的选择中我们曾经考虑选用 .NET，因为 .NET 开发相对简单和快捷，但 .NET 限制服务器在 WINDOWS 平台上运行，考虑我们今后客户中的“甲级设计院”，服务器一般都采用 UNIX 系统，为便于今后产品的移植，我们选择了 J2EE。版本控制工具选用 ClearCase。

核心资产开发最主要的是如何从众多不同需求中抽象出相同部分，并进行概括或分类。比如我们对员工绩效管理分析时就发现，相同的部分是：输入均为图纸量、其它文档、项目总利润、职工职称等，输出则为与项目相关的奖金。但是对奖金的算法，基本每个单位有自己的一套公式：有的对图纸量分专业计算复杂度系数；有的按本专业计算复杂度（比如关键技术、新技术或普通技术）；有的按复用度（该图纸与原有图纸的相似度）计算折扣；有的不分职称只计算单纯的图纸量；有的则考虑职称或工龄；有的将计算书折算为图纸进行考核；有的专门对计算书或其他文档考核、有的计算考虑加班有的不考虑。

面对我们从 20 来个设计院收集不同的考核方法，我以及我们的系统分析员确实伤透了脑筋。如果全部采用抽象类或接口来实现，那么抽象的层次肯定比较多，产品开发人员今后很难理解核心构件的意图，而通过组装类构造的话，组装类的开发工作量比较大。而最大的问题则是如何适应用户考核方法的变更。我们最终的做法是：收集和整理 20 来个设计院的考核方案中所有考虑因素的全集，对这些全集进行分类，然后使用 MVC 设计模式设计成用户自己可以控制的控件，在控件间设定一定的逻辑表达，由用户自定义计算公式计算奖金。这种设计虽然核心资产的开发周期相对较长，但我觉得是值得的。因为大大缩短了今后产品的开发周期，并减少了产品定制时的错误。

本次核心资产开发中公用部分较多的是对“知识、资源管理平台”的开发，虽然最初我们也了解到不同设计院管理方法也有所不同，我们在研究几家管理比较好的设计院方案后，拿出了一套比较规范的管理流程方案，这几个设计院表示愿意改进他们的管理措施以适应我们的方案。这一点是我本次产品线开发最值得骄傲的地方，实现了“双赢”——客户改进了他们的管理，我们节约了开发成本。

我觉得最为困难的是在众多不同需求中如何抽象出相同部分，而对不同部分进行灵活的组装或构造比如本次设计的核心模块：“项目管理”的流程：项目开发→可行性研究报告→可行性评审→初步设计→初步设计评审→施工图设计→施工图交底→项目后期支持；但有的项目由生产厂家进行可行性研究，那么同样省略的就有可行性研究评审；还有的小型项目，不需要做初步设计，项目开工后直接进入施工图设计。这种情况下我根据流程最长的需求开发所有需要的构件，构件间的接口做成松藕合。演化成产品时进行构件组装。

因为目前我们通过核心资产演化的产品还不多，所以很难评价核心资产开发的整体质量。这个问题的解决有赖于更多实际产品的开发以及更多的用户反馈。本次软件产品线开发，我觉得有如下一些收获：1) 对我们公司员工的素质是一个大锻炼，2) 在开发过程中我们也与设计院一些高层领导及部门领导结下了深厚的友谊，加深了对该类技术密集型单位的认识，非常便于我们今后推广最终产品；3) 通过核心资产的更多开发，我们今后产品开发的工程量、产品开发质量都会有数量级的改善。

论软件产品线技术

摘要

根据公司软件系统开发的需要，我们在软件的开发过程中引入了软件产品线技术，成立了基于软件产品线的项目组。本人有幸参加了该项目，并在其中担任软件分析与设计、软件产品线核心资源开发的工作。

在软件产品线的开发过程中，我们使用了 ROSE 建模工具，有效地完成了产品线中核心资源和产品的建模分析与设计实现；我们使用了国际标准 POSC 数据模型框架，有效地解决了数据的多样性与可扩展性，实现了统一开放的测井数据访问系统；建立了统一的可扩展的地质绘制组件和统一公用的数据处理模块。最终圆满的完成了公司产品线的建立和各子系统的开发。

正文

我公司是中国西部最大的石油测井服务公司，主要从事测井工程服务、测井仪器与测井软件的研发工作，在公司内部已经开发了多套测井数据软件，其中包括测井综合地面监控采集软件、测井数据解释处理系统、测井实时快速处理与解释系统、生产测井解释系统，分别由多个项目组分别开发。

其中，测井综合地面监控采集软件主要配合公司的综合地面系统，通过与地面硬件系统的前端计算机的通讯，实现对井下测井仪器状态的实时监视和控制、测井数据的实时采集和显示。整个软件分为通讯模块、数据采集模块、实时解码模块、实时简单处理模块、实时显示模块、实时出图模块和主控界面模块等。

测井数据解释处理系统主要完成测井数据的测后处理，包括测井数据的解码、数据编辑、环境校正、常规处理、成像处理、成果图绘制、解释报告生成等功能，整个软件按分层架构开发，包括系统底层（包括数据底层、绘图底层、通讯层）、业务逻辑层（包括各种测井数据处理模块）和应用软件层（包括数据解码程序、数据编辑程序、深度校正程序、综合绘图程序、解释报告生成程序等）。它主要是针对勘探开发测井中产生的新井的测井数据处理，所涉及到的数据主要是按深度采样的连续的测井曲线数据，而成果图主要是一种按深度比例的连续的测井图。

测井综合地面监控采集软件和测井数据解释处理系统均以推出了多个版本，进行了大范围的生产运用。而测井实时快速处理与解释系统和生产测井解释系统是我们正在准备开发的系统，测井实时快速处理与解释系统主要通过通讯卫星与在野外进行工作的测井车上的测井综合地面监控采集软件进行数据通讯，准实时的获得测井数据，然后由在解释中心的测井解释专家根据邻井资料和区块经验进行实时分析，整个平台分为实时数据通讯模块、实时解码模块、实时处理模块、实时深度校正模块、实时环境校正模块和实时绘图模块等，而生产测井解释系统主要完成石油开采后期动态监测时的测井数据的数据处理，测井数据主要是离散的测井曲线数据，在数据绘制上和處理上与测井数据解释处理系统类似。

在以上的各个项目中本人主要负责了测井数据解释处理系统的软件分析与设计、软件构建的工作。对于以上的各个系统，如果用抽象的视角去观察，我们可以发现，其实他们处理的对象是一样的都是测井数据，在功能上都是获得数据后，处理数据，然后按规定的绘图格式展现给测井工程师，所不同时各系统的侧重点不一样，测井综合地面监控采集软件主要是完成对测井数据的采集工作，和通过数据的曲线图形方式监视井下仪器的工作状态和测井数据的正确性，对系统的实时性要求比较高；测井数据解释处理系统主要完成测井数据的测后处理和测井数据的多角度观察分析，对实时性和可靠性要求不是很高；生产测井解释系统与测井数据解释处理系统基本层次结构相似，只是数据的格式和显示方式不一样；测井实时快速处理与解释系统对实时性要求稍高一些。通过以上对系统的分析可知，在以往的开发中我们各个项目组的人员实际在数据的存取、测井曲线的显示、测井数据的处理方面我们做了大量了重复性工作，各个项目组都形成了自己的一套软件架构，在资源上造成浪费、且代码质量不能够得到有效保证。因此，由于数据存储和处理的一致性和绘图方式的一致性，我们可以引入产品线技术，有效的提高公司的软件系统的复用性和可扩展性，同时能够有效提高公司软件的开发效率和质量。

我们给上级领导提出了合并测井综合地面监控采集软件项目组和测井数据解释处理系统软件项目组，成立公司的软件研发部，并将测井实时快速处理与解释系统和生产测井解释系统交由软件研发部来进行开发，在软件研发部，依托测井综合地面监控采集软件项目和测井数据解释处理系统软件项目的软件开发经验建立公司的软件产品线，这个建议得到了上级部门的一致肯定，最终成立了软件研发部，其下根据软件产品线的要求设立核心底层组、产品组。

核心资源的建立。核心资源就是在测井行业软件中能够共享的部分。数据存储与访问部分、通用测井数据处理方法模块部分和测井曲线绘图部分，在我公司已建立和即将建立各系统中都是高度共享的。由于我们的核心开发小组的成员就是原来各系统开发的核心成员，对原系统都有充分的理解，加上成立统一的核心小组后，更方便程序开发人员的之间的交流与沟通，我们很快就核心资源的需求达成了共识，对于需求的分析和核心资源的设计，由于核心资源涉及到的系统边界较多，特别是测井数据的存储与访问接口，因此我们使用了 Rose 建模工具，通过对原有系统的逆向工程和 UML 建模分析，理出了核心资源的接口设计和核心资源子系统的划分和子系统内各设计类，最终有利于我们建立合理的核心资源软件架构。

在数据存储与访问部分，由于涉及的测井数据格式较多，难以形成统一的数据存储格式和访问接口。通过采用国际石油软件组织的 POSC 行业标准有效的解决了这一问题，使用 POSC 的 Epicentre 数据模型实现测井数据的统一化存储，而 POSC 的 DEA 标准描述了数据的存取交换的方法，使用 DEA 使我们的系统具备了数据访问的标准接口。另外由于采用了国际通行的 POSC 标准，使我们的软件具备了开放性，能够支持满足 POSC-Epicentre 数据模型的所有数据格式和对外提供基于 POSC-DEA 的标准数据访问接口，使得我们能够方便实现与第三方软件的数据交换。

测井数据处理部分，根据处理的通用性，将处理包分为通用数字信号处理包和测井专用处理软件包，将数据滤波等通用数字信号处理方法放到一个包中，将测井行业专用的 POR 处理方法等放到一个包中，实现了方法模块的高效复用。同时使用简单工厂模式实现了处理模块的动态加载。另外在构建测井数据处理构件时，由于测后处理和测时处理模块的处理方式不一样我们很难进行统一接口的设计，测时处理主要是基于深度点的单点处理模式，而测后处理是基于深度段的处理方式，测后处理由于处理的复杂性可能会用到历史数据，在实现处理模块核心资源时，我们以单点处理模块为主，使用模块聚合方式实现了基于深度段的处理模块。而核心绘图资源的建立过程中也遇到了同样的问题，主要是实时系统和测后解释系统之间的绘图方式的不同，一个时基于深度点的行模式一个基于深度段的绘图模式，数据对象绘制图件较多，实时系统注重绘图的实时性，测后处理系统更关注绘图的效果。但是在图头的绘制上具有相同性对实时性要求不高，且图头的绘制要求基本一致，因此通过建立了一套图头绘制子系统和两个版本的绘图子系统，以应对不同的时间响应需求和绘图质量。

通过使用软件产品线技术，最终有效地提高了公司软件生产效率和质量、缩短了开发时间、降低了总体开发成本。但是也遇到了一些问题：1）核心人员原先都没有从事软件产品线的开发，往往只关注自己熟悉领域的软件设计，没有全局的观念，为此我们通过加强团队人员间的沟通，互相了解彼此的领域，使开发人员的全局观念加强，另外通过软件架构设计培训，提高软件人员的架构观念；2）没有合适的构件集成工具，处理流程的建立主要依靠代码来实现，因此在未来的开发迭代周期里有必要开发集成工具。

论企业信息化战略规划技术

摘要:

本文通过建立就业服务系统的信息规划,讨论了企业信息化战略规划技术。企业信息化战略规划在企业信息化建设中占有非常重要的地位,是企事业信息化建设成功与否的关键。我们采用的方法是,成立信息化建设领导小组,由主要领导担任组长。一、了解就业服务系统的战略目标,使信息化的战略目标和就业服务系统的战略一致。二、统一部署,进行资料收集,除考察现有软件系统外,还要考察手工操作,形成资料,制作功能表。三、根据收集的资料,我们采用自顶向下规划与局部设计相结合的方法制定出了信息系统的框架。四、加强培训,提高员工素质。

本人在此次战略规划中担任技术层面的工作。

正文:

企业信息化战略规划,在企业信息化建设中占有非常重要的地位,是企业信息化建设成功与否的关键。

本人有幸参加了市就业服务系统的信息化建设规划,担任系统分析和需求分析等相关工作,主要负责技术层面的问题。

随着社会的发展,加强信息化建设成为各单位的首要任务。为了建设“数字城市”,加强就业培训的管理工作,构建信息化的培训管理格局,我们开始着手进行全市的就业培训信息化建设,根据省里的要求,做到同人同城同库,实现就业培训的“实名制”。

目前,就业服务系统的信息化建设缓慢,各单位各自为政,分别开发了不同的信息管理系统,这些系统的开发,没有统一的标准,系统之间互相不“说话”,无法实现数据共享,浪费了巨大的人力物力。为了从根本上改变目前的混乱状况,在上级领导的支持下,在2003年我们开发了就业服务系统的管理软件,实现了资源共享。

就业服务系统管理软件涉及到就业服务系统几个相关单位,包括培训、考核、职业介绍、就业、财政等单位。就业服务系统管理着全市失业人员的所有信息,要求系统具有从失业人员登记,发布培训信息,用工信息,到学员参加考核,顺利就业为止等一系列功能。

为了使就业服务系统的信息化建设达到预期的目的,我们进行了战略规划,成立了就业服务系统信息化建设领导小组,由主管就业服务系统业务的领导担任组长,主抓此项工作。具体实施的步骤如下:

一、我们根据全系统发展计划的文件,全面了解本系统的任务,目标和长期发展的战略规划。我们和上层领导座谈,认真了解全系统的任务和目标,及时,方便,准确,快捷为下岗失业人员服务,是我们工作的宗旨。认真阅读相关资料,准确理解长期发展的战略规划。

根据需求做好设计,形成报告,请有关的业务专家来进行讨论,并参考就业服务系统的发展计划,检讨我们制定的信息化战略规划是否符合战略发展计划。

二、由信息化小组招集各单位各部门的负责人开会,统一布置了全面进行信息化建设的事宜,要求各部门配合信息化小组的工作,把全系统的信息化建设工作做好。

随后我们紧锣密鼓的展开了调研活动。资料收集是非常重要的,没有详细真实的第一手资料,要想进行战略规划就是纸上谈兵,建设空中楼阁。

首先,我们考察现有的软件系统。

由于本系统有独立功能的业务部门,他们已有现成的信息系统。为了了解各业务部门现有软件,我们制定了各业务部门软件系统调查报告。该报告主要要求各部门填写现行系统的现状,主要内容包括:软件运行的操作系统,使用的数据库管理系统,主要的数据库和表以及表结构;现行系统运行所占用的空间大小,主要表的大小,软件的业务流程,基本框架结构等内容。

其次,对各单位的手工业务活动,进行调查。

我们主要采用了“请进来,走出去”的方法进行工作。

我们根据各单位提交的业务报告,请相关的业务骨干,进行讲解。并对不理解的地方进行提问,从而把业务流程“吃透吃准”。为进一步形成功能表打下坚实的基础。

我们到各单位进行走访,现场观摩和实习操作,对各业务部门的流程有了更进一步的理解。在进行走访活动之前,我们详细研究了各单位的资料,对其中描述的不清楚或者不明确的地方,做了重点标注,并记录下来,以备走访时使用。除了观摩和操作外,对业务人员和技术人员,按照事先准备好的问题进行采访。

第三,形成资料,制作功能表。

根据各单位提交的报告和我们走访的结果,对形成的文档资料,我们进行了整理。为了清楚的表达各业务段的流程,我们使用 Visio 制作了各种业务流程图,并用 Excel 制作了所需的所有表格,对表格的名称进行了统一,把各单位五花八门的表名进行了修改。例如:有单位把一月一报的表格叫月报,其中的内容为每日的内容,而对每月的累计报表,也叫月表;我们把表名进行了修改,规定:含有每日数据的报表叫日报表,含有每月数据的表格叫月报表。如果有多个月表,则能合并的就合并,不能合并的,则叫月表一,月表二,或根据报表的主要内容进行相应的命名。表格名称统一以后,便于交流,省去了许多解释的时间。

我们根据各部门的业务活动制定出了各部门功能表。

文档制作完成之后,我们进行审查,这次是由我们来讲,各业务单位听,对我们理解不对的地方进行修正。如此反复几次后,可以基本确保我们获得资料的正确性。

三、根据采集到的资料,我们采用自顶向下规划与局部设计相结合的方法制定出了信息系统的框架。将整个就业服务系统的业务分为相对独立的几个部分,失业人员管理系统,培训管理系统,鉴定考核管理系统,就业管理系统等。对于一些具体的业务采用局部设计的方法进行,如:报名,分配培训机构等。

我们设计了人员档案主题数据库,主要用于管理失业人员的各种情况,如:就业,培训,考核,技能等级,家庭住址,所在街道等信息。此数据库由各管理系统进行维护,保证数据库信息的及时、准确。

四、加强培训,提高人员素质。

在我们的调查过程中,发现各单位使用计算机软件系统的人员素质较低,对信息化几乎没有什么概念。开发出来的信息系统,需要使用的是“人”,没有人员素质的提高,盲目的进行信息化建设,也是很难成功的。

为此,我们组织了多期电脑扫盲班,提高班,加强了员工对信息化建设的认识,以及进行信息化建设所能带来的各种优点。通过培训,提高了员工的操作水平,为我们与各单位业务人员进行沟通起到了良好的作用。

总之,就业服务系统的信息化建设起步晚,底子薄,经过信息化建设规划后,为开展信息化建设指明了方向,为就业服务系统的信息化建设起到了良好的作用。

目前,本系统各单位已经使用了新的软件系统,结束了各单位各自为政的历史。但这仅仅是完成了信息化战略的第一步,为实现数据仓库,建立决策支持系统提供了基础。

在现阶段,本单位的信息化也仅仅是在业务方面进行的。其它方面的信息化建设依然开展不够。例如内部办公系统仅有和上级部门相联的一个办公系统,内部办公效率不高,手工操作还是比较多。对已有系统的利用率还不是很,领导决策也没有完全借助计算机的帮助。计算机的使用大部分还停留在打字机的水平。要解决这些问题,我认为主要是领导对信息化建设要有足够的重视(特别是一把手),不能仅仅停留在口头上,要有实际的措施。比如领导要参与培训和计算机的使用,并增加一些物质鼓励。在已有信息系统的基础上,经过若干年的数据积累,构建数据仓库,决策支持系统。建立内部办公系统,邮件系统,使各部门员工自觉自愿的使用计算机来进行工作,尝到信息化建设带来的好处。

论行业应用软件系统的开发规划

摘要:

本文通过地震基本参数快速获取与发布系统论述了行业应用软件系统的开发规划。本人去年有幸参加了《地震基本参数快速获取与发布系统的研制》项目的开发,该系统按功能被分为三个子系统:地震实时波形数据监控子系统;地震事件处理子系统;数据发布子系统。该系统具有地震行业的专业背景,需要地震监测中心(用户单位)、地震监测专家和项目开发人员间良好的分工协作;与常规的通用软件开发过程相比,具有明显的差异,主要表现在系统的分析、设计、实现、测试和维护均带有明显的行业特点。行业应用软件的成功开发必须遵循行业规范,正确使用专业术语,及时与专家和用户进行交流。

本人在项目中主要负责需求分析和软件系统设计,在实践中我发现,行业应用软件系统的成功开发是从其规划到诞生起就应从行业中汲取各种营养,才能长出行业特征,具有行业味道,满足行业应用的需求。

正文:

本人去年参加了《地震基本参数快速获取与发布系统的研制》项目的开发。在项目中担任组长,负责项目的规划、分析、设计。由于防震减灾是公共安全的重要内容之一,该项目主要针对大地震的震前预警和震后消息的快速发布,是防震减灾工作的关键业务系统,是地震应急工作的支撑技术系统之一,对公共安全具有重要意义。项目范围包括:地震数据采集、传输、存储、处理、发布;该系统是一个硬、软件相结合的信息系统;硬件包括:拾震器(地震波形采集器)、数据传输设备(基带 modem 和多路复用器)、多用户串口卡、服务器、手机。软件可以按功能分为三个子系统:地震实时波形数据监控子系统;地震事件处理子系统;数据发布子系统。

由于地震行业的专业性强,由开发人员现学领域知识是不现实的,这就需要用户和行业专家的协助。

聘请行业专家做项目组的顾问是应用软件开发过程中较好的策略。行业专家主要是作为领域知识源,提供技术文件和已有应用。在地震系统的这个项目中,我们聘请了地震监测系统的资深专家作为项目的顾问,获取到《地震数据分类和共享指南》等技术文件和部分以前编写的源代码,该源代码是多个独立运行的、凌乱的、较小的系统,经过整理后可以用到新系统中,这为领域模型和设计模型的建立打下了良好的基础。

用户是应用软件目前和未来的需求提供者,其对项目成败的重用性是显而易见的。OOA 过程是从对系统将被使用的方式的理解开始的。在需求诱导时需要从用户的观点对系统建模,需要项目组和用户共同确定使用场景和定义系统的功能和运行需求;需要用户提供清楚而无二义性的终端用户和系统如何相互交互的描述;需要用户提供确认测试的基础。在地震项目中,我们与用户充分合作,请用户给我们讲解地震常识和本项目涉及的专用仪器的功能,将用户作为项目组成员,作为每个阶段的技术评审的主要成员。

开发者是应用软件系统的建设者,需要从用户方进行需求诱导,建立领域分析模型;需要将分析模型变换为设计模型,建立软件的构造蓝图;需要完成编码和测试。由于应用软件具有行业背景,在建立分析模型是需要从各种渠道了解行业知识;在建立设计模型是需要使用逆向工程技术从现有的系统中得到关键的算法。在进行测试时需要使用用户提供的具有行业特点的真实测试数据。开发人员需要随时在涉及到行业知识时向用户或行业专家请教,必须克服在开发一般应用软件时的问题讨论在相对封闭的范围内进行的特点。

与常规的通用软件开发进行比较,专业背景的应用软件开发过程及各个环节的主要差异有:

(1) 在计划阶段,具有行业背景的应用软件系统与普通应用软件的目标与动机一般是不同的,普通应用软件以给本公司或部门带来经济上的收益为目的,而行业应用软件以更好的完成某项政府职能为目的。地震行业的应用系统开发目的是为了有效的减少地震灾害造成的损失。系统方案的制定必须符合行业的特征。地震具有突发性、不可抗拒性、影响面广、广大群众积极参与、数据量非常大、算法复杂等特征,在系统方案制定时必须针对这些特点进行。在地震行业的应用系统中,从拾震器(地震波形采集器)、数据传输设备(基带 modem 和多路复用器)、多用户串口卡、服务器、手机到通信信道均实现冗余备份,当任一环节出现故障时,均能由热备件自动接管任务,应用软件必须支持这种功能。这就增加了应用软件开发的复杂性。

(2) 在需求分析阶段,行业背景的应用软件系统与普通应用软件相比一般需求比较明确、相对固定和有章可寻。必须获得行业的各种技术规范、数据共享资料及现行系统的运行数据。在此基础上划出应用软件的作用范围和与周围环境进行数据交换的接口,在类的划分时需要从行业术语中选择类的名称,类的属性需要在专业知识中有明确的定义,类的方法也需要使用行业内的主流算法;在地震行业的应用系统开发中,分析模型的建立需要从行业中得到基本素材,从行业知识和管理中得到优化,产生出行业专家认可的类-责任-协作者模型和对象-关系模型。

(3) 在设计阶段,行业背景的应用软件系统与普通应用软件相比,从用户界面设计、数据管理、任务管理到控制机制均要表现出行业特征。我们在开发地震应用软件时,邀请有专业知识的用户参加到项目组实施的关键环节中,将设计模型在地震行业进行各种地震知识的解释。确保设计模型与领域模型保持一致。在描述每个子系统并将其分配到处理器或任务时,子系统设计需要由行业中完成特定的功能的“层次”来确定;地震实时波形数据监控子系统需要完成各监控点的数据采集、地震事件识别和生成地震事件数据地震;地震事件处理子系统需要完成大数据量的传递、进行各种复杂计算和生产地震的各类基本参数;数据发布子系统需要完成在有新震情时第一时间以 web 或手机短信的方式对相关用户群进行数据广播;各子系统间通过分布式数据库完成数据复制和保证数据一致性。数据和任务管理的设计策略需要参考与《地震数据分类和共享指南》相类似的行业标准;用户界面设计也要根据地震数据的特点和物理意义来展示数据;为系统设计具有行业特点的控制机制,在地震应用软件中,我们采用地震触发作为系统的控制机制;通过创建每个操作的过程表示和类属性的数据结构而完成对象设计,数据结构和算法必须符合行业需要;使用对象间的协作和对象关系完成消息设计。在评审设计模型时需要行业专家和用户的参与,从行业的角度提出修改意见。

(4) 在集成测试阶段,行业背景的应用软件系统与普通应用软件相比,一般更强调进行 β 测试和进行新旧系统对比,需要使用行业的数据作为测试数据,测试案例需要由用户和专家参与制订,行业数据需要由用户和专家提供,测试结果需要由用户和专家确认和解释。由于地震数据具有突发性,当地震发生时系统的数据量非常大,因此系统测试需要在真实的数据环境中进行。我们在测试地震应用软件时使用的是真实的地震数据,从数据的采集、传输、存储、处理和显示等的各个环节均使用的是地震事件数据,根据测试结果和用户意见进行修改。

从上面的几个主要差别可以看出,具有行业背景的应用软件的开发需要行业中的各类标准、专业知识、现有相关系统和人员参与到开发过程中,要使行业特点与信息技术有机结合,在项目管理上还需要包含行业内的相关的人和物。

在地震应用软件的开发过程中,人员、设备、资料等的方方面面的资源均是由项目组与地震系统共享的,在知识和思想上充分融合,经过双方的共同努力不仅开发出了满足地震行业需要的应用软件,还丰富了双方的知识面,拓宽了眼界。

本人在项目中主要负责需求分析和软件系统设计,在实践中我发现,行业应用软件系统的成功开发是从其规划到运行的全过程中均应从行业中汲取各种营养,才能长出行业特征,具有行业味道,满足行业应用的需求。

论软件的可维护性设计

摘要:

随着软件大型化、复杂化的发展,软件维护所耗费的资源越来越多,软件可维护性设计日益得到重视。我单位近几年开发综合业务 ATM 交换机,用户对交换机的可维护性要求很高。我参加了该项目并负责软件的维护性设计工作。根据当前工作中在维护性设计中的不足。通过在各个软件开发阶段注重软件可维护性的应用,规范文档,使用 CASE 工具管理软件版本和成立软件可维护性设计小组等方面,为软件的可维护性设计提供了帮助,并最终开发出具有良好可维护性的交换机软件。但是由于初次实施这方面的工作,大家思想上认识不够,许多操作不习惯,并且单位里不具备专用的测试软件和其它 CASE 工具,在一定程度上制约了软件可维护性的实施。

正文:

经过一系列的需求分析、设计、编码和测试之后,软件正式交付用户使用。至此,软件变进入维护期。软件维护的工作量特别大,随着时间的推移,软件维护对开发商带来的成本压力也越来越大。许多软件开发商要把 70%的工作量用在维护已有的软件上,平均来说,大型软件的维护成本是开发成本的 4 倍左右。因此,在开发软件时,就应该考虑到可维护性问题,进行软件的可维护性设计。

2002 年底,我单位开始为某集团开发综合业务 ATM 交换机。该交换机支持多种业务应用,包括话音、IP 数据、图像和视频等;用户可通过维护台或网管对交换机进行配置和管理;由于特殊的应用,用户对该交换机提出了很高的要求,并且提出要求产品交付使用之后,我单位要有很好的服务支持,鉴于将来要大批量生产交换机,软件的可维护性设计被提上日程。我有幸参加了该项目,并负责软件的维护性设计工作。

在以前的课题中,也曾提到过要进行软件的可维护性设计,但在真正实施过程中,还存在诸多问题,主要表现在:

(1) 在软件开发过程中对可维护性设计考虑欠缺,导致软件设计从根本上就忽视了可维护性的重要性。

(2) 软件设计文档不规范,内容不一致。在维护阶段出现问题,根据文档不易得到有帮助的信息,难以定位错误的类型和根源。

在本交换机软件的设计过程中,我们通过注重软件可维护性的开发过程,规范文档,使用 CASE 工具管理软件版本和成立软件可维护性设计小组等方面进行软件的可维护性设计,最终开发出具有良好可维护性的交换机软件。

一、注重可维护性的开发过程

在整个交换机软件的开发过程中,从软件易于理解、易于测试、易于修改的角度出发,提高软件的可维护性。

在需求分析阶段,和用户进行充分的交流和协商,对将来要改进的和可能要修改的部分进行明确。由于该交换机所涉及的业务种类广泛,并且综合了话音、IP 和网管等多种技术,任何一种技术实现的功能不完善或者扩展性不好,都不会让用户满意。但是,另一方面,又考虑到用户需求和功能需求并非容易获取,所以通过和用户定期交流,举办各种形式的讨论等方式尽可能了解当前的需求和以后需要扩展的需求信息,由专人整理记录这些信息,作为以后的跟踪内容。即使在其它设计阶段对需求的临时变动,也要在这个记录中体现。

在设计阶段,交换机软件被划分为不同模块进行设计,并遵循“高内聚、低耦合”的设计原则;这些工作便于将来软件维护工作的进行。同时也已考虑到,对可能要扩展的地方,预留出充足接口。在一些模块中,如网管模块中,根据功能,尽可能使用面向对象的设计方法,以便维护时的修改和升级。

在编码阶段,我和小组成员制定了统一的编码规范,经过半天的培训,强化编码人员对注释的使用,并强调要保证注释的质量,对有可能出现误解的地方,注释的要详细。并且,每个文件都要注明编写者,生产日期和版本号。

在测试阶段,测试组成员已经负责进行测试,我们小组这时的工作是根据测试报告,对照测试大纲和用例设计,对当前的测试进行总结,比如,何种测试用例发现何种错误,最常见的错误,如何从测试结果判断是哪种错误,该错误所在的模块是什么。在相关人员修改错误时,记录排错时的思路 and 过程。特别是,根据这些总结,我们编写了“ATM 交换机软件故障解析”,这篇总结在后来的维护阶段被证明是最受欢迎的文档之一。

在维护阶段,制定严格的管理要求。每一次维护工作之后,都要按照配置关联,同步更新维护有关的系统文档和用户文档,包括维护需求、源代码、注释、设计文档、测试文档和用户使用手册等,保证系统的一致性。维护中所进行的修改要专人记录,生成“ATM 交换机维护更新”文档,做为内部文件存档。同时把一些内容扩充到“ATM 交换机软件故障解析”中。在用户使用时,做好用户的培训工作,初期由专人和用户一起操作交换机,直到能熟练操作,以免用户使用交换机时产生不满。

二、规范文档

交换机交付用户使用之后,除了在培训时所了解的内容之外,为了让用户对交换机软件能够更好的理解和使用,向用户提供了多种随机文档,包括功能说明,安装文档,用户使用手册,参考手册,管理员指南等。在文档的编写过程之前,我们编写了“ATM 交换机软件文档编写规范”,对文档格式和一些必要内容进行了规范,保证各文档的风格一致,内容一致。对于一些用户使用中容易出错的地方,比如配置某种功能时,在用户使用手册示例说明。在具体编写文档时,根据设计人员的反馈信息,也及时调整了文档编写规范。

在设计开发过程中,对某个问题进行修改,或者功能增删,要充分考虑到问题所涉及的不同文档,保证前后文档在该问题的一致性。对于所修改的部分,要填写“更改单”,需要写明更改人,更改理由,更改所影响的程序和文档,更改日期,批准人。采用 CASE 工具在这一方面也起到了事半功倍的作用。

三、使用 CASE 工具管理软件版本

在软件的设计编码过程中,尤其是在调试阶段,会不断的生成新的程序版本。为了有效的管理版本问题,采用 Ration 公司的 ClearCase 工具,由专人负责进行管理,从而保证软件版本的一致性。

四、成立软件可维护性设计小组

为了有效的对软件可维护性设计进行管理,成立了软件可维护性设计小组。我担任小组组长,明确了维护性设计的工作内容和各人的责任,针对不同的模块,又确定四个责任人。在运作过程中,组长对软件开发阶段所需进行的工作进行协调,各负责人对维护性设计所涉及到的变动控制进行维护。

因为交换机软件的各个模块开发时间有穿插,因此,对开发过程中出现的一些问题,包括技术方面和管理方面的问题,我们都及时进行了记录,对后面开发的软件模块进行指导,避免了同样问题的再次发生。现在这份文档已经成为单位新课题启动时的“必修”文档。

按照上面的思路,经过两年多的工作,我们已完成了交换机软件的开发,新的软件运行良好,交付用户后,用户很满意,受到了业务部门和技术部门高层的赞许。尤其是我们所总结的“ATM 交换机软件故障解析”和“ATM 交换机软件文档编写规范”等文档,对单位其它课题也起到了很好的指导和规范作用。并且,在提高软件可维护性的同时,也提高了软件产品的质量,我自己的开发管理水平也得到了很大的提升。单位的高层领导对我们制定的规范和做法也表示认可,正式在其它课题中推广。

然而,由于初次在整个软件开发过程中进行可维护性设计,还有许多要改进的地方。许多情况下,现有的可维护性设计措施理论性太强,具体实施时可参考的依据少,比如我们测试小组如何更好的与各课题开发组间进行协调工作,感觉还是有不尽人意的地方。在测试阶段,单位条件所限,没有采用专业的测试软件进行测试,主要靠人工根据经验生成测试用例,测试力度不够,隐含的问题较多,这也不利于今后的维护工作。

软件可维护性设计是一个长期的工作过程,我希望今后能够不断的充实自己在这方面的知识,提供整体能力,为单位软件可维护性设计方面提供新的、好的思路。

论软件设计模式的应用

摘要:

本人 2004 年有幸参加了中国石油集团的高性能数控测井系统项目的开发研制工作。该系统是在当前测井成套测井装备的基础上,为了满足高精度,高性能,高效率的要求开发的测井系统。该系统由井下成套仪器,测井遥测系统,测井地面系统,测井软件系统,测井解释评价系统等子系统组成。本人在其中主要是负责测井软件系统的分析、设计以及部分开发任务。设计模式是前人设计面向对象软件的经验总结和,在软件设计中灵活的使用设计模式可以极大的提高系统的稳定性,可扩展性,以及良好的可维护性。本文描述了在测井软件系统开发过程中,如何分析和发现相关模式,以及如何选择和应用设计模式,特别是介绍了 MVC 模式在软件框架和相关系统模块中的应用和使用效果。在文章的最后,讨论了在实际项目开发中,设计模式应用的有关想法和教训。

正文:

随着当前石油测井技术的发展,为了能更快,更好的得到储层地层信息,解决目前国内测井系统不统一,测井精度不高,效率低下的缺点,2004 年 1 月中国石油集团公司科技局成立了高性能数控测井系统项目,目的是为国内测井行业提供一个从井下到地面以及解释评价的整套测井系统。系统的设计目标是一次测井,取得所有合格资料,并且能保证 60 井次的免维修率。整个系统由井下成套仪器,测井遥测系统,测井地面系统,测井软件系统,测井解释评价系统等子系统组成。我主要是负责测井软件系统的分析,设计和部分开发工作。整个测井软件系统完成三个主要任务:测井数据的采集、测井数据的工程值计算、测井过程的监控。测井数据采集主要是采集井下仪器通过测井遥测系统传输的测井数据,并保证数据的完整性,正确性。测井数据工程值计算主要是把采集的数据根据不同仪器刻度计算方法进行工程值的计算。测井过程监控主要是把计算的测井数据用曲线和图像的方式实时的显示在屏幕和打印成图,由测井操作员进行实时监控。

设计模式是前人设计面向对象软件的经验总结和,在软件设计中灵活的使用设计模式可以极大的提高系统的稳定性,可扩展性,以及良好的可维护性。在测井软件系统框架进行分析和设计时,考虑如何提高系统的稳定性、可扩展性和可维护性时,我们采用了 MVC 设计模式。

MVC 模式构架包括三个部分:模型(Model)、视图(View)、控制(Control)。模型主要是对系统的数据和逻辑运算的封装。它独立与系统的界面和 I/O。视图把表示模型的数据和逻辑关系用特定的形式展示给用户。控制处理用户和软件之间的交互操作,当模型的数据有所变化时,控制负责通知视图做出相应的更新。模型、视图、控制的相互分离有利于模块之间内聚性的提高,耦合更加松散。一个模型可以对应多个视图,由控制来传播模型的变化从而更新视图。

MVC 模式如何在测井软件系统实现,我们主要是从如下四个方面进行:

一、分析系统功能,分离功能模型

首先根据系统的主要任务进行系统的模块分解。根据测井软件系统数据采集、数据转换和测井监控三个主要任务,把系统分为三个模块对应于 MVC 模式的三个部分。其中模型(Model)对应于数据的采集和工程值的计算。测井视图(View)对应于测井监控功能。测井模型所要实现的功能包括:测井数据的采集、数据的刻度计算、数据的存储、数据的操作。测井数据的采集负责硬件平台的初始化,下井仪器的初始化,井下仪器数据的中断相应,数据帧的采集,数据帧的重组等。数据的刻度计算主要是根据不同的仪器实现数据的刻度计算,包括刻度系数表的获取、刻度计算、深度延迟计算等。数据存储主要是原始数据的存储和测井数据的存储。这里我们采用的是测井公用的 XTF 格式做为数据存储格式。数据的操作是视图和模型之间数据交互的接口。它主要是提供数据输入和输出功能。

二、视图的设计与实现

视图主要是提供测井数据的图形显示。通过调用模型中的数据操作方法,提取测井数据,根据不同的测井数据提供曲线、波列、图像等多种表现形式。在本系统的实现中,为了提高数据采集的稳定性和程序的健壮性,采用进程间通讯的方式。就是说视图的实现本身一个独立的程序。它与模型之间的通过 TCP/IP 网络进行通讯。视图主要包括数据源、数据表象对象、绘图打印模块等部分组成。数据源负责得到模型(Model)的数据,然后把数据分配给每个数据表象对象。数据表象对象是个有层次的类家族,其基类是绘图类(CDrawObj),所有的数据表象包括道(CDrawTrack)、曲线(CDrawCurve)、波列(CDrawWave)、图像(CDrawImage),数值对象(CDrawData)等都是从其派生的。最后有绘图打印模块提供管理,负责视图的区域更新,数据表象的绘制和打印等功能。

三、控制的设计与实现

控制主要功能是提供用户的输入输出反馈,同时监控模型的数据变化,通知视图进行更新。由于控制和视图的耦合非常的紧密,在架构实现中,控制和视图是在一个应用程序中实现的。控制主要分为井下仪器控制和视图控制两个部分。其中井下仪器控制主要是由操作人员根据视图中的曲线和图像信息,对仪器发出的状态控制命令,以保证测井过程中数据和仪器的安全。视图控制则是操作人员对视图显示参数的调整,包括鼠标的响应和键盘的响应以及用户对测井原始图的特殊要求如道大小,曲线位置的摆放,颜色的调整等。

四、使用可动态添加算法模型

由于每次测井作业中下井仪器串的仪器种类和仪器的数量都是变化的,为了能更好的抽象出实际的测井模型,提高系统的灵活性,在模型中数据刻度计算部分,我们采用的动态添加的方式。我们把不同测井仪器的刻度算法封装到动态连接库,然后根据测井作业的不同,调用用不同的仪器动态库中的刻度算法。由于视图和控制与模型之间的松耦合,当用户添加算法模块,视图与控制基本不要修改。

在采用 MVC 模式的软件框架后,整个系统分为两个部分,数据采集管理器和数据实时浏览器。数据采集管理器对应于模型(Model)的实现,数据实时浏览器对应于视图(View)和控制(Control)的实现。我们采用的是 Visual C++.net 基于 Window2000 平台来进行系统开发。采用 MVC 模式给我们带来了如下好处:

1、由于模型(Model)与视图(View)和控制(Control)之间的松耦合,使得我们非常容易就实现了一个模型运行同时建立多个视图。这在调试仪器时非常有用,当硬件人员调试仪器时直接连接网线就可以一边看仪器一边看数据。不再需要象以前必须到地面系统控制室查看数据了。

2、适合多硬件平台的跨接。由于不同的硬件平台上采集数据的方式都不同,有的系统采用的是 PCI 总线,有的是 USB 接入,有的是 ISA 卡接入。由于模型(Model)和视图(View)的松耦合,当要移植到不同的硬件平台上是我们只有修改相应的模型(Model),有可以实现对不同硬件平台的支持。

3、良好的可维护性和扩展性。由于采用 MVC 模式,系统模块功能划分明确,代码实现也相对容易。代码的错误不会在系统中扩散,同时由于可以动态添加仪器算法模块,当用户添加新仪器时,不需要更改系统程序,只有添加仪器动态库 DLL 就可以了。

在整个系统的开发中,我们还应用了一些别的模式,有些模式是在进行系统设计时,就考虑到而特意实现的,有些模式是在采用别的方法实现后,效果不太理想,在代码重构时引进的。在应用设计模式进行系统设计和开发后,整个系统各个模块之间逻辑变的相对独立,耦合也很松散,结构的扩展性良好。而且使得代码的重用的程度变好,减少了错误的发生和错误在代码中的扩散。但是在实际应用模式的过程中,我还发现模式应用的经验越丰富,模式应用的就越好。有时在采用何种模式时,有几种模式方案可以采用,但是具体采用那个模式就需要不断的尝试,看看模式是否满足实际的需要。特别要注意的是不能为了设计模式进行设计,也就是过分设计的问题。这样会导致设计过于复杂,偏离程序设计简约够用的基本原则。

目前设计模式在软件开发中的应用正引起广大开发人员的注意,各大软件开发商也在软件开发工具中提供了有关设计模式的自动应用的工具,相信设计模式会越来越多应用于软件的设计和开发中。

论软件的性能优化设计

摘要:

本人 2004 年有幸参加了中国石油集团的高性能数控测井系统项目的开发研制工作。该系统是在当前测井成套测井装备的基础上, 为了满足高精度, 高性能, 高效率的要求开发的测井系统。该系统由井下成套仪器, 测井遥测系统, 测井地面系统, 测井软件系统, 测井解释评价系统等子系统组成。本人在其中主要是负责测井软件系统的分析、设计以及部分开发任务。作为整个系统控制核心的测井软件如何才能保证有整个系统的高性能和高可靠性呢? 本文从系统优化、程序设计优化两个方面来详细讨论如何提高整个测井软件系统的性能。其中系统优化主要是通过调节软件运行环境来优化软件性能, 程序设计优化主要从程序架构设计、语法、内存管理、输入输出等方面来讨论如何采取措施提高软件的性能。

正文:

随着当前石油测井技术的发展, 为了能更快, 更好的得到储层地层信息, 解决目前国内测井系统不统一, 测井精度不高, 效率低下的缺点, 2004 年 1 月中国石油集团公司科技局成立了高性能数控测井系统项目, 目的是为国内测井行业提供一个从井下到地面以及解释评价的整套测井系统。系统的设计目标是一次测井, 取得所有合格资料, 并且能保证 60 井次的免维修率。整个系统由井下成套仪器, 测井遥测系统, 测井地面系统, 测井软件系统, 测井解释评价系统等子系统组成。我主要是负责测井软件系统的分析, 设计和部分开发工作。

整个测井软件系统完成三个主要任务: 测井数据的采集、测井数据的工程值计算、测井过程的监控。对应于这三个任务, 整个系统分为三个系统层: 数据驱动层(简称前端), 数据处理层(主控)和表象管理层(后端)。数据驱动层的主要目的是封装地面系统和井下遥测系统, 为数据的上传和命令下达提供统一的接口。数据处理层的主要目的是按每种仪器的刻度算法计算测井原始数据。表象管理层则负责测井数据的表象处理, 主要是曲线和图像的显示与绘图。系统前端运行在 Vxworks 系统上, 主控程序和后端显示程序运行在 Window2000 系统中。前端主要是采用 Tornado2.0 开发, 主控程序和后端显示程序采用 VisualC++2003 开发。

由于测井软件系统负责着整个井下仪器的数据采集、仪器状态控制和测井资料获取。所以对测井软件系统的性能要求是非常高的。这主要表现在以下方面:

采集数据的完整性和正确性要求。由于目前井下仪器主要是通过 100K 的 CTS 遥测传输数据。要求每 80 毫秒采集一个完整的数据帧, 由于仪器算法的特殊性, 要求不能丢掉一帧, 也不能采集错误的数据。这就要求数据驱动层有非常好的实时性和高可靠性。

数据显示和打印的实时性要求。由于井下地质情况复杂, 为了保证测井安全以及测井数据质量, 需要把测井的数据用曲线和图像的形式实时的显示和打印出来, 以便于监控测井过程中井下的各种情况。

在软件分析设计和开发中, 我们主要是系统优化和程序设计优化两个方面采取措施来提高软件的性能并取得了比较好的效果。下面我主要从这两方面进行讨论:

一、系统优化

系统优化主要是从测井软件运行的系统环境角度方面采取措施来提高软件的性能。由于 Window2000 操作系统的分时多任务系统, 不能保证在 80ms 时间间隔内测井系统所要求的数据的完整性和正确性, 所以我们采用 VxWorks 实时操作系统做为前端操作系统, 运行数据采集程序, 保证测井数据完整和正确。再通过网络连接到主控程序处理数据。为了保证系统网络数据传输的性能, 采用千兆网, 连接主控和前端。同时为了提高数据 IO 的响应速度, 以及数据的可靠, 采用 SCSI 接口的磁盘阵列, 提高数据的冗余度来保证系统数据的可靠性。同时, 主机系统采用 Intel 公司 P4EE 结构的多线程处理器, 以及 512M 的 DDR 内存, 来满足数据吞吐的要求。

二、程序设计优化

程序设计优化主要是程序设计优化主要从程序架构设计, 语法算法和编译, 内存管理, 输入输出, 软件测试等方面采取措施提高软件的性能。

1、程序架构设计

由于优化性能并不是个局部的过程, 整个程序的性能提高必须在程序设计的开始阶段就给予考虑和规划, 以设计良好的软件架构保证系统的良好性能。本系统采用三层系统结构, 各层之间采用网络连接, 实现松散的耦合。即使后端显示程序出现错误, 也可以保证测井数据的完整和正确。同时引入组件对象的开发思想, 把每个仪器的算法及控制都做成组件, 并进行单独的测试, 从而保证组件的正确性。采用这种办法不仅可以保证错误不会在系统的扩散, 方便系统的调试。同时也可以利用以前成熟的仪器算法和控制代码, 不会引入新的风险, 提高系统的成熟度和可靠性。

2、语法算法优化和编译优化

在代码开发阶段, 可以通过 C++语法特点来优化系统, 实现更好的性能。例如程序中大量的简单函数, 为了提高运行效率, 一般都采用内联函数的方式实现。在循环语句中, 对各种情况进行度量主要通过把临时变量提出到循环体外, 以及设置 break 和 continue 语言减少循环次数。在对大内存对象的参数传递上, 例如仪器服务表结构, 基本都是利用引用和指针实现, 减少系统在堆栈上的内存分配。在仪器的刻度计算中经常要用到各种各样的算法, 采用在保证测井精度的原则下, 根据测井原理优化算法, 尽量把一些递归和非线性的算法, 转换为非递归的简单运算。同时为了充分发挥 Intel CPU 的功能, 我们选择 Intel 的 C++编译器, 做为系统的编译器。Intel 编译器对 C++错误检查比较严格, 这也保证了程序运行时的错误减少。在开发库的选择上, 我们也采用了精练的 ATL 库开发仪器组件, 使用 WTL 库开发用户的界面, 同时使用 STL 库进行数据结构的实现。WTL 库的使用使得我们可以抛弃传统的 MFC 动态库, 减少程序的内存占用, 也可以减少 MFC 常见的内存泄漏。STL 是经过考验的 C++标准库, 实现的代码精练高效, 管理方便。经过我们测试, 同样的数据结构如队列、链表等, STL 的实现比 MFC 的实现, STL 的速度大约快 10%。

3、内存管理的优化

内存管理对系统也很重要, 在测井后端显示程序中, 由于要分配大量的小对象, 例如曲线和矩形等, 通过分析性能测试数据, 发现大量的小内存释放和分配, 成为后端显示程序的速度瓶颈, 所以我们采用自己的内存管理机制来减少系统的调用。首先根据测试数据度量, 分配一大块内存给应用程序, 然后通过管理这块内存, 来满足程序对小内存块的请求和释放。如果内存不足的话, 可以另外分配大块内存进行扩充。通过采用自己的内存管理机制后, 再运行测试程序, 发现程序的运行速度几乎提高了两倍。

4、输入输出优化

输入输出在系统的应用非常的多, 特别是本系统采用三层网络的结构, 其中的网络输入输出更加的重要。为了系统取的最好的性能, 我们对前端和主控之间, 主控和后端显示之间的网络通讯采用的都是多线程异步通讯的方式。与以前采用同步的方式对比, 系统的速度提高了一倍, 而且网络的故障也减少了很多, 基本没有发生网络通讯中断, 和数据掉帧的现象。在本系统中还有一个比较关键的技术难点, 就是在如何在显示测井曲线和图像时同步连续的打印出显示的曲线。由于程序的实时性要求, 程序在网络通讯和数据计算上已经消耗了大部分的资源。在这方面我们主要采取的技术手段的同步缓存显示数据, 然后在安排一个单独的线程专门负责测井曲线的打印。经过测试, 打印和显示之间只有一米左右的延迟。在 200:1 的通常测井显示比例下, 这点延迟是在允许范围内的。

5、软件测试

测试并不能直接提高软件的性能, 但是测试是提高软件性能的有效手段。一个好的测试工具更能提高工作的效率和质量。在本系统的开发过程中, 我们主要是使用 Rational 公司的 Purityplus 工具进行测试。特别是用其中的 Quantify 工具进行系统性能测试, 可以精确到代码行。通过它可以快速发现系统的性能瓶颈在哪儿, 在哪儿耗时特别多, 以及整个程序的运行时间。大部份的优化措施都是在测试过程中发现问题, 然后在进行具体的优化。

在采用以上各种优化措施后, 软件系统的整体运行效率提高了 50%左右, 今年上半年, 整个系统完成了系统联调。在下井实验过程中, 在连接井下仪器、地面系统、和测井软件系统的基础上, 软件运行可靠, 保证了 24 小时不断电的情况下, 数据帧掉帧率为 0, 误码率为 0 的好效果, 得到用户的认可。

在整个系统的开发过程中, 特别是对软件的性能优化, 使我认识到软件开发中采用不同的设计和措施对软件的性能影响是很大的。好的测试工具对性能的提升有很大的帮助。在目前对软件性能要求越来越严格的要求下, 只有在重视测试方法和测试工具使用的情况下, 采用恰当的优化方法才能提高系统的性能。

论软件产品线技术

摘要:

根据“十五”国防科技重点实验室——“机载××PD火控雷达性能开发与评估实验室”的建设需求。我所在的中国×集团公司×所电子对抗研究部组织了用于该实验室目标产生、信号干扰、欺骗等的“射频半实物仿真目标系统”的设计开发。该系统采用分布式联网试验,主要任务是试验机载雷达的各项技术指标,模拟较逼真的雷达信号环境和其他电磁信号环境,检查机载雷达工作性能、探测和跟踪精度及飞机综合火控系统性能。我担任了该软件系统负责人。

我成功的将软件产品线技术引入我部,复用构件库并对构件库按照产品系列进行改造,加强核心资源的形成,将系统模块化,复用构件的集成测试,使系统高效、高质量的圆满完成,并通过空军广州五所的第三方测评,得到了解放军总参装备部的认可与好评。但现在看来,如何在缺少通用的组装结构标准情况下减少开发风险,可重用性和可协调性等方面值得进一步探究。

正文:

伴随我军武器装备国产化的进程,“射频半实物仿真目标系统”作为“机载××PD火控雷达性能开发与评估实验室”的重要组成部分。从2003年立项到2005年交付军方使用,历时两年完成。目前运行状态良好,各项功能、性能指标都满足设计要求,得到了解放军总参装备部的认可与好评。我担任了该软件系统负责人。该系统采用分布式联网试验,主要任务是试验机载雷达的各项技术指标,模拟较逼真的雷达信号环境和其他电磁信号环境,检查机载雷达工作性能、探测和跟踪精度及飞机综合火控系统性能。产品主要由主控计算机、射频源控制计算机、阵列控制计算机、校准计算机和监测计算机组成,主控软件是射频仿真目标系统的控制中枢,它对整个系统进行调度,控制其它分系统的工作,并接收作战导演系统传来的初始化数据和目标RCS特征等各种数据,然后做相应的处理把处理数据传给射频源控制软件和阵列控制软件。射频源控制软件通过信号特性模型模拟雷达回波目标信号和电磁干扰信号。阵列控制软件根据主控计算机传过来的目标位置控制面阵上开关矩阵及馈电通道的程控移相器、程控衰减器实现目标运动轨迹的模拟。

我部是一家从事目标射频仿真、雷达模拟器、电子侦察干扰的专业研究部,为部队建设过大批以射频仿真为基础的武器装备和实验室产品。目标射频仿真、雷达模拟器、电子侦察干扰等均以“射频半实物仿真目标系统”为蓝本,这三类产品的区别只是数据来源、数据处理等形式上的差别。目标射频仿真、雷达模拟器的数据来源于作战导演系统,目标射频仿真是按照作战导演系统的数据要求对真实目标进行模拟,给雷达提供电子侦察信号,与目标射频仿真的区别是目标的信号宽度和信号样式有所区别。雷达模拟器是对雷达的模拟,给反辐射导弹等提供攻击目标。电子侦察干扰是对真实的雷达的信号进行角度欺骗和距离欺骗、速度欺骗,实际上也是射频仿真系统的变形,只是来源是雷达的回波信号。既然我部的三种产品都是基于射频仿真,那么我们就可以实施软件产品线技术。所以,我在项目负责人碰头会上,就提出了在产品中实施软件产品线技术的构想,并详细阐述了软件产品线 and 软件构件技术、面向对象之间的关系。结合我部目前软件开发的特征:面向对象开发,并且积累了大量实现系统功能的软件构件库,表示要提高软件的质量和保证软件开发的进度,就必须走软件产品线的道路。当时就得到了项目专家组专家的肯定,并确立了以该项目为基础进行可复用资产整理。

在工程实施时,我们首先对以前的可复用资产进行整理:

1、以前开发的系统的设计结构、数据流程、数据规则做全面的分析,对所有系统文档做一次彻底地整理,将系统的源代码统一收集管理。

2、在上述的基础上,对原来系统一些好的流程,效率高、质量好的系统组件当满足当前体系结构的情况,分别从代码中抽取出来,并对有价值 and 文档和代码、组件实行数据库管理。详细地记录了各组件在原系统中的使用环境,编程语言、接口定义等,为下一步新系统的设计提供了大量的“资产”。

核心资源的建立。核心资源就是在射频仿真软件中能够共享的部分。软件产品线最难开发的是核心资产的分析与建模,如何从用户重多不同需求中抽象出共性的东西,如何使得核心资产通过继承、参数化等方式能够组装成用户实际需要的产品,我以及我们公司的系统分析员做了大量工作。概括起来我们用了如下2种方式:

(1)加强核心资产开发的灵活度,部分产品作成用户能够自定义的功能,彻底免除产品化时的问题,但这样做难度很大,实施周期长;

(2)根据流程最长的需求开发所有需要的构件,构件间的接口做成松耦合。演化成产品时进行构件组装。

由于我们的核心开发小组的成员就是原来各系统开发的核心成员,对原系统都有充分的理解,加上成立统一的核心小组后,更方便程序开发人员的之间的交流与沟通,我们很快就核心资源的需求达成了共识,对于需求的分析和核心资源的设计,由于核心资源涉及到的系统边界较多,特别是数据的存储与访问接口,因此我们使用了Rose建模工具,通过对原有系统的逆向工程和建模分析,理出了核心资源的接口设计和核心资源子系统的划分和子系统内各设计类,最终有利于我们建立合理的核心资源软件架构。

数据分解合成部分,这部分是我们碰到的难点。目标射频仿真、雷达模拟器、电子侦察干扰的数据来源分别是作战导演系统的初始化数据,雷达模拟器来源于雷达处理机,电子侦察干扰的雷达频率来源于瞬时测频信号,其他数据来源作战导演系统的初始化数据。由于数据来源多而且比较“杂”。在试验中,大量的时间用于数据的分解与合成,如能合理快速的进行数据分解和合成将有效提高数据的通用性并提高系统性能。通过分析,我们将目标射频仿真、雷达模拟器、电子侦察干扰全部当作“目标”信号来处理,将“目标”信号的工作频率、载波频率等统一为国际单位,减少单位换算,将目标方位、俯仰、距离、速度统一为同一到机体坐标系,减少坐标转换,过去由于数据来于不同,有的数据要进行机体坐标系到大地坐标系的转换,有的要将雷达坐标系转化为大地坐标,现在统一后相当与整个阵面是在空中“悬浮”,从而减少了大量的数据解算,但目标射频仿真、雷达模拟器、电子侦察干扰的数据不同部分分别按照各自的特征形成联合数据结构,跟在工作频率、载波频率目标方位、俯仰、距离、速度等的后面。以上所有数据均由主控计算机进行统一解算并发送到通用公司的VMIC反射内存网络上,由阵列控制计算机和射频源计算机使用。并决定在没有更好的快速网络出现的情况下,将VMIC反射内存网络作为我们的必选设备,以有效提高系统实时性能,并将目标射频仿真、雷达模拟器、电子侦察干扰三种产品的数据分解合成实时性能都提到到2ms左右。

阵面信号合成、处理部分,根据处理的通用性,将处理包分为通用数据处理包和专用处理软件包,将阵列数据、射频源数据处理等通用数字信号处理方法放到一个包中,将雷达目标合成、雷达模拟、电子干扰等放到一个包中,实现了方法模块的高效复用。同时使用简单工厂模式实现了处理模块的动态加载。这个部分技术比较成熟,在我们原有的信号处理构件的基础上只做了少量的改动,主要的是将信号合成后的数据编排上做了些改动,主要优化了信号产生的速度性能等。

结束语:

通过使用软件产品线技术,最终有效地提高了我部软件生产效率和质量、缩短了开发时间、降低了总体开发成本。但是也遇到了一些问题:1)核心人员原先都没有从事软件产品线的开发,往往只关注自己熟悉领域的软件设计,没有全局的观念,为此我们通过加强团队人员间的沟通,互相了解彼此的领域,使开发人员的全局观念加强,另外通过软件架构设计培训,提高软件人员的架构观念;2)没有合适的构件集成工具,处理流程的建立主要依靠代码来实现,因此在未来的开发迭代周期里有必要开发集成工具。

论分布式数据库的集成

摘要:

本文讨论了某公司发货系统的分布式数据库集成解决方案。该公司由于业务的发展,要在另三个城市设立货仓进行发货。为此,需要增加原先的 MIS 系统实现这一功能。公司委任我作为项目经理完成系统的设计和开发的工作。我经过分析,使用了 Sybase 的分布式数据库技术。我设计的这个系统是采用典型的 C/S 结构,但客户端连接服务器的网络采用电话线拨号,速度有限,传统 Windows 界面的客户端应用程序相应速度比较慢。于是我采用了优化数据库结构的方法,把数据分两部份存放,基础数据放客户机,销售资料主要采用键码放服务器,应用程序再现数据时从服务器取键码,到客户机取对应的解释。由于键码的数据量少,网络传输便快。在构建这个分布式数据库系统的过程中,我着重研究并解决了数据同步和事务协调的问题,到得了良好的应用效果。

正文:

2004 年 3 月,由于公司业务的发展,要求在其它三个城市设立货仓,处理发货业务。公司本部运行着一套用 Sybase 数据库的 MIS 系统可以实现发货,该系统用的是 C/S 结构。由于客户端连接服务器的网络采用电话拨号,所以直接把客户端软件直接安装在外地访问本部数据库,速度很慢。于是,公司成立了一个项目,专门解决这个问题。在这个项目中,我担任项目经理。经过对现有系统的分析,我们决定利用 Sybase 提供的技术,采用分布式数据库集成的方法来改造目前的系统使之能适应新的需要。项目分三个阶段进行,一是进行需求分析,确定要增加的功能。二是进行系统设计,改变后数据分布如何,系统架构如何。最后是实现和测试,上线。整个项目历时从分析到实现历时三个月,最后于 2004 年 6 月份系统成功上线。

在分析阶段时我发现由于客户端地域的分散,遍及三个省境内,连接服务器数据库的网络采用电话拨号方式,速度有限,在使用客户端应用程序时感觉界面速度很慢。我经过分析,认识到许多操作都要从服务器中取数据,速度慢就慢在数据访问上。服务器是没有瓶颈的,问题出在网络速度上。出于成本和业务量方面的考虑,公司不会用专线连接,只能是电话拨号。这时只能改变目前软件的实现方法,来适应这种低速网络的使用模式。

经和项目组的人员一起探讨,结合关系数据库的知识,我认识到,应用程序的每一次数据库操作,都要访问多个相联的表,其中,有销售订单表和物料基础数据表/客户资料表/货仓的基础数据等。销售订单表中存放着出销售的订单编号,成品编号等,数据量少。而基础数据表则就放着成品的相关信息,有大量的数据。如果考虑把销售订单放在服务器,基础数据放在客户端,当应用程序中访问数据时,总是从服务器上存取销售订单,从客户端提取成品/订单的详细信息。由于订单的数据量少,便减少了网络上传递的数据量,从而提高了界面的响应速度。

把数据分散存放只是工作的第一步,接下来要考虑应用程序怎样访问这种分布式数据。开发应用时,如果每一功能都针对两个数据库进行,就带来了很多麻烦。所以,我通过研究 Sybase 的分布式数据库技术,决定采用 CIS(组件集成服务)部件,来合并两个数据库成一个统一的分布式数据库。应用程序只要连接一个数据库,就可以透明统一访问到两个数据库中的数据。

该技术具体实施方法是:在客户端数据库中建立一个对服务器数据库的远程访问服务名,包含访问地址,登录用户名,登录密码等关键的连接信息;前且对服务器中销售订单建立一个本地代理表。结构和服务器中远程表完全一样,它是访问服务器中会员资料的中转和代理。客户端应用程序访问本地代理销售资料表时,实际上是通过预先定义的远程访问服务名中包含的连接信息到服务器中对应的实际销售资料表中访问数据。这种访问对于客户端完全透明,感觉不到是从物理上独立的两个服务器中存服数据。所以,这种数据库结构是典型的分布式数据库。

部署这种分布式数据库不是难事,只要在客户端和服务器上安装 12.0 版本以上的数据库服务器,在客户端服务器上建立远程服务名和代理表即可。由于 Sybase 数据库的安装支持脚本方式,在客户端应用程序的标准安装过程中,嵌入 Sybase 数据库的安装和配置脚本,就自动化地完成了所有工作。

在实际使用该分布式数据库系统的过程中,遇到了几个问题,第一,数据同步。客户端基础数据不是绝对静态的,也有变化,因此在服务器要设置一个统一的基准,称为主点数据。客户端总是要复制使用,称为复制点数据。如何及时感知到服务器端主点数据的变化,有效率地复制到客户端,是个难题。Sybase 针对这种应用场合,提供了复制服务器技术,但为了避免过于复杂,我们采用实际应用程序来管理同步。当服务器端主点数据有了更改时,保存一个相应的标识和时间戳,客户端应用在登录服务器时,检查这些标识,一检测到了数据有更新,就首先下载,然后再进入系统正常使用。这种方法实现起来,增加了额外的开发量,且不能判别绕过应用程序对数据的直接修改,但是,是最简单和有效的方法。

第二个问题是事务协调问题。物理上独立的两个数据库,在协同操作时,如果服务器正好停机或者网络故障,完整的一个事务没能完成,就会“事务崩溃”。虽然 Sybase CIS 内嵌了两阶段提交技术,能够自动恢复。但是应用程序在这种情况下,敏感性不够,操作界面会无端凝固,影响了使用的方便性。我针对 PB 对劲于连接的判断和感知,用了一个小小编程技巧,使应用程序能够及时感知到数据库连接故障,及时停止和恢复事务,使操作界面表现友好灵活。

在具体的应用中,我们在三个城市安装了增强的客户端应用程序,同时安装了 Sybase 数据库。初始化时,把基础数据放从公司本部的数据库导入客户端的数据库中。用户在外地进行发货时,先拨号上网,然后启动客户端程序。在登录过程中,客户端程序会检查服务器上的标识和时间戳检查这些主数据是否有更新,如果有就先下载,下载完成后进入系统正常使用。在服务器更新的数据比较多的情况下,下载的时间会比较长,这时如果遇到急需发货,则会影响到货物不能及时发出去。为了解决这个问题,我设置了在每天凌晨的某个时刻自动登录和启动客户端程序,在下载更新数据完成后自动关闭。这样可以把一部份数据更新的内容放在非工作时间内完成,减少了发货登录的时间。用户登录后开始进行发货操作。输入销售订单,通过本地代理表系统自动到服务器获取该销售订单的数据,如发货的数量,客户编号等。而一些基础数据则可以直接从本地的数据库中得到,如销售产品的描述,客户的地址/电话/传真,发货的库位等。完成出货动作后,会自动更新服务器的库存。而这一更新通过提交事务在后台进行,不影响前台的操作。所以,对用户来讲,能够进行正常的操作,不会因为速度慢而进行不下去。

在当今的信息社会里,互联网带来了相互连通的方便,而且知识爆炸,数据的分布式访问是个必然的趋势。目前新起的 XML 技术,提供了各种平台数据库之间的一个公共数据访问标准,可以用来构建更加灵活,适应性更强的分布数据库技术。将 XML 用在分布式数据库中,将是未来的一个趋势。

论基于 Web 服务的企业应用集成

摘要

去年，我所在的公司承担了一项为 X 卷烟厂建立企业应用集成的项目。主要目标是将该厂原使用的办公系统、人力资源系统、生产管理系统等系统进行集成，使之成为一个无缝连接的整体。我有幸参与了该项目，并担任主要的规划设计工作。

针对该项目，我们综合运用了 J2EE 平台构架、Web service、IBM WebSphere 等技术，来实现基于 Web Service 的企业应用集成，其效果相当显著。而且如果今后有新的系统加入，也能够很容易地添加。用户在使用该系统后，生产、经营效率有了大幅度的提高，得到了用户的一致好评。

正文

去年，我公司承担了 X 卷烟厂的企业应用集成项目。该卷厂是一家大型国有企业，经过多年的发展和积累，形成一套行之有效的管理机制，并实施了一系列的管理、生产自动化系统，如 LOTUS 的 Domino 办公自动化系统、人力资源管理系统、CRM 客户关系管理系统、用友财务软件等商品化软件，也有该厂与一家软件公司合作开发的生产管理软件（具有源代码）。这些软件实施于不同时间，也为各部门的生产、管理效率的提高起到了巨大的推动作用。但是各个分散独立的系统也给企业的发展造成了瓶颈，生产第一线的数据不能随时反馈给决策层领导，供应链、销售链不能与生产管理系统衔接，只能用手工方式将数据输入生产管理系统，不仅速度慢，而且经常容易出错。

随着企业的发展，这些系统也成为了一个个信息孤岛，给企业的进一步发展造成了瓶颈。为彻底消除这些限制企业发展的因素，该厂决定实施企业应用集成，将各个独立的系统集成起来，成为一个无缝连接的整体，使各种信息在整个企业范围内流动起来，促进企业的进一步发展。

我公司成立了项目组，我有幸参与了该项目，并担任主要的规划设计工作。

根据以往项目的经验，我们首先进行了技术方案的选择。目前企业应用集成目前通行的做法有三种：数据层集成、业务层集成和表示层集成。数据层集成需要修改大量的应用程序，而使用的系统中一些是商业软件，我们无法得到源代码。而表示层集成只是集成了各个应用程序的显示界面，就好像只是做一层外装修，无实质性的改变，而且有可能成为整个集成系统的瓶颈。业务层集成则克服数据层、表示层集成的这些缺点，而程序代码的修改量不大，且只是针对没有 API（应用程序接口）的应用程序。因此，我们根据企业目前系统使用的状况及我们以往开发类似项目的成功经验，经项目组讨论决定采用业务层集成方式来完成该项目。

我们采用了 J2EE 的平台构架，并为企业将来在企业间和电子商务的进一步集成，采用了基于 Web 服务的集成方案。Web 服务具有各个应用程序之间松散耦合、位置透明、协议独立的特点，某个应用程序的增减变动不会影响到其它应用程序的运行，并且它所使用的 SOAP、XML 等协议和技术也不被某一家公司所控制。

选定了实现平台，我们开始对各个系统进行封装、组合。各个系统的接口的异构性带来了访问上的麻烦。为了屏蔽接口上的差异性，我们采用了将这些接口映射为统一的虚拟组件。虚拟组件一边通过 J2EE 低层接口显示现有应用程序的功能，另一边通过现有应用程序接口 API 和它们进行连接。这样用户就可以用统一的方式访问虚拟组件而获得应用程序的功能。我们通过采用 J2EE 的 JCA 方式实现虚拟组件对各个应用程序的访问，为每一个应用程序定制一个资源适配器，并插入 J2EE 的应用程序服务器中。资源适配器含有一个指定的原系统库（用 Java 写的或带有自身接口单元的库），这个库同它描述的原系统具有连通性。通过 J2EE 平台的 JCA 这样一种方式，我们向用户提交了一个统一的调用接口。

解决了接口问题，接下来就要将各应用程序提供的服务通过 WSDL 描述服务，并且注册到企业内部的 UDDI 中心。我们采用了 Apache axis 1.0 来开发 Web 服务，IBM 的 WebSphere 作为消息服务器。根据该厂的要求，将各应用程序提供的服务分为 5 个种类来提供 Web 服务，分别是销售、人事、生产、供应、办公五个部分，再将各应用程序提供的 Web 服务具体细分，如生产部分提供了物料需求计划服务、库存服务、产品目录更新服务等。并 WSDL 实用程序生成代理代码，发布 Web 服务程序；在统一的服务发布服务器上，通过 UDDI 服务实现发现 Web 服务功能，便于查询系统各模块提供的 Web Service 服务。服务请求者按照标准接口构造 XML 格式的 SOAP Request 并通过 HTTP 方式向 UDDI 注册中心发出请求。当有一个新的物料需求计划生成时，我们通过使用 JMS 中间件来发布物料需求计划，使得订阅该主题的供应商得到消息。

企业信息门户 EIP 我们采用了 IBM WebSphere Portal Server 来实现。对用户帐号采用统一管理，统一授权，用户登录系统时采用统一认证服务调用接口以实现用户身份的认证过程，并通过单点登录 SSO，从根本上改变了现有各应用程序帐号之间进行漫游，不需要进行重新登录，给用户提供了极大的便利，并减少了不法用户利用认证攻击应用程序的可能，极大地提高了系统的安全性和可用性。

在系统实施过程中，我们也遇到了一些困难。

生产部门使用的生产管理系统是由该厂与一家软件合作开发的，没有预留 API，我们便对该系统进行了改造，由于该软件公司已找不到，而且该系统文档难以找到，因此我们对接口改造会对原应用程序造成的兼容性、稳定性的问题估计不足，改造完成后只进行了接口改造的回归测试，使得该应用程序一直不稳定。于是我们重新对该系统进行了深入透彻的了解，决定采用 CORBA 规范的接口定义语言 IDL 描述接口，而 J2EE 与 CORBA 又有极大的相关性，便于今后的封装、组合。接口改造完成后，我们进行了反复的回归测试和稳定性测试，才使系统又恢复原来的运行水平。

还有就是与各应用程序供应商的协调，大家都站在各自的立场上，设想对方的系统接口应该如何改，以适应自己的接口。经过我们和该厂的努力，最终大家都同意以最有利于集成系统的角度进行改造。

在本项目中，我们综合运用了 J2EE 平台构架、Web service、IBM WebSphere 等技术，来实现基于 Web Service 的企业应用集成，其效果相当显著。而且如果今后有新的系统加入，也能够很容易地添加。用户在使用该系统后，生产、销售、供应、办公成为一体化的无缝连接的一个整体，生产、经营效率有了大幅度的提高，得到了用户的一致好评。

我从实践中领会到，企业应用集成是一项技术复杂，工程繁杂的工程，但也是企业目前最急需解决的问题。同时我也认识到，企业应用集成还有很多领域和知识点需要继续研究和实践，新技术的应用也带来了新的要求和挑战，我将在企业应用集成领域不断探索，不断实践。

论企业应用集成

摘要:

2004 年 10 月,我参加了***车站综合信息平台项目的开发,承担项目的方案设计任务。该项目力图通过对车站现有信息子系统的集成,以达到共享各子系统的数据库,优化企业运输作业流程,提高企业经营管理水平之目的。

本文结合笔者的实践,以该综合信息平台建设项目为例,讨论了企业应用集成技术。在本着集成、开放标准、管理配套的原则下,提出了基于 Java 技术的 J2EE 应用服务器作为统一的应用集成平台,以集成适配器作为系统集成架构模式的总体设计思路,并着力介绍了该项目关键部件——集成适配器的构建过程。还就项目的具体实施作了详细叙述。最后,提出了企业应用集成的持续性,并确定了下一步集成的目标。

正文:

2004 年 10 月,我单位承接了***车站综合信息平台的建设任务。由于我具有多年的铁路行业软件项目开发经验,所以我有幸被单位指定为该项目的负责人,主要负责项目的方案设计工作。

该车站是一等客货运编组站,车站所在地是矿区,有规模不等的国有煤矿及个体煤矿数十个,车站主要以煤炭运输为主。近些年来,随着铁路 TMIS 系统(铁路运输管理信息系统)建设的逐步深入,该车站建立了若干相应的应用子系统,主要有列车确报系统、车站现在车系统、货票制票系统、车号红外线自动识别系统、货运计划系统、货运安全系统及货车轨道衡计重系统等。车站希望通过综合信息平台的建设达到以下几个目标:(1)实现各子系统间数据共享。(2)能够实时地向企业客户(货主)发布请车、承认车及货物运价调整等相关信息。(3)通过各子系统的应用集成,使车站运输作业流程得以优化。

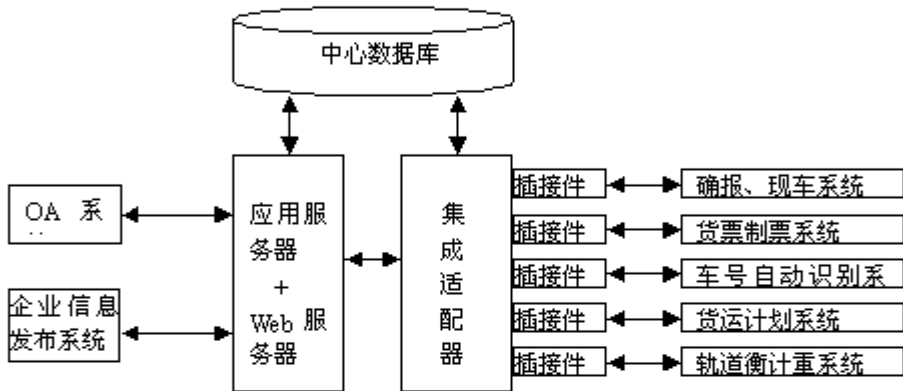
在设计综合信息平台建设总体方案时,我充分考虑了下面三个原则:

集成原则:综合信息平台最主要的目标是整合车站的信息资源,在考虑 最大化集成各个信息子系统时,应该避免产生新的“信息孤岛”。

开放标准原则:在综合信息平台建设时,应该站在整个系统宏观的高度,采用开放的标准和统一的架构来集成各信息子系统,避免各子系统“点到点”的低效落后的集成方式,以利于将来其他新的系统能够便利、无缝的整合。

管理配套原则:建设综合信息平台的目的是通过信息共享,达到业务流程的优化,以提高车站各方面整体管理水平。因此项目设计之时,应充分考虑企业管理方面的需求。

在遵循上述原则的基础上,经过对企业需求的认真分析,结合当今成熟的 EAI 技术,我提出了以基于 Java 技术的 J2EE 应用服务器作为统一的应用集成平台,以集成适配器作为系统集成架构模式的总体设计方案。如下图所示:



本设计方案从集成的广度来说,既包括了数据的集成,也包括了应用的集成;在集成的方法论方面来讲,大部分系统采用可白盒集成方法,少数系统采用的是黑盒集成方法。

根据综合信息平台的总体设计方案,集成适配器是该信息平台的最关键部分,它负责不同子系统之间数据的采集、转换和交流。因此,集成适配器的选型或设计的合理与否,是项目成败的关键。由于车站各信息子系统存在操作系统平台和数据库的异构性,无法从现有的中间件中找到完全适合的产品,因此我们决定自行开发此中间件。由于铁路各车站的业务领域和业务流程存在高度相似性,所以,此中间件有较高的复用价值。

对于集成适配器的设计,我采用了一种“可插拔”的设计理念。即为每个需要集成的子系统单独设计一个插接件,该插接件负责为与之相连的子系统提供数据及应用接口。各插接件通过 XML 格式的装配文件,自由组装到集成适配器这个容器中。集成适配器为所有插接件提供一个统一的调度模块,来协调和指挥所有插件,使之能够协同运作。

在集成适配器的开发中,我选用了开源的集成开发环境 Eclipse 作为开发平台。对于集成适配器调度模块的开发,我们采用了 Eclipse 提供的 Jobs API,Jobs API 封装了 JDK (Java Development KitTools) 的定时及同步方面底层 API,降低了编程的复杂性,提高了开发效率。在插接件的开发过程中,我们充分利用 Eclipse 的高度可扩展特性,在因特网上搜集该项目可用的插件,以这些插件为扩展点,来扩展我们自己的插接件。最后我们利用 Eclipse 提供的 RCP (Rich Client Platform) 技术,集成我们的功能插件,并生成独立于 Eclipse 平台的、可独立执行的集成适配器。

各信息子系统具体集成过程及效果如下所述:

(1) 轨道衡计重系统

该车站的轨道衡设在矿区与车站之间,距车站站场大约 2 公里左右,从矿区到车站的重车要经过轨道衡检斤,对于超重货车,要通知车站相关部门处理。由于该系统是个独立的单机系统,计重软件采用的是 VFP 编制,数据库是单机的 DBF。因此,我们首先从网络方面进行了集成:利用一对网桥将该系统接入车站 TMIS 系统。重车通过轨道衡会产生结构为:(车辆顺位、车号、车种、铁重、计重)数据,与轨道衡相连的插件会通过 JDBC--ODBC 桥,将数据转换为两个不同格式的副本,一份写入中心数据库,一份作为矿区站发来的确报报文,写入确报系统的到确报库中。

通过该系统的集成,轨道衡工作人员免去了每天去车站递送过衡报表的工作,车站相关部门可随时通过浏览器查询过衡数据,并能方便的生成各种统计数据;车站的车号员通过查收确报报文,即可掌握过衡列车的组成内容,减去了每列车都要到现场抄车号,回去录入的工作流程,大大减轻了劳动强度,提高了工作效率。

(2) 货票制票系统

该系统的集成我们采用了批量文件传输(FTP)方式。货票制票机在制票完成后,会自动向分局传输货票数据。因此我们在货票传输地址表中增加一个条目,使货票数据在传送分局的同时也传送到集成适配器,与之相应的插件将到达的 80 列格式货票数据进行解析,形成结构为:(票号、车号、车种、计重、运价、发货人、收货人)的数据,利用 JDBC,一份写入中心数据库,另一份通过车号与现车系统的车辆库匹配,写入相应信息。

通过该系统的集成,减轻了编制出发列车确报的工作量,因为输入车号后,与该车号相关的货票信息会自动生成,免去了去查阅货票步骤。另外,该系统的集成,使货主通过因特网查询货票信息成为了可能。

其他的信息子系统如确报、现车系统、货运计划系统、车号识别系统等等,也都通过各自的接口插件进行了数据或应用的集成,在此不再详述。

(3) OA 系统和企业信息发布平台

通过 TMIS 各子系统集成而建立的中心数据库,是办公自动化和企业信息发布平台的数据基础。通过设在货服大厅的大屏幕显示系统,能够实时动态地发布与货运业务相关的系统,方便了货主,提高了办公的透明度。

通过项目组成员努力工作,车站高层领导的高度重视,以及车站相关人员的通力合作,历时四个月,该车站的综合信息平台初步完成,达到了建设之初的需求目标,得到车站的一致好评。系统到目前为止运行稳定。

我们应该知道,企业应用集成是一个持续集成过程,是一项长期、不断进行的工程,不能指望短时间内达到深度集成。随着铁路信息化建设的深入,还会有新的应用需要整合。我们下一步的目标是建立该企业的企业门户,使我们前期的后台整合成果,能够在 Internet 上得以展现。

论企业应用集成

摘要:

本文讨论了某公司的应用系统集成项目。某公司为了应对市场变化的需要,决定把公司几个主要的应用系统 ERP 系统, PDM 系统, E-mail 系统集成在一起,系统集成完成后,ERP 系统可以与 PDM 系统交换数据,大大减少了重复工作。通过 ERP 系统与 E-mail 系统的集成,可以把 ERP 出来的报表自动发送给相关人员。我作为该项目的主要负责人之一,担任了系统分析和设计的工作,通过分析需求,设计三层体系结构,选择合适的平台等措施使项目能顺利完成。在项目实施过程中,我发现 XML 作为新的 WEB 开发语言,应是今后选择的一个方向,并且企业应用集成是一个不断发展的过程。

正文:

企业的应用集成(EAI)是指在企业范围内将多个应用系统的过程,软件,标准和硬件集成起来,使其成为无缝运作的整体。目前企业应用集成正越来越受到人们的重视。我在 2004 年 1 月参加了公司应用系统集成项目,作为项目的主要负责人,我担任了系统分析和设计的工作。项目首先是和相关部门的用户一起讨论系统集成内容,用户希望完成系统集成后能提供什么样的功能和服务。确定了这些需求后,就进行软件开发平台的选择和集成方案的设计。最后进行开发测试。测试完成后上线正式使用。整个项目用了五个月的时间完成,在 2004 年 6 月交付使用。

该企业的信息化程序比较高,主要的应用系统有:ERP 系统,实施了物流和财务模块,把公司的采购,财务集成到了一起;PDM 系统,产品的设计和开发在该系统上进行;E-mail 系统,主要是收发内部和外部的电子邮件。但是,随着企业的发展和市场竞争的激烈,问题也逐渐暴露出来。因为三个系统是独立的,没有数据的交换和共享,这样,大量相关的数据不得不重复输入。象 PDM 系统负责产品数据的维护。当一个产品的 BOM (Bill of material, 物料清单)成熟后,要把该 BOM 导入到 ERP 系统。因为两个系统没有关联,这样就需要安排人员专门负责数据的录入,往往要加班加点,且容易出错,很不利于产品快速地生产并推向市场。为此,要考试应用集成,把企业的这些“信息孤岛”联系起来实现信息的交流。我在分析了企业应用系统的现状后,分以下几个步骤来实现该企业应用集成。

一、仔细研究现有的系统之间的关系,确定要集成的内容并考虑实现的方法

首先是 ERP 系统和 PDM 系统的数据的交换,要求在 PDM 系统开发出来的 BOM 下达后,要自动导入 ERP 系统。因为两者针对的是 BOM,结构是一致的,实现两者的数据集成,只需定义统一的数据接口和格式即可。其次是 E-mail 系统和 ERP 系统的集成。用户希望有 ERP 系统批准的采购订单能自动发送给供应商而不必手工要从 ERP 系统中下载下来整理后再发给供应商。驻外地的销售人员要求能每天收到关于成品库存的 E-mail。通过分析,发现公司用的 E-mail 系统是 Microsoft Exchange Server,而 Exchange 提供了与外界的 API 接口,只要照接口规定的格式填写,如发件人,收件人,信件标题,内容等,进行填写的文件,转送到了 Exchange Server 后,Exchange Server 就可以将其发送到指定的收件人邮箱。利用这一功能,可以在 ERP 上开发接口程序,下载符合要求的文件到 Exchange Server 上实现 E-mail 自动发送。

二、设计企业应用集成的体系结构

为了避免传统的点到点的系统集成的缺点,我提出了三层的系统集成体系结构,即把企业应用系统分成表示层,中间层和企业信息系统层(包括数据系统),企业信息系统层由该企业的 ERP 系统, PDM 系统和 E-mail 系统组成,它们通过一个面向消息的中间件来实现数据的交换。中间层主要实现应用的业务逻辑和各种服务支持,它可以访问企业信息系统层运行且与应用相联系的数据和函数。比如前面说的 ERP 系统与 PDM 的数据交换,ERP 系统与 E-mail 系统的接口就在中间层实现。数据从底层进入中间层,在这一层完成数据交换和集成的功能。这样通过中间层就可以实现三个系统数据的共享。客户层则包括不同类型的客户端应用,之前三个系统都是 C/S 结构,各自有自己的客户端程序运行在个人计算机上,集成后,要开发出基于 WEB 的客户端程序,统一界面,方便访问。实现 B/S 与 C/S 共享。这样,通过中间层实现不同系统的协调工作和对企业各种事务活动的支持,屏蔽了低层的接口和技术细节,使数据能方便地交流。

三、选择合适的应用集成平台

目前,可作为开放式企业应用集成的规范和平台的主流技术有两种:一种是微软公司的 COM+规范和 Windows .NET 平台,另一种是 SUN 公司的 EJB 规范和 J2EE 平台。在平台的选择上我进行了反复的比较,最后选择了 J2EE 平台。因为 J2EE 平台的开放性与支持异构性,可移植性,支持的广泛性。对企业现有遗产系统的继承性和技术优势等。更重要的是它具跨平台的功能。公司的 ERP 系统是运行在 HP Unix 服务器上的, PDM 系统用的则是 Microsoft NT 操作系统, E-mail 是 Microsoft 的 Exchange Server,这些应用使用不同的操作系统和平台,而微软的 .NET 只能用在 Windows 的操作系统。所以基于公司的实际, J2EE 平台是合适的选择。选择 J2EE 平台存在的问题在于,一是成本相对比较高。二是公司的 IT 人员对 JAVA 还不熟练。为此,我在确定选用该平台后,先了解该平台的软件构成,只购买了我们需要的开发软件,从而节省了成本。对 JAVA 则安排了几次培训,使 IT 人员能很快地上手用 JAVA 开发程序。

四、应用集成方案的实践

在 J2EE 平台上,我通过 JAVA 在业务逻辑层的开发,实现了以下的业务流程。首先,在 PDM 系统的产品结构成熟后将会有有一个下达的动作,然后 BOM 数据通过中间层的处理自动导入到 ERP 系统,这能过之前已定义好的接口规范和数据格式可以很空易地实现。ERP 接收到产品结构数据后,即可以运行 MRP 产生采购订单和生产订单,投入产品的生产。此外,通过业务逻辑层,在 ERP 系统上下达 PO 后,采购订单的数据会自动传到 Exchange 服务器,然后通过 E-mail 自动发送给供应商。产品库存的数据也可以从 ERP 系统传到 E-mail 发给驻外地的销售人员。该应用系统集成后,用户不必再加班加点地在 ERP 系统中维护产品数据,并且保证了 ERP 的产品结构与 PDM 中的是一致的。由于实现了 ERP 与 E-mail 系统的集成,用户也不用担心哪张 PO 没有发送给供应商了。

项目上线后,满足了用户的需求,运行半年多来,系统基本稳定,大大提高了工作效率,得到用户和管理层的肯定。

但也存在着一些问题。在项目实施的后期,我们发现用 JAVA 编写代码效率比较低,运行速度慢,并且取数据还是不太方便。于是引入了 XML 技术进行数据的抽取,组织和表现,结果大大增加了开发的效率。通过该项目我认识到 XML 将会是今后 WEB 开发和实现企业集成的主要技术之一。此外,现在仅是实现了 PDM 系统和 ERP 系统的集成,之后随着企业的发展,ERP 将运行在更开放的平台上。会从企业的内部应用转为一个连接到 WEB 上的分布式应用系统。即 ERP II,这将是以后的发展方向。

财务数据仓库系统的设计与实现

【摘要】

近年来,数据仓库技术在信息系统的建设中得到了广泛应用,有效地为决策提供了支持。2004 年 6 月,本人所在单位组织开发了财务管理决策系统,该系统主要是使高层领导掌握企业的经营状况及进、销、存情况,分析市场趋势。

本文通过对财务数据的分析,结合数据仓库开发原理,完成对财务数据仓库的数据组织,介绍了财务数据仓库的设计和实现方法方法。财务数据仓库的设计步骤主要是遵循数据库设计的过程,为分概念模型的设计、逻辑模型设计、物理模型设计和数据仓库生成等几个阶段。

目前,该项目已顺利上线,领导反映良好。在该项目中,本人担任系统分析师职务,主要负责系统架构设计和数据仓库的设计工作。

【正文】

2004 年 6 月,我所在的单位为了快速适应市场的变化,使高层领导及时掌握企业的经营状况及进、销、存情况,分析市场趋势,决定开发财务数据仓库系统。在该项目中,本人担任系统分析师职务,主要负责系统架构设计和数据仓库的设计工作。

在这个系统的设计过程中,我们遵循了数据库设计的过程,整个财务数据仓库的设计步骤如下:

- (1) 概念模型的设计;
- (2) 逻辑模型设计;
- (3) 物理模型设计;
- (4) 数据仓库生成。

1、概念模型的设计

进行概念设计所要完成的主要工作有:

(1) 决策需求分析:对于数据仓库系统而言,决策者最为迫切的需求在于,更加准确的掌握企业的经营状况及进、销、存情况,包括分析进货趋势,分析销售市场波动趋势,分析企业存货情况,分析市场经营状况发展趋势。所要求的操作数据库的数据有商品进货数据、商品销售数据、商品库存数据、顾客信息和销售商信息。

(2) 确定系统的主题域及内容在上述需求分析的基础之上,我们可以确定企业财务仓库系统的 3 个主题,分别是商品、顾客和销售商。如图 所示。

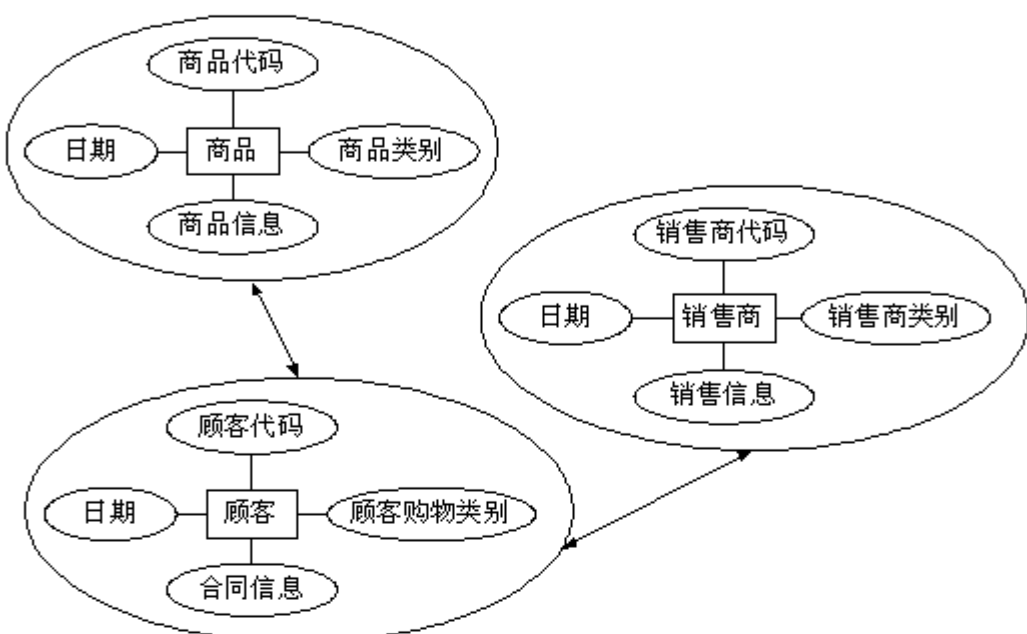


图 财务数据仓库分析模型图

2、逻辑模型设计

商品、顾客、销售商是财务数据仓库的 3 个主题,是其经营运作 3 个框架。商品主题描述企业商品分类及销售情况;顾客主题描述了企业对顾客进行分类及有关顾客合同的管理情况;销售商主题描述了企业销售人员销售商品及销售地区情况。其中商品主题作为中心,将这 3 个主题联系起来。它们的内容列出如下:

(1) 商品:商品固有信息(商品代码、商品名称、商品类别等);商品库存信息(商品代码、库房号、库存量、日期等);商品销售信息(商品代码、顾客代码、销售日期、销售单价、销售数量等);

(2) 顾客:顾客固有信息(顾客代码、顾客名称、地址号、电话等);顾客合同信息(顾客代码、合同代码、起始日期、终止日期、数量、价格等);顾客购货信息(顾客代码、商品代码、单价、数量、日期等);

(3) 销售商:销售商固有信息(销售商代码、销售商品、销售商品名、销售商地址等);销售商地区信息(销售商代码、销售地区名、电话等)。

以“商品”为主题可以看到,首先,在从面向应用到面向主题的转变过程中,丢弃了原来不必要、不适合分析的信息,如各类领料单、出库单、入库单等;其次,在原有数据库模式中,有关商品的信息被分散在各个子系统中,如:商品销售信息存放于数据管理子系统,商品库存信息存放于商品库管理子系统中等,根本没有形成一个有关商品的完整一致的描述。而向主题的数据组织形式所实现的就是要形成商品一致的信息集合,以便在此基础上针对“商品”这一分析对象进行分析处理。

3、物理模型设计

(1) 确定数据的存储结构

通过定义功能/数据的交叉参照图,决定谁需要访问哪个范围的数据据。每个数据仓库实施的最初阶段,必须标明最终用户的词汇表,定义恰当的商业术语,与底层数据联系起来。

由于数据仓库本身通常是面向主观意识的,基于最终用户的需求,创建数据仓库的第一步是识别和分析有关的内部数据和外部数据源。

数据模型在将内部和外部操作数据转换和集成到数据仓库里的过程中起着关键性的作用。在这个阶段,系统分析师必须收集信息,实施从数据源到数据模型的逻辑转换,确定数据按简要或详细的程度保存在数据仓库中。

创建词汇表、清除数据是产品数据与仓库数据之间的转化基础。词汇表是关于数据的数据。对于决策支持分析员来说,它是一个确定数据位置、理解计算法则的商业定位指导。清除数据和过滤数据包括转换数据、巩固数据和通过应用一致的命名法则协定解决在操作数据库中数据不一致的问题。

从长远的角度优化数据仓库,系统必须是灵活的,可扩充的,模块化的,以便有足够的去适应系统的不断增长。

(2) 确定索引策略

由于数据仓库的数量比较大,因此在库结构设计时将每一个子库设计为树型索引结构,初始结点为所要决策的主题,其中间结点即为不同优先级的与该主题有关的查询角度与层次,而最终结点为经过预处理的有定义的数据集合。

4 数据仓库的实现

(1) 接口设计接口设计

在财务数据仓库中有几种方式:菜单式、问答式和图形式。接口设计涉及如下几个模式:输入响应模块、输出模块、人机对话管理模块和外围设备。由于时间和篇幅有限,这里就不详细介绍这些方式和模块了。

(2) 数据的采集

接口设计结束后,下一步工作就是将数据采集到数据仓库中。数据采集是将原始数据从多个传统事务处理数据库系统中提取出来进行清洗,集成等有关处理,使之符合数据仓库环境中对数据质量的要求后再装载入数据仓库中。

数据采集的主要工作有确定数据源的次序、元数据的管理、粒度划分、数据分割和数据的定期维护。数据在装入数据仓库底层的目标数据库后,还要完成的任务包括数据的定期清理、刷新、重建索引、初步划分主题等工作。

目前,该系统已经上线并顺利运行了一段时间,达到了开发的目标,为高层领导决策提供了可靠的数据来源和支持,得到了单位领导的一致好评。

数据仓库技术的发展包括数据抽取,存储管理,数据表现和方法论等方面。

在数据抽取方面,未来的技术发展将集中在系统集成化。它将互连、转换、复制、调度、监控纳入标准化的统一管理、以适应数据仓库本身或数据源可能的变化,使系统更便于管理和维护。

在数据管理方面,未来的发展将使数据库厂商明确推出数据仓库引擎,作为服务器产品与数据库服务器并驾齐驱。在这一方面,带有决策支持扩展的并行关系数据库将最具发展潜力。

在数据表现方面,数理统计的算法和功能将普遍集成到联机分析产品中,同时与 Internet/Web 技术紧密结合,推出适用于 Intranet 和终端免维护的数据仓库访问前端。

数据仓库实现过程的方法论将成为数据库设计的一个明确分支,并将成为管理信息系统设计的必备。

论软件需求分析方法和工具的选用

【摘要】

本文通过一个集成电路设计有关的软件项目，讨论了该项目的主要特点和本人所担任的工作，着重讨论了在项目需求分析过程中采用的具体方法和工具以及选用的理由。

由于项目的专业领域的特殊性，分两类不同的需求讨论了需求分析中遇到的问题及解决方法；在这个过程中给出了对选用的具体工具和方法的效果的描述。接着本文讨论了对使用方法的改进的一些想法以及具体的实现过程。最后提出了我对需求分析的某些看法，强调了与客户沟通的重要性。

【正文】

近年，我一直从事某企业中有关 IT 项目的开发，有一个系统是用于计算机辅助电路设计的，包括了从上流设计（注：日式说法）到下流设计（注：日式说法）的所有流程，如用于可设计百万门数量级的逻辑门电路。有关方面把电路中路径的提取、过滤以及表示的某软件开发任务交给我公司，我有幸担任了该部分的需求分析以及设计。

我所设计部分为一单独可启动的软件，主要是解析文件中的连线路径，以列表视图和用直方图等把它们显示出来，还可以执行诸如查找与过滤等功能。

委托方对此提供了很初步的需求说明，把一些基本功能及性能要求描述了一下。我在需求分析时的工作主要有两点：第一，对该软件的界面等详细需求要自己重新进行分析提取。第二，对于已提供的功能要求需要深化和细化，以形成真正完整的需求分析文档。

在接到需求分析任务后，我分析了一下所要完成的工作。发现由于是专用领域的软件，对专业领域要求相当高，所以准备把此项目分成两部分：

（1）界面所受专业领域影响几乎没有，但由于全部没有任何要求，反而会感到风险和改动可能是最大的。

（2）功能方面由于委托方的许多功能都可以调用相应模块来得到，并且已有了相应的书面的简单需求，相应来说只是完成深化。对界面，我采用了部分 RUP 的思想迭代与渐进。而对功能需求采取了分层细化，每细化一层就要求委托方确认、修改和补充。

首先把风险较大的部分完成，这是现代软件开发的基本常识。我选择先进行界面的需求分析。第一步是根据功能描述抽取出逻辑模型，并使逻辑模型与界面元素及功能一一对应，大体上决定了界面应有的功能，然后根据该界面功能描述，确定具体的控件，这时，我参考了委托方已初步完成的主窗口的界面布局及控件的使用规律，然后根据需要完成的功能从 Qt（由于要支持 Windows 和 Unix 双平台，所以控件库采用 Qt）的类库中选择相应的控件。在提取和抽象逻辑模型时，我采用了 Rose 2000 中的用例图，即以 USE-CASE 图来描述与外部的关系。之所以采用 Rose，我是基于以下的原因：第一，在已开发的部分中，委托方统一要求我们使用 Rose 进行类和顺序图等的设计和代码生成。第二，Rose 提供了标准的图来描述系统与外部的关系，在全球范围已是一种标准结构。第三，使用上的方便性。我用 Rose 的 USE-CASE 图，理清了我们的软件窗口（日式说法，即：沟通人员）与委托方主窗口（日式说法，同上）以及外部角色（操作者）之间的相互关系。

在确定了界面元素后，考虑到文档的可理解性不是很强，我采用 Visio 2000 把界面的外观绘制出来，写上了基本的控件作用，随后送给委托方评审，幸运的是除了几个小功能的修改，委托方基本批准了我的方案。

下面的工作是为控件的行为及状态变化制定相应的状态迁移图，我选用的工具仍是 Rose，我用了状态图和时序图，把重要的控件状态变化及相应顺序进行了描述，随后的几天把相应的 DOC 文档建好写明，基本上界面设计就完成了。

下面的需求是针对功能需求的。虽然委托方技术部门有初步的需求文档，但由于领域的专门化不对，我不清楚其中复杂的路径提取关系及较深入的专业术语，一直有一种举步维艰的感觉。只能采用分层细化的原则，从最初几条深入一层变成十几条。这样的话，不会一下子碰到太深的专业问题，可以循序渐进从委托方与文献的解答中不断学习，深化自己对专业领域的了解，这样在设计中自己始终是层层推进的，不至于碰到无法逾越的专业障碍。

在这一阶段的开发中，由于一直是与自己不熟悉的专业领域打交道，所以我觉得一些辅助设计工具似乎无法发挥应有的功能。在这期间，对我帮助最大的应是公司的 E-Mail 系统，所有不清楚的问题的提出，以及对问题的解答都通过它进行周转。换句话说，在需求分析阶段，它起到了一个与客户的交流沟通和客户需求的提取作用。所以，我认为在这一阶段，E-Mail 系统是对我帮助最大的工具，其次是 Excel，我用它建立了问题跟踪图表，对每一个提出的问题，均需要记录上去，把问题结果（可分为已清楚、仍不太清楚、不清楚、尚未回答）均记录下来，根据这些表，我可以很好地了解自己工作中的核心问题，并有了解决它的方向，提高了工作效率。

每进行一层的细化，我都把结果交付委托方审核，由他们进行提出何时能终止细化，大约在八层细化后，对方认为已达到了效果，确认可以结束。至此，分析工作全部完成，项目的需求分析基本成功了。

在这次需求分析中，我认为取得成功的原因主要是方法和工具选择得正确。在界面设计中采用了流行的辅助工具，对需求及逻辑模型的建立提供很大的帮助，可以更方便帮助自己理清思路。选用了迭代法，把一些错误的影响在功能分析和界面分析的不断迭代过程中加以改正。在后期，以功能需求为主时，我主要依赖的是沟通工具和表格工具，这也说明辅助工具不是万能的，需求分析的关键之关键，应是与客户的交流与沟通。

通过这次案例，我认为在软件的需求分析工作中，方法的重要性应远超过工具的使用，应当首先确定分析中的风险，把风险分类，用不同的方法去解决各类风险，而工具的选择不仅是要看影响力和名气，而是要真正为我所用，应把握其精髓，即是此工具到底可以对开发有什么帮助，而不是仅限于如何使用。我认为在需求分析中工具的作用不外乎两个：一是实际系统与环境模型等的抽象工具，二是需求表达工具。第一类的代表是 Rose，第二类的代表是 Word、WPS、Visio 等，在这次项目中由于地理上的限制还用到了沟通工具，Web 浏览与 E-Mail 服务系统。

最后我还是总结一下，在需求分析中工具方法都只是辅助项目成功的因素，真正的决定因素还是一“与客户的沟通”。

企业应用集成的实践

摘要

为了向铁路各部门用户提供高可用、整合的信息，受铁道部科技司委托，我单位承担了“信息应用集成的研究”这一项目的研究和开发工作。本人作为该项目的负责人之一，担任了方案设计师的职务。该项目的目标是实现现有系统中的数据共享，有机的结合相关联的数据，搭建统一的使用平台，为今后铁路信息化更大规模应用集成建立可行、可靠的依据。本文主要从以下三个方面描述作者在该项目中的工作：确立企业应用集成的解决方案；选择应用集成规范和平台；在现有的铁路信息系统中选择二至三个，对应用集成的方案进行实践，分析并改进方案。

正文

近年来，铁路信息化建设取得了突飞猛进的发展，特别是正在建设实施中的“铁路运输管理信息系统（以下简称 TMIS）”工程。TMIS 是实现铁路运输管理现代化的一项重要工程，覆盖了铁路运输组织管理的各个环节。从功能上来看，TMIS 以货运管理为核心，分为列车预报系统、货票系统、车号自动识别系统、货运营销和技术计划系统等多个子系统。从组织结构上看，TMIS 由铁道部、14 个铁路局、50 个铁路分局和 2000 多个车站四级组成。目前，TMIS 已经成为一个由多个子系统构成的、庞大的多级分布式应用系统。

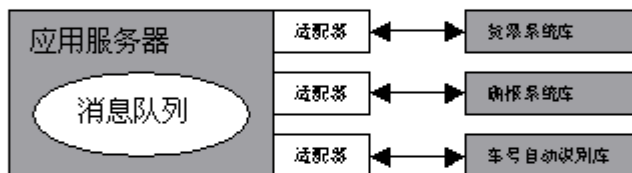
在部、局、分局的数据库中，已经建立了能够实时接收数据的货票库、预报库和车号自动识别库。货票库主要用于货物运费清算、预报库主要用于实时掌握调度行车信息、车号自动识别库主要用于车辆位置查询管理。为了实现列车、机车、车辆的实时追踪管理，必须共享各子系统数据；同时为了实现各级子系统的应用集成，必须选择一个开放式的应用集成规范和平台。为此受铁道部科技司委托，我单位承担了“信息应用集成的研究”这一项目的研究和开发工作。由于在 TMIS 工程建设中的多年工作经验，特别是在货票子系统中担任系统设计师的工作经验，本人有幸成为“信息应用集成的研究”项目的负责人之一，担任了项目方案设计师的职务。

经过分析，我对该项目的目标总结如下：为实现现有各系统中的数据共享，有机的结合相关联的数据，应确立企业应用集成的解决方案；为实现各级子系统的集成，建立统一的应用平台，应选择一个开放式的应用集成规范和平台；为建立今后铁路信息化大规模企业应用集成可行可靠的依据，应在现有的铁路信息系统中选择二至三个，对应用集成的方案进行实践。

下面我就从三个方面对企业应用集成的实践进行描述。

一、确立企业应用集成的解决方案

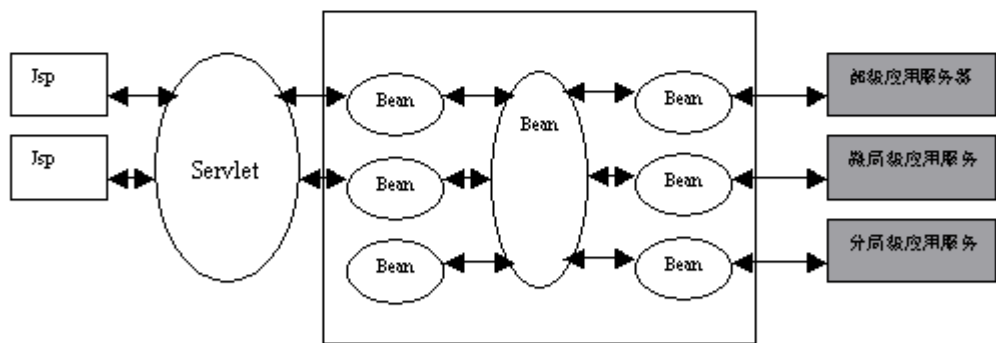
从集成的深度上来说，本方案包含了数据的集成，同时也包含了应用系统的集成。企业应用集成常用的方法有：基于客户端/服务器的方法、基于消息代理的方法、基于应用服务器的方法等。在这里我选择了基于应用服务器的方案，即通过建立独立的底层架构来连接企业的异构系统、应用数据等。如图所示：不同的子系统通过不同的适配器连接到应用集成服务器，应用集成服务器内部通过消息传递实现不同应用之间的交流。



二、选择应用集成规范和平台

从集成的广度上来说，本项目涉及部门之间的系统集成，同时也涉及了企业不同级（部、局、分局、站段）的系统集成。为了保证各级系统的数据一致，加强各级之间的交流，应该从整体来考虑企业的集成方案。较好的方法是选择一个开放的集成平台，把不同级的应用纵向的结合起来，通过统一的门户信息网站向外发布信息。

目前，开放式企业应用集成规范和平台的主流技术有两种：微软公司的 COM+规范和 Windows DNA 平台；SUN 公司的 EJB 规范和 J2EE 平台。由于 J2EE 架构的与操作系统无关的特性，为了更好的支持现有的不同级的系统，我选择了 EJB 和 J2EE 的组合。如图所示：系统采用 MVC 的设计模式。Jsp 所在为表示层，提供数据的展示功能；Servlet 所在为控制层，处理表示层与业务层的关系；业务层由多种 Bean 组成，Session Bean 处理交互相关的信息，Transaction Bean 处理具体的业务逻辑；Entity Bean 与数据层交互。



三、应用集成方案的实践

为了实现列车、机车、车辆、货物的实时追踪管理，需要在现有的几个系统中提供近乎实时的系统集成和数据集成。这些系统包括：货票系统提供了货物的名称、重量、装货车辆、发送人、收货人等信息；车号自动识别系统提供了车辆的位置等信息；预报系统提供了列车的车次、机车号、车辆号等信息。应用服务器通过编写各自适用的适配器连接这些系统。

应用服务器上有一个匹配作业，当车号自动识别数据入库时启动的该作业，通过车号和通过时间，在预报库中匹配该车所在的列车、机车等信息，然后在货票库中匹配货物的名称、收发人等信息，匹配后的信息写入动态追踪库，至此一次作业完成。同一时间段内，应用服务器上有可能有 n 个等待匹配的作业，依次放在消息队列中，通过编码实现的应用集成程序统一调度，消息队列的实现选用 IBM 的 MQ series。

企业不同级（部、局、分局）的系统之间的数据更新不同步，造成各级之间的矛盾，通过数据同步服务器的加入，实现三级数据的有机整合，保证数据的一致性。

数据同步后写入中央数据库，由三台集群的 web 服务器向外发布信息。同时满足铁道部、路局、分局用户的使用，包括货票信息应用、预报信息应用、车号自动识别信息应用和动态追踪综合应用。编码工作量相对较少，系统响应时间快。

企业应用集成后，实际的效果证明本项目的系统方案设计到达了项目目标的要求，横向能够集成不同系统的相关数据，纵向能够提供统一的应用平台。并且扩展方便，通过编写新的适配器，可以集成新的应用系统；通过广域网和局域网的建设，能够加入更多的用户。同时，集成后的新增的动态追踪系统也成为越来越多用户的热点应用。

应用系统集成后，信息通过相对统一的界面向外发布。由于用户的部门和级别不同，因此主要关心信息的角度各不相同，所关心信息的内容也有所不同，用户可能需要多次选择或点击才能得到想要的信息。引入 XML 技术，实现个性化的展示层，方便用户的使用，是我在该项目中的下一个工作目标。

论分布式数据库的设计与实现

摘要

本文通过 xxx 高速公路收费系统（以下简称收费系统），来论述分布式数据库的设计与实现。收费系统是我公司近年来接的较为大型的项目，管理结构为三层结构：公司级、收费中心级、收费站级，各级之间即可独立的完成自身业务，又有自上而下的管理关系。收费中心、收费站均为三层 C/S 结构，公司级采取 B/S 结构。该系统的数据库也按照三层来设计，收费站存放本站的所有流水数据，收费中心存放所有数据，公司本部存放查询用汇总数据，收费站与收费中心使用事务复制来同步数据，而收费中心与公司本部使用快照复制来同步数据，并且使用分级的方法来测试收费站、收费中心与公司本部之间的数据同步。

在本项目的开发过程中，我担任了数据库的设计工作。

正文

2000 年 10 月—2001 年 12 月我公司开发了高速公路收费系统（以下简称收费系统），收费系统项目从管理层面分为三层结构：公司级、收费中心级和收费站级。

公司本部：供领导、运营部、财务部等业务部门了解业务情况、检查工作。B/S 结构各部门通过 Web 服务器查询数据库服务器，而公司数据库服务器定时要求中心数据库服务器复制汇总数据。

收费中心：收费系统的管理中心下达管理制度，管理数据到收费站，接收统计收费站的收费数据，上报汇总数据到公司，负责日常的管理工作。

收费站：具体进行收费的单位，收费车道的数据通过通信系统实时上传到收费站数据库保存、分类、汇总，并且实时传送收费中心下达的数据库管理，并通过通信子系统下载到车道收费机上具体实施。

系统采用三层 C/S 与 B/S 的混合结构，收费中心与收费站为三层 C/S 结构，而公司级为 B/S 结构。我在项目中担任了数据库的设计工作，负责数据库的设计、测试及实施。

数据库设计

此收费系统的结构较为复杂，分为公司级、收费中心、收费站三级管理结构，既可独立工作，又有管理的联系。数据实时传送到收费站数据库服务器，再实时传送到收费中心数据库服务器。在数据库设计方面我们按物理的分布也分为三层结构。

在收费站，根据系统的需求分析的结果，一辆车通过收费站时产生的最基本的数据有：通过日期、时间、车型、收费类型及收费金额，因为收费标准不轻易改变，考虑到我们采用的是专用的车道收费机，存储量较小，所以收费金额项在此处不计入数据库，上传至收费站数据库服务器后，可以用车数乘以此类车型的收费标准而得到。当车道收费机上传数据到数据库时，还要加上工班、车道及收费员信息，保证数据的唯一性，所以我们把日期、时间、工班、车道、收费员、收费类型、车型设为组合主键，为车辆流水数据。收费员下班后还要上缴实收金额，因此还要保存实收金额，包括日期、工班、收费员及实际收费金额，为工班收费数据。在此基础上，分析、汇总数据，得到以下几类数据：

业务类型数据：车辆流水数据、工班收费数据、车道开通情况、收费员上班情况；

扩展的数据：为了查询、打印的方便、高效，流水表经过分类汇总，产生了以车道分类统计的车流量表（日、月）、以收费员来分类统计的收费金额表（日、月）及不收费车辆统计表；

管理类型数据：收费标准、收费员信息、收费站信息、车道信息、工班信息、收费类型信息；

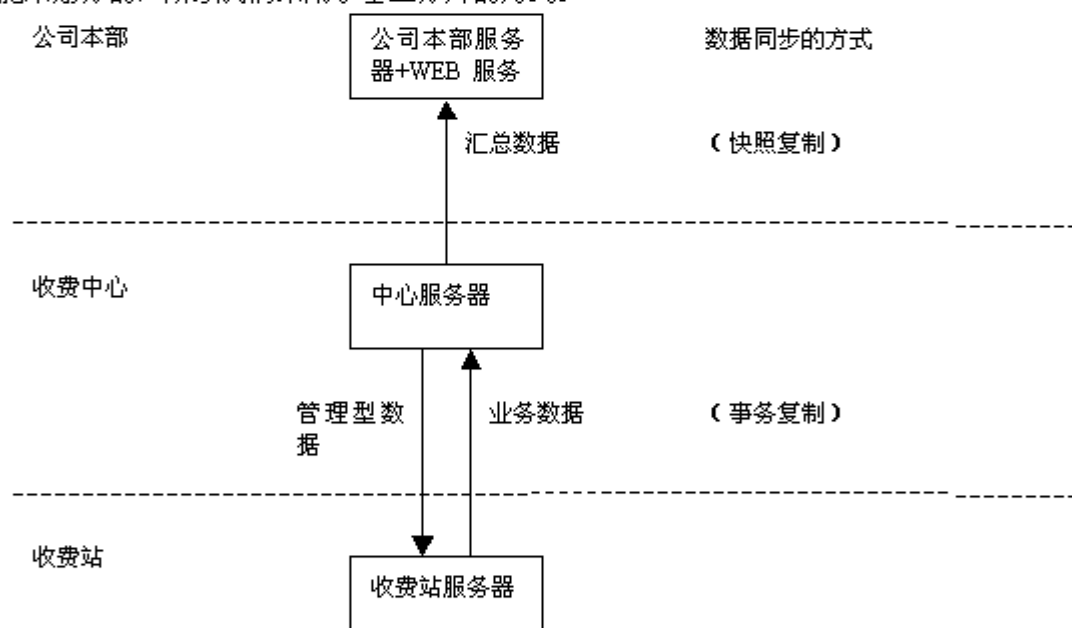
从全局应用的角度出发，各收费站存放本站的数据，收费中心的数据库则存放所有数据，并对数据进行完整性和一致性的检查，这种做法虽然有一定的数据冗余，但在不同场地存储同一数据的多个副本，能提高系统的可靠性、可用性，使系统易于扩充，也提高了局部应用的效率，减少了通讯代价，同时也使得各处理机之间的相互干扰降到最低。

公司总部存放所有收费站的汇总数据，通过浏览器进行查询。

数据的分布

（1）在收费中心数据库服务器与收费站数据库服务器的数据关系中，由于收费站的数据是收费中心数据的子集，我们采用了水平分片的方式，通过并运算实现关系的重构。

（2）在收费中心数据库服务器与公司总部数据库服务器的数据关系中，数据是按照其应用功能来划分的，所以我们采用了垂直分片的方式。



数据分布在多个地点，为了保证数据的一致性、完整性，我使用了事务复制和快照复制两种数据同步的方式，在收费中心与收费站之间使用事务复制，而在收费中心和公司总部之间使用快照复制。

对于业务类型的数据，收费站在本地存放收费车辆的实时数据，而用户也要求收费中心也要实时的收费车辆数据，延迟不超过 2 秒，所以我采用事务复制进行业务数据的同步，收费站只需将更新的数据发送到收费中心的数据库即可。具体过程如下，把收费站的数据库作为出版者和分发者，收费中心的数据库作为订阅者，对收费站的数据建立快照代理，并在分发数据库中记录同步状态的信息。每一个使用事务复制的收费站数据库均有自己的日志读取代理，运行在分发者上并连接出版者。分发代理的任务是将分发数据库中保持的事务任务直接推动到订阅者。当推订阅被创建时，每个为立即同步而建立的事务出版物通过自己的分布代理运行在分发者上并与订阅者相连。

而管理型的数据是在收费中心设置，虽然修改的频度不是很大，但在修改后即要发生作用，所以也采用事物型复制，收费中心为出版者，收费站为订阅者。将管理型的数据发布到各收费站。

由于公司总部不需要实时更新，所以收费中心数据库服务器与公司总部服务器之间的数据同步设置为快照复制，公司总部数据库中建立收费中心表的快照，对这些数据的修改在收费中心进行，把收费中心数据库服务器设置为出版者，出版物为汇总数据，公司总部数据库服务器设置为订阅者，快照代理将准备包含有被出版数据表的结构与数据的快照文件，在分发者上存储这些文件，并在分发者的分发数据库中记录同步任务。在收费中心数据库服务器上建立一个作业，设置为每天早上 8 点更新一次（收费日的天为 8 点到第二天 8 点），公司总部就能得到截止到前一天所有收费站的汇总数据。

测试

数据库设计好了，就要对它进行测试。我们的测试策略为分步测试：首先测试收费站数据的正确性及完整性，在多台收费机上同时输入几组车型、收费类型的数据，查询数据库的流水数据是否正确，再看汇总数据是否正确；收费站正确后，再测试收费中心的数据的正确性、完整性及延时；再测试公司总部的数据正确性、完整性。

一开始我们把所有的表都建立在一个出版物中，但在测试中我们发现，由于收费站一个表的错误而造成的复制的中断，经常要重新配置复制，所有的表都要重新选择及设置一遍，非常繁琐，我们采取一个表建立一个出版物的方法来简化操作，只要恢复出错的复制，其他的复制仍然能正常执行，而且哪一个表的复制发生中断也很明确，不仅简化了操作，也加快了处理时间，就是日后用户维护起来也简单明了。

在设计过程中，基于查询及安全性的需要，我们大量的使用了视图，第一加快了查询速度；第二也防止了人为因素造成的数据的更改。

总结

按照以上的设计方案实施后，完全满足高速公路系统对数据实时性和完整性的要求，系统目前只在内部使用，而以后全省的高速公路收费系统实行联网，在外网上发布信息，那时数据的安全性及查询的响应速度将是我们考虑的重点。

论分布式数据库的设计和实现

摘要

本文论述《金蚕工程》的分布式数据库的设计和实现。该项目的设计目标是实现企业间茧、丝等的合同交易（交易规则和期货交易一样）、实时行情和成交数据的发布、茧丝质押和质押数据的发布。所有功能均要求既能在企业局域网交易大厅和 Internet 上进行，许多功能又要在苏州和成都分中心进行。系统从设计时就基于分布式数据库应用的可用性和可靠性作为系统一个关键目标。为了达到系统的上述要求，系统分别从数据库设计、应用数据集成和测试以及分布式数据库部署等做了大量工作。针对上述各部分，本文论述了分布式数据库的设计和实现及遇到过的典型问题和措施，最后对系统改进，谈一些自己的体会。

在项目开展期间，我担任了系统分析、系统设计与关键模块的编程等大量工作。该项目在 2002 年通过了浙江省软件评测中心的评测。

正文

我于 2000 年底到 2002 年 6 月组织了中国茧丝绸市场由国家经贸部资助的《金蚕工程》一、二期项目的开发。由于原有系统存在如下问题：（一）原有系统采用传统 C/S 结构，客户端运行在 DOS 平台，后台使用 Foxpro 数据库。系统处理慢，前后台数据一致性，并且 Foxpro 服务器在交易高峰经常莫名其妙的死锁和 Down 机。（二）市场是中国最大的茧丝交易市场，由于会员单位不断增加，原有远程电话拨入设备严重不够用，系统又不能通过 Internet 连接访问，进行合同交易和行情的及时查询，严重限制了市场的发展。（三）原有系统只部分实现了合同交易和资金清算，功能和企业发展不相适应。（四）由于会员在地区上分布不均匀，为了更好地服务于会员要求在满足一定条件下建立分中心交易系统。在这种情况下，市场决定开发基于 Internet B/S 结构和基于 Xnetserver 通讯中间件（类似 IBM MQSeries 的通讯中间件）的 4-Tier 结构的综合软件，分中心采用基于分布式数据库地应用系统。一期项目包括：基于 4-Tier 的合同交易系统和资金清算系统，基于 Internet 的行情查询和信息发布的企业网站。二期项目包括：茧丝质押、仓库管理系统、Internet 质押查询系统和分中心交易和资金结算系统。

鉴于该项目业务比较复杂，流程比较多，系统要兼顾企业交易大厅会员、远程 Internet 交易的会员和分交易中心会员，项目完成时间短等特点，为了既要项目的按期投产又要实现基于分布式数据库的分中心交易和资金结算系统，我从数据库选型、数据库设计、应用数据集成和测试以及分布式数据库部署如下工作。

（1）数据库选型。现在的主流数据库一般都可以按分布式进行部署，例如：SQL Server、Oracle、Informix 和 DB2 等。由于一期项目的数据库平台是选择了 Oracle 7.0，经过了近一年的运行，其数据自动备份、多用户并发处理性能、磁盘表空间管理、Web 数据发布等表现良好。我公司又由于开发过基于该平台的分布式应用系统，公司对其分布式实施也有相当的经验，所以经我们开发部相关人员的讨论，我决定在各分交易中心继续采用 Oracle 数据库来实现，其分布功能主要基于 SQL*NET 来完成。

（2）分布式数据库设计。由于各交易品种的每节交易价格由市场的交易大厅决定，所以各分中心必须在每节交易开始前取得相关的交易数据。各分交易中心必须保留其会员已经交易成功的买入和卖出交易数据，并根据会员需要进行资金结算。经对分交易中心充分业务需求分析后，我觉得分交易中心的主要功能还是和主市场交易中心（简称：主中心）功能是一致的，只不过分交易中心的会员数比较少和无需保留各节的历史交易行情，所以，分中心的主要数据库中的各表还是照搬主市场数据库的各表，分中心只保留各自的会员基础数据和成交数据，保留会员历史交易数据，只保留当前的各节的交易行情，不保留历史行情。主中心必须保留各分中心的各项明细数据，主中心也必须及时计算分中心会员的汇总资金，所以主中心必须增加分中心汇总资金表。

（3）数据集成。为了保证分中心及时下单，必须保证每节的交易行情要及时的传送到各交易分中心。节处理最后一步必须把下一节的交易行情传送给各交易分中心。同时下发给分中心的数据还有，分中心本节各会员的成交数据。由于主中心保留各分中心的各项数据，所以，分中心有新的会员加入必须同时更新分中心和主中心数据库；分中心会员资金有变动时也必须同时更新主数据库。由于各会员只能在当日交易结束后才能进行数据平仓，所以分中心平仓数据汇总后同意发送到主中心进行平仓。在上述设计的数据库到数据库操作的数据完整性和一致性由数据库的二阶段提交（2 Phase Commit）来保证。由于 Oracle 是一个比较成熟的数据库平台，又由于每次事务都不大，所以运行到现在也没有发现数据不一致现象，没有出现“单边帐”，数据处理也比较快。为了进一步保障各中心数据的一致性，我还做了一个同步程序，由主中心发起，对主中心和分中心数据进行比较。该程序可定期进行。

（4）测试。由于一期项目已经运行了相当长的一段时间，系统已经相当稳定了。由鉴于 2 Phase Commit 出色表现，所以基于分布式的测试没有觉得和集中数据库有太大的区别。唯一的感受是在第一次建立远程数据库连接时，连接速度比较慢（有时超过 20 秒）。为了提高测试速度，我在同一个局域网中安装了三套 Oracle，分别充当主中心和两个分中心进行测试。

（5）部署分布式数据库。和集中时式的数据库不同，采用分布式系统，必须配置 SQL*NET。其他的 Oracle 组件和子系统安全同本地安装数据库一样。

分布式数据库在逻辑上属于同一系统，使得应用不必关心远程数据库的物理位置，就可以像访问本地数据库一样访问远程数据库。采用分布式数据库可充分提高系统的处理能力、均衡网络负载，所以该分布式数据库应用方案在银行代收、代付业务处理、保险代理点等单位 and 机构中大量采用。但由于近些年来通信基础设施建设比较迅速，特别是 2M、10M 和 100M 光纤的推广（100M 厂电月租费仅为 800 元），又为了节省下属网点的机房费用，各单位又不约而同的采用的集中方案。

【缺点和需要改进的地方】

（1）分中心或主中心由于不明原因而 Down 机后，由于 Oracle 有自动恢复功能，可能造成对方数据库会收到过时的垃圾数据。该问题现在通过手工方式解决。

（2）由于分中心没有收到下一节的交易行情，可能会造成分中心会延时交易。分中心的主控系统可通过主动拉取数据来解决。

【结束语】

《金蚕工程》项目是成功的，年交易量已经有 100 多亿人民币。通过该项目的顺利投入运行，中国茧丝绸市场不仅牢牢把握住了中国的茧丝行情，随着我国进入 WTO，中国茧丝绸市场在世界上也有举足轻重的分量！

总之，中国的软件产业应该走自己的路，只有通过“已过程为核心，以度量为基础，已人为本”的管理政策；根据我国的国情、软件业和公司的现状，只有通过“干”，才能有实绩，才能实现“以信息化带动工业化的大思路”！

论分布式数据库的设计与实现

摘要

分布式数据库系统把应用所需的数据存放在多个数据库服务器上，完成某个数据操作要涉及到访问多个服务器，这适用于某种特定需要的应用。我在主持设计开发的一个 MIS 系统中，为了达到了在低速网络通道下有效提高应用程序性能的目的，使用了 Sybase 的分布式数据库技术。我设计的这个系统是采用典型的 C/S 结构，但许多客户端连接服务器的网络采用电话线拨号，速度有限，传统 Windows 界面的客户端应用程序相应速度比较慢。考虑到 B/S 结构也避免不了大量数据从服务器端传输到客户端，我认为 WEB 界面并不能有效解决这个问题，所以采用了优化数据库结构的方法，把数据分两部分存放，基础数据放客户机，会员资料主要采用键码放服务器，应用程序再现数据时从服务器取键码，到客户机取对应的解释，由于键码的数据量少，网络传输便快。在构建这个分布式数据库系统的过程中，我着重研究并解决了数据同步和事务协调的问题，取得了良好的应用效果。我认为，分布式数据库系统的技术在 Internet 时代正当其道，大有发展前景。

正文

分布式数据库系统把数据存放在多个数据库服务器上，当应用提取所需数据时，要访问多个服务器，综合多点数据才能完成。

分布式数据库技术在很多场合得到了应用。譬如某企业随着业务量的扩大，原有数据库服务器已经达到了容量和性能极限，如果不希望丢弃原有投资，可以建立另外一套新的数据库，跟原有的系统组成一个分布式数据库系统，给应用提供透明统一的数据访问。还有，如果某企业分成多个业务部门，而且地域分散，可以在某个部门放置单独的数据库服务器，用于存放该部门最常用的数据，而部门和部门之间相互引用的数据可以通过分布式数据库技术来方便地完成。分布式数据库不是简单地把集中数据库分散实现，而是针对某种特定应用需要而诞生，它必然具有自己特有的性质和特征，需要上面做许多的工作，来满足应用的要求。

我在设计、开发一个 MIS 系统时，针对应用的需要而引入分布式数据库技术，取得了良好的效果。

该系统针对会员资料的管理而设计，用于管理会员入会、缴纳会费、申请资助、办理资助审批、关系转移、退会和注销手续等业务流程。分三个级别的应用权限——基层单位级、总公司级和集团公司级，各个级别只能操作各自范围内的业务数据。该系统采用典型的 C/S 结构，后台数据库采用 Sybase，前端应用采用 PB 开发工具来设计标准的 Windows 操作界面。我在其中任系统分析和数据库设计的角色，担任了调查业务需求、业务建模和数据库建模、数据库设计以及指导应用程序测试、优化系统和应用的性能等一系列工作。

由于客户端地域的分散，遍及多个省境内，许多使用该系统的基层单位连接服务器数据库的网络采用电话线拨号方式，速度有限，在使用客户端应用程序时感觉界面速度很慢。经过分析，认识到许多操作都要从服务器中取数据，速度慢就慢在数据访问上。服务器是没有性能瓶颈的，问题出在网络速度上。不可能要求众多使用客户改善和升级他们的网络，只能充分挖掘软件的潜力，来适应这种低速网络的使用模式。

经探讨，结合关系数据库的知识，认识到，应用程序的每一次数据库操作，都要访问多个相关联的表，其中，有会员资料表和基础数据表，会员资料表中存放许多的键码值，在基础数据表中有键码相应的解释。键码值的数据量比较少，而基础数据是静态的，几乎不会更改。如果考虑把会员资料放在服务器上，基础数据放在客户端，当应用程序中访问数据时，总是从服务器上存取会员资料，从客户端提取会员资料中键码的相应解释。由于键码的数据量少，便减少了网络上传递的数据量，从而提高了界面的响应速度。

同时考虑到基层单位总是操作自己所属的部分会员，增删转移操作少，会员列表比较固定，而每一项业务操作都涉及到要从会员列表中查找定位到某个会员，所以会员列表是最常访问的数据项。把会员列表从会员资料数据中抽取出来，也放置在客户端，这样，便进一步改善了性能。

把数据分散存放只是工作的第一步，接下来要考虑应用程序怎样访问这种分布式数据。开发应用时，如果每一功能都针对两个数据库进行，就带来了很多麻烦。所以，我们研究了 Sybase 的分布式数据库技术，决定采用了 CIS（组件集成服务）部件，来合并两个数据库成一个统一的分布式数据库。应用程序只要连接一个数据库，就可以透明统一访问到两个数据库中的数据。

该技术具体实施方法是，在客户端数据库中建立一个对服务器数据库的远程访问服务名，包含访问地址、登录用户名、登录密码等等关键的连接信息；并且对服务器中会员资料数据表建立一个本地代理表，结构和服务器中远程表完全一样，它是访问服务器中会员资料的中转和代理。客户端应用程序访问本地代理会员资料表时，实际上是通过预定义的远程访问服务名中包含的连接信息到服务器中对应的实际会员资料表中访问数据。这种访问对于客户端完全透明，感觉不到是从物理上独立的两个服务器中存取数据。所以，这种数据库结构是典型的分布式数据库。

部署这种分布式数据库不是难事，只要在客户端和服务服务器上安装 12.0 版以上的数据库服务器，在客户端服务器上建立远程服务名和代理表即可。由于 Sybase 数据库的安装支持脚本方式，在客户端应用程序的标准安装过程中，嵌入 Sybase 数据库的安装和配置脚本，就自动化地完成了所有工作。

在实际使用该分布式数据库系统的过程中，遇到了几个问题。第一，数据同步。客户端基础数据不是绝对静态的，也有变化，因此在服务器端要设置一个统一的基准，称为主点数据。客户端总是复制使用，称为复制点数据。如何及时感知到服务器端主点数据的变化，有效率地复制到客户端，是个难题。Sybase 针对这种应用场合，提供了复制服务器技术，但为了避免过于复杂，我们实际采用应用程序来管理同步。当服务器端主点数据有了更改时，保存一个相应的标识和时间戳，客户端应用在登录服务器时，检查这种标识，一检测到了数据有更新，就首先下载，然后再进入系统正常使用。这种方法实现起来，增加了额外的开发量，且不能判别绕过应用程序对数据的直接修改，但是，是最简单和有效的方法。

第二个问题是事务协调问题。物理上独立的两个数据库，在协同操作时，如果服务器正好停机或者网络故障，完整的一个事务没能完成，就会“事务崩溃”。虽然 Sybase CIS 内嵌了两阶段提交技术，能够自动恢复。但是应用程序在这种情况下，敏感性不够，操作界面会无端凝固，影响了使用的方便性。我们针对 PB 对于连接的判断和感知，用了一个小小编程技巧，使应用程序能够及时感知到数据库连接故障，及时停止和恢复事务，使操作界面表现友好灵活。

以上遇到的这些问题，都找到了解决办法。分布式数据库技术的应用并不是非常复杂，它往往为解决特定问题、满足特定需要而被采纳，使用得当，会给应用带来了许多便捷。

在当今信息社会里，互联网络带来了相互连通的便捷，而且知识爆炸，数据的分布式访问是个必然趋势。潮流兴起的 XML 技术，提供了各种平台数据库之间的一个公共数据访问标准，可能会用来构建更加灵活、适应性更强的分布式数据库技术。

论信息系统的架构设计

摘要

本文讨论医保通零距离实时赔付系统项目的架构设计。该系统主要实现了中国人寿保险公司通过与医院合作，让中国人寿客户在出险住院并完成治疗后，即可获得实时的健康险理赔服务，从而在提升保险公司服务的同时减轻病人经济负担、减少客户理赔困难。

在医保通实时赔付系统设计架构中，整个系统中分为 B/S 结构的管理中心端与 C/S 结构的医疗机构前端两部份。在管理中心端采用 J2EE 架构，使用了与传统 EJB 为核心的重量级架构有所不同的轻量级架构方式，其中主要使用 Spring 框架作为系统的基础平台，充分体现了 Spring 的高开发效率、易测试维护性及应用服务的可移植性等优点。同时，在架构设计中，充分考虑了系统的可扩展性、稳定性、安全性、可维护性、灵活性等因素。

在本项目的开发过程中，我担任了系统架构设计与项目管理的工作。该项目从目前推广与应用情况看，达到了项目的预期目标，得到了各级公司的一致好评。

正文

“理赔难”一直以来都是各大商业保险公司与参保客户所关注的重要问题，为解决该问题，中国人寿保险公司提出了商业保险的实时赔付系统——中国人寿医保通零距离实时赔付系统，该系统是国内首个用于商业健康险的实时理赔服务系统。系统主要通过中国人寿保险公司与医院合作，将常规保险理赔工作从事后理赔调整为事中理赔，也就是将以往的串行工作方式“客户出险—>住院—>出院—>理赔开始—>理赔结束”，改为了并行方式，即客户住院时也就开始了理赔工作，当客户出院时理赔工作同时结束，有效地缩短了理赔时间。医保通实时赔付系统的出现，提高了保险公司理赔服务时效、降低了保险公司调查难度，保证了理赔调查的准确性，加强了理赔风险管理，降低了赔付率；同时，为参保客户提供了方便的理赔服务方式，使客户在住院期间可少缴医疗保证金，减轻了客户住院的经济压力，让客户治疗出院时即可获得理赔，免去了客户理赔奔波等烦心事。

医保通系统第二版于 2006 年底开始全新开发，并对系统功能及系统架构进行了全面改进，在通过公司近 30 位员工的共同努力下，该版本于 2007 年 6 月通过客户验收。我在本项目的开发过程中，担任了系统架构设计与项目管理工作。

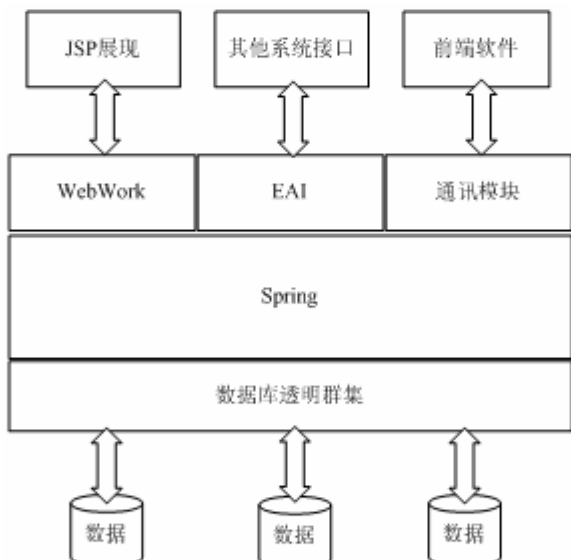
信息系统架构设计是软件需求分析与软件设计的桥梁，架构在软件开发中为不同人员提供了共同的交流语言，体现了系统早期的设计决策，为系统的开发提供了强有力的支持。好的架构将奠定优质的信息系统，同时好的架构需要源于对需求的充分了解。因此，在进行架构设计前，我们对软件需求进行了充分的了解与分析。

通过对需求的了解与分析，我们知道该项目主要干系人员是保险公司（包括管理人员与各岗位工作人员）、医院工作人员及客户，真正使用与操作的人员是保险公司与医院，他们分别有各自的职责，分别负责不同的工作内容。从系统功能上区分，保险公司主要负责各项管理工作及众多的统计分析；而医院则主要负责对客户住院信息与诊疗信息的采集及进行理赔金的垫付。系统的所有数据与信息都必须存储于保险公司。

在软件功能上，医院需要使用的功能相对简单，保险公司需要使用的功能则复杂很多。在软件的使用环境上，由于保险公司多年来一直较重视信息技术的发展，已建立有良好的网络及信息技术平台；而医院则不同，由于国内医院众多，其各自对信息技术的开发与利用各不相同，甚至某些小型的卫生院目前都没有安装使用计算机。保险公司领导与医院领导对软件的关注点也存在差异，保险公司领导主要关心软件能否实现预定的各项功能，能否达到预期的管理目标，而开发与运维费用则能少则少，成为了他们次要关注内容；对于医院领导来说，医保通系统的实施给其带来的显式经济收益不多，反而会增加其成本的投入，因而他们会考虑更多的成本投入。最终，我们总结出以下几点：

- 1、软件在保险公司与医院都需要有操作界面，医院在本地准备与组织数据，需要进行数据通讯时再进行网络联接，不需要实时在线操作，以减少网络使用费用的支出；
- 2、网络连接上需要提供多种连接方案，对于使用量小的医院，可选择拨号连接方式，对于使用量大的医院可使用 ADSL 或光纤进行 VPN 接入；
- 3、医院端需要提供多种与医院已有管理系统（HIS）的接口方式，便于医院重复利用已有信息数据，减少重复录入；
- 4、医院端必须提供简单的信息录入方式，包括入院信息、诊疗信息及出院信息，使没有 HIS 系统的医院能进行信息的录入工作；
- 5、保险公司端则需要在结合功能实现的基础上，更多的考虑系统的稳定性、安全性、可扩展性、灵活性、可维护性等方面；
- 6、医保通系统将保险公司核心系统及众多的 HIS 系统进行信息接口，所使用的信息编码应符合国际、国家及行业编码规范；
- 7、最大限度地降低系统开发及运维成本。

基于这些特点，我们将整个系统分为 B/S 结构的管理中心端与 C/S 结构的医疗机构前端两个部份。在管理中心端采用 J2EE 架构，其中主要使用 Spring 框架作为系统的基础平台。C/S 结构的医院端软件在整个系统的架构中被视为一个表现层，通过专用的通讯模块与管理端实现数据接口，而对于医院来说，他又是一套独立的应用软件。如下图所示：



从图中可以看出，本系统是一个典型的多层轻量级 J2EE 架构，主要分为表现层（管理端表现层与医院端表现层）、企业应用集成层、业务逻辑层、数据访问层。

在表现层，医院端通过 TCP/IP 方式和中心端进行数据交互，可看作系统在医院端的表现层，通讯模块是医院端与管理端数据交互的主要途径；而管理端表现层则以 JSP 及报表等多种方式将数据展示出来，供用户使用，管理端用户操作主要集中在这一层。Webwork2 作为轻量级的 MVC 框架，能够较为容易的与 Spring 进行集成；同时，优雅的设计提供了良好的扩展接口，可以有多种的数据表现形式。

企业应用集成层（EAI），所涉及的数据较多，提供从多方进行数据采集，如保险公司已有的三套核心系统及社保系统等。此层的作用即是为与其它企业应用提供数据交互的接口。

业务逻辑层，是系统的核心，它负责处理系统中的各类业务逻辑。它是以 Spring 作为基础平台，借助它的强大功能，以轻量级的 POJO 为基础，即可完成复杂的业务逻辑处理。

数据访问层，因 Spring 提供了良好的数据访问支持，如 JDBC 及事务支持等，使得数据访问较为容易。

该架构中各层相互间以松耦合的方式进行联系并负责各自的工作。各层在应用部署时可合并部署也可分开部署，在工作负荷较小时，所有层可合并部署在同一服务器中，如果某层工作负荷增大，则可将其独立出来部署于其他服务器中，或以服务群集的方式进行部署，以提高系统的整体运行效率。

总的来说，该架构充分体现 Spring 的高开发效率、易测试维护性及应用服务的可移植性等优点，同时也能体现出该系统的可扩展性与灵活性。从目前推广与应用情况看，达到了项目的预期目标，得到了各级公司的一致好评。但该架构仍存在其缺点，目前，保险公司已有三套信息相对独立的核心系统，虽然医保通系统通过 EAI 层很好地解决了与之分别相对的信息接口，但仍存在很多“信息孤岛”，该架构不利于解决信息系统的整合。

在架构设计之初，医院端与管理端通讯并未使用专用通讯模块，是将两个软件看作是相对独立的软件，其通讯则是根据不同交易使用不同的处理过程，这样将导致开发与运维工作量的增加，甚至影响到系统的稳定性与安全性，同时也意味着相应成本的增加。在进行架构设计论证时，有员工对此提出了新的看法。其方法是将通讯处理过程进行统一，形成独立的数据通讯层，并负责网络通讯的安全管理，所有通讯工作都需要经过该层，但他不对通讯的具体内容进行处理，也就是说不管通讯内容或格式有什么变化，其通讯翅都不需要进行修改。从而降低了开发工作量，减少了对通讯的运维，也提升了系统安全性、稳定性与可维护性。通过这事，我们再次体会了在软件开发中需要大家的共同努力，集中大家的智慧，团结才能发出更大的力量。

论新技术的引进

摘要

根据国家税务总局对税务系统内所有系统进行集成与整合的需求,我所在的开发单位组织了全国金税工程防伪税控系统网络版的升级开发工作。该项目工程浩大,要求在具有严格的安全、可靠性能的基础上,将基于 DOS 操作系统、Foxpro 数据库的原单机版防伪税控子系统集成到基于网络的、大型数据库的“集中存储、分布操作”的分布式系统中来,并实现与基于 AIX 等操作系统和 Oracle 数据库的稽核协查等其他应用系统的数据共享和互操作。在项目中,我担任项目主管,主要负责系统规划和组织实施工作。我在将近一年的可行性研究、需求分析、系统研发与试点工作中,通过引进面向对象设计方法、采用 B/S/S 三层体系结构、利用群集实现负载均衡等新技术,使该项目取得了圆满成功,受到了用户的一致好评。但是现在看来,由于新技术的使用,怎样实现软件开发公司对新技术的渗透、怎样开发自主知识产权的中间件等问题,需要我们在今后系统开发中做进一步探索。

正文

2003 年元月,我作为项目主管,有幸参与了全国税务系统金税工程防伪税控系统的升级开发工作。防伪税控系统主要由基于增值税发票的企业开票、企业发行、报税、认证、发票发售等五大子系统组成,系统组成模块如图 1 所示。具体流程是:企业在当地税务机关通过企业发行系统取得用于开具增值税发票的相关设备(金税卡与 IC 卡)和权限,再到发票发售系统领取增值税发票,通过企业开票系统开出发票后产生发票明细和申报纳税数据存入 IC 卡中,然后、企业持 IC 卡到税务机关由报税系统进行报税、并将取得的可抵扣的发票数据通过认证系统进行认证后,经过 Internet 上传到税务机关,所有数据经由税务机关处理后,传递到其它系统进行处理。系统要求具有严格的安全、可靠性能,必须建成基于网络的,分布式实时数据库处理系统,达到数据共享的目的。在此次开发过程中,我特别强调在认真细致的需求分析的基础上,最大限度地引进新技术新方法的思想。

一、采用面向对象分析方法

防伪税控系统网络的应用将极为广泛,涉及全国所有的税局和一般纳税人企业,具有基于增值税发票的开票、认证、报税、传递等业务操作,面对如此复杂的系统,怎样从中理出头绪来,最大限度满足用户要求,实现整个开发流程的“无缝”连接,需求分析是最重要的环节。我在认真研究和分析旧单机版开发文档后,提出采用面向对象的开发方法。系统的主要业务范围是增值税专用发票的防伪与税控,通过对整个系统流程的分析,我抽象出“发票、操作员、安全卡”等类,从类出发,建立对象,再通过对象类之间的继承、聚合关系、消息和关联,分别建立开票、认证、开票、企业发行等子系统和共用的系统管理等模块,并通过操作员类中权限属性实现各子系统的权限控制。通过面向对象分析方法,不仅提高了系统的开发效率,而且提高了软件的复用性和可维护性。

在工具的选择过程中,我们选择了现在十分流行的系统 Rational 系统系列工具,包括 Rational Rose、Rup、Soda 和 Requisite Pro 等。特别的,从防伪税控系统的对于增值税发票的防伪和税控功能可以看出,整个系统具有很长的生命周期,必须面对税务系统因为经济发展而出现的多变的需求变更,并且,又必须服务于税务系统其它的如征收管理、稽核协查等应用系统。所以,要求系统具有很好的可维护性、可扩展性。而公司在原有单机版的升级上,因为没有统一的、规范的开发文档而吃尽了苦头。所以,我决定采用 Requisite Pro 作为我们现在和未来的系统需求管理工具。我们在对经过详细的用户调查后,按照我们的基本理解写成了基本需求,交给用户进行评审和补充,再形成正式的需求录入到 Requisite Pro 中去,并记录需求的变化情况、需求之间的依赖关系,实现需求的全面管理。公司决策层在认真听我的汇报后,表示了很高的兴趣并给出支持,使我更加增强引入新技术进行开发的决心。

但是,新技术并不意味着是最好的或是最适合的。在引用新技术中,我特别注意它的负面。面向对象分析方法虽然有效地表达和描述了现实世界,但有时也会忽略外在的表层的需求,有些关键需求要等到用户使用后才会提出,然而等到用户使用后再维护是不现实的,作为原型开发模型中的原型也是收集用户需求,描述与解释需求的一类相当有效的方法工具。在分析过程中,为了更好让用户了解我们的系统,更完整提出需求,我沿用了公司熟用的原型开发方法。通过利用 Access 开发出系统原型,让用户试用,取得很好的效果。所以在新技术引进过程中,我们熟悉的优秀的开发思想与方法不能丢,这是我们要特别注意的地方。

二、采用 B/S/S 三层软件体系结构

瘦客户端设计表示层。进行系统设计时,采用客户/服务器(C/S)模式还是采用浏览器/应用服务器/数据库服务器的(B/S/S)模式,成为我们开发小组成员争论的焦点。我认为,采用哪种系统体系结构相当重要,决策时,不但要考虑系统运行成本,还要考虑系统运行的稳定性、可靠性、可维护性以及业务需求变动时的可扩展性。新系统将所有数据集中到市级税务机关(参看图 1),如果将业务处理逻辑分散到客户端或数据库服务器,虽然可以低成本运行,但是,我们分析得到,同一时刻,区县级税务机关和企业到服务器的连接可能达到 30-1000,在发达地区可能更多,这显然击中了 C/S 模式的要害,并且,税务系统的需求经常变更,所以,我们决定采用应用服务器集中处理实现业务逻辑。事实证明,这个决策是正确的。新系统在沿海五省四市试运行,不仅稳定、可靠,易维护,而且,客户端采用 IE 浏览器,受到了维护人员和广大操作者的好评。

业务层引入 WEBLOGIC 应用服务器。众所周知,WEBLOGIC 是用于开发、集成、部署和管理大型分布式 WEB 应用、网络应用和数据库应用的 Java 应用服务器。可以说,采用 Weblogic 的主要理由有:首先是税务现有应用系统的异构的数据库平台如用于(如用于 CTAIS 征管系统的 Sybase、稽核协查系统的 Oracle、其它的 SQL Server);再有,对多种操作系统的支持,包括 NT 系列、AIX、Sco 等;并且,税务的系统需求经常变化,比如在我們的开发过程中,因为国家税务总局推出“一窗式”管理,税局要求我们的报税系统有向征管系统传出某种格式数据的功能,我们利用 WEBLOGIC 对 XML 的支持,很快地解决了这个需求。总之,这要求系统具有很好的可移植性和可伸缩性,应用服务器能使程序员从常规设计中脱离出来,集中精力组织和优化业务逻辑。我特别注意了服务器的选型,主要从开发效率、可复用性、可伸缩性和可扩展性等几个方面考虑。

三、利用工作站实现负载均衡

企业将取得的可抵扣的增值税发票经过认证系统认证后,形成加密的认证数据,经过互联网传到税局的外网 WEB 服务器,再由 WEB 服务器传到内网的应用服务器进行解密和税局再认证。对于一个市级税局,这种发票数据可能非常庞大,如果由应用服务器信中进行解密,很显然,时间上是不允许的,并且,可能造成应用服务器的死机。我提出采用解密工作站队列处理解密工作,让应用服务器统一分配各工作站的工作量。这样,大大地减少了应用服务器的工作量。

综上所述,由于采用了面向对象的系统分析方法,B/S/S 系统体系结构、解密工作站实现负载均衡等新技术,提高了整个系统的开发效率和质量,使新系统按计划完成,且具有很好的安全性、可靠性、可维护性、可扩展性和外部接口。但是,由于新技术的采用,也使我们产生了一些新的问题。

怎样让员工理解、支持和掌握新技术。引进新技术,将打破开发人员习惯的开发方法和程序设计方式,这种习惯的打破,往往要付出更多的精力和时间,如果靠权力或压力来解决只能是一种表面现象,我认为,关键是要营造一种引进新技术进行软件开发的文化氛围,让软件公司开发人员学习并引进新技术到我们的开发中来形成一种习惯,怎样营造这种氛围,是我们管理领导层常思考的问题。

怎样开发自主知识产权的中间件。我们的应用服务器采用的是 BEA 公司的 WEBLOGIC 中间件,虽然,它提高了我们新系统的开发效率和安全性和平台无关性,但是,也大大提高了我们的软件开发成本。中间件在我国具有很大的市场和利润空间,我想基于中间件的开发,首先要把握它的市场价值,然后在开发上注重它的通用性。

论计算机网络的安全性设计

摘要

在计算机与网络技术飞速发展的今天，医院信息系统的建设已经成为医院现代化管理的重要标志，同时也是医院管理水平的一种体现。尤其是医疗保险制度的改革，与医院信息系统形成了相互促进的态势，我国很多医院都建立了自己的信息系统。由于行业性质的缘故，医院信息系统必须 7 X 24 小时不间断运转，因此对网络系统的安全性和可靠性有很高的要求。本文通过一个医院信息系统项目，阐述了医院计算机网络的安全性设计方面的一些具体措施，并就保障网络的安全性与提高网络服务效率之间的关系，谈了自己的一点体会。

正文

我于 2001 年 4 月至 2003 年 10 月参加了某医院的医院信息系统的建设工作，在项目中，我担任了系统分析与系统设计工作。医院信息系统是指利用计算机软硬件技术、网络通讯技术等现代化手段，对医院及其所属各部门对人流、物流、财流进行综合管理，对在医疗活动各阶段中产生的数据进行采集、存贮、处理、提取、传输、汇总、加工生成各种信息，从而为医院的整体运行提供全面的、自动化的管理及各种服务的信息系统。由于行业性质的缘故，医院信息系统必须 7 X 24 小时不间断运转，因此对网络系统的安全性和可靠性有很高的要求，在该项目的系统设计阶段，我们就将网络系统的安全性作为一个重要部分考虑在内。由于该信息系统是建立在一个物理上与公众网完全隔离的局域网基础上的，所以我们并没有过多地考虑防御来自外部入侵者的威胁方面的安全问题，我们认为该系统的安全核心一是保证信息系统的正常运行，二是保证数据的安全，也就是说该医院网络信息系统的安全可以分为信息系统安全和数据安全。下面就我们在这两方面所采取的措施加以论述。

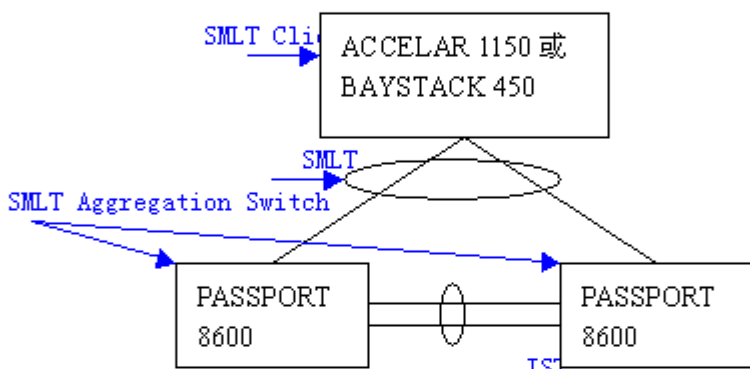
信息系统安全

信息系统安全涉及网络安全、服务器组的安全、供电安全、病毒防范等。

1、网络安全

对于医院的业务局域网，威胁网络安全的主要因素有：网络设计缺陷、网络设备损坏、非法访问等。经过充分调研，认真分析，结合该医院的实际情况，我们设计了一个主干为三层路由千兆交换以太网的网络方案。

我们采用具有三层路由功能的两台核心交换机 NORTEL PASSPORT 8600、两台具有三层路由功能的 NORTEL ACCELAR 1150 交换机和千兆级光纤组成网络主干，边沿交换机为 BAYSTACK450。本方案我们采用 SMLT (Split Multi-Link Trunking) +VRRP (Virtual Router Redundancy Protocol) 技术。NORTEL 公司的 MLT (Multi-Link Trunking) 是一种允许多条物理链路模拟成一条逻辑链路的聚合链路协议，它通过将两个交换机之间（或交换机与服务器之间）的两条或以上的物理传输链路虚拟为一条逻辑上的传输线路进行数据传输，进而可以成倍地提高两个交换机之间（或交换机与服务器之间）的数据传输带宽，同时提供了传输链路的冗余备份。当构成虚拟传输链路的几条物理链路有一条由于端口或传输介质本身失效时，不会影响数据的正常传输，所受到的影响仅仅是虚拟链路的传输带宽。SMLT，分离的多链路聚合主干，同 MLT 相比，SMLT 在构成上，不再是两个交换机之间，SMLT 的一端是一个支持 MLT 的交换机，而另一端则是由两个交换机通过 IST (Inter Switch Trunk，是连接两台聚合交换机以实现信息共享，使两台聚合交换机能作为一台逻辑交换机运转的点对点链路) 形成的一个逻辑上的交换机。MLT 交换机分别与这两个 SMLT 交换机连接，因此，SMLT 在增加带宽的同时，可以提供最高级别的可靠性——交换机级别的可靠性。两个 SMLT 交换机不论是端口失效还是端口模板失效，甚至是交换机失效都不会影响数据的正常传输，避免了单点失效对网络正常连通带来的影响。同时，传输负载由两个交换机来均衡完成，可以大幅度提高网络主干的传输性能。SMLT 体系结构由 SMLT Aggregation Switch、IST (Inter Switch Trunk) 和 SMLT Client 构成，其结构图如下：



在没有使用 SMLT 的情况下启动虚拟路由冗余协议 (VRRP)，通常只有主交换机进行数据包的转发，如果主交换机出了故障，备用交换机会自动顶替主交换机，完成数据包的转发工作；使用 SMLT，使得 VRRP 的性能得到扩展，除了主交换机进行数据包的转发外，备用交换机也进行数据包的转发，主交换机和备用交换机互为备份并互相侦听，这样既可以实现流量的负载均衡，也可以实现故障恢复，避免单点失效。为了避免边沿交换机出现单点失效，我们采用了堆叠技术，把若干台 BAYSTACK450 用堆叠电缆堆叠起来，在堆叠的某些交换机上加装光纤模块，由这些光纤端口捆绑成一条逻辑链路上联到网络主干，这样就算堆叠中的某台交换机损坏了，整个堆叠还可以正常工作。特别地，门诊收款处和门诊药房是医院的窗口单位，为了避免由于门诊楼交换机堆叠中的某台交换机出现了故障而导致门诊收款系统和门诊发药系统瘫痪，我们把门诊收款工作站和门诊发药站分散地接到堆叠中的七台交换机中。

在防止非法访问方面，我们采用了密码管理、权限设置、虚拟子网 (VLAN) 的划分等措施。

2、服务器组的安全

服务器是全院计算机网络的大脑和神经中枢，保证服务器可靠长期有效的运行是网络信息系统安全的一个特别重要的问题。

由于本方案中的应用程序是采用安全性较高的三层体系结构，所以服务器组包括域控制器、应用服务器和数据库服务器。

域控制器我们采用了两台稳定性较好的 IBM xseries 230 服务器，一台做主域控制器，另一台做备份域控制器，这样既可以实现登陆验证的负载均衡，又可以避免域控制器的单点失效问题。

应用服务器部分我们采用了六台 HP 380G3 服务器和一台 F5 BIG-IP5000 控制器。BIG-IP 控制器是针对企业本地网站或数据中心的一种产品。它能够提高可用性和智能负载平衡功能。六台 HP 380G3 服务器通过 F5 BIG-IP5000 控制器连接到核心交换机 PASSPORT 8600，F5 BIG-IP5000 控制器可以持续监视六台 HP 380G3 服务器，以确保服务器运行正常，然后再自动将输入的服务请求路由到六台中可用性最高的服务器。这样连接，只要有一台 HP 380G3 服务器不出现故障，中间层应用程序便可以正常运行。这样设计既可以实现中间层应用程序的负载均衡，同时在 F5 BIG-IP5000 控制器不出现故障的前提下，又避免了应用服务器的单点失效问题。

数据库服务器部分我们采用了一台稳定性较高、存储性能较好的 HP DL760 G2 服务器、一台 HP DL580 G2 服务器和一台 HP MSA1000 光纤磁盘阵列柜。两台服务器分别通过光纤通道连接到磁盘阵列柜，组成存储局域网 (SAN)。本方案采用了微软的群集技术，实现了 Active/Passive 双机热备份模式，HP DL760 G2 做主数据库服务器，HP DL580 G2 做备份数据库服务器，在主服务器发生故障的情况下，备份服务器将自动在 30 秒内将所有服务接管过来，从而保证了数据库服务器的正常运行。在磁盘阵列柜，我们安装了 5 块 146G 的 SCSI 服务器硬盘，其中 4 块硬盘做 RAID5，一块硬盘做 Hot spare，这样可以保证阵列柜在两块硬盘发生故障时，系统还可以正常运行。

3、供电安全

由于医院许多大型诊疗仪器设备启动时有瞬间高压、强磁场等，会对计算机产生影响因此我们要求院方做到中心机房的电源专线专供，同时采用功率足够大的 UPS。

4、病毒防范

我们通过设置 VLAN 和要求院方安装网络版杀毒软件来防范病毒。

数据安全

我们采用数据备份来保证数据安全。

本方案我们采用冗余备份策略。1、利用 Veritas Backup Exec 9.1 软件将数据备份到磁带库中。Veritas Backup Exec 能为跨网络的服务器和 workstation 提供快速可靠的备份和恢复能力。我们利用 Veritas Backup Exec 的作业管理功能设置备份定时任务，每天进行一次数据库数据完全备份，每三个月进行一次系统的灾难备份。灾难备份能在数据库服务器崩溃时，避免重装系统，利用最新的数据备份使系统尽快恢复到运行状态。2、利用 MS SQL SERVER 2000 自身的备份功能，每天定时自动地进行一次数据完全备份，备份数据存放到另一台数据备份服务器 HP ML570 中，同时在 HP ML570 中设置一定时任务，对每天的备份数据进行一次完整性检测，这样可以保证备份数据是完整、可用的。

通过数据备份，能使医院在破坏数据的灾难事件中造成的损失降到最低。

计算机网络安全是一个系统工程，除了采用保障网络安全的技术外，还要加强安全教育和制度管理，因此我们强烈要求院方重视对各级计算机操作人员进行计算机网络安全的教育，并制定较为完善的计算机网络管理制度，如严禁非操作人员使用电脑，计算机中心指定专职系统管理员掌握服务器密码，每次更新或升级计算机软件必须有两人同时在场，并做好记录等。

在整个项目方案中，我们用于保障计算机网络系统安全的措施主要是设备的冗余、链路的冗余。采用冗余措施，除了可以避免单点失效问题，还可以增加网络带宽和实现业务流量的负载均衡。因此，保障计算机网络的安全性不仅可以保证网络服务的持续不中断，还可以更好地提高网络服务效率。

整个项目完成至今近一年，从系统运行的情况来看，整个方案是合理的，高可靠性的，院方也感到很满意。当然，方案中也有不足的地方，如 F5 BIG-IP5000 控制器存在单点失效问题；随着院方的网上挂号等业务的开展，本方案中没有考虑到内网与公众网相连的安全措施等。

论计算机网络的安全性设计

摘要

我在一家证券公司信息技术部门工作，我公司在 9798 年建成了与各公司总部及营业网点的企业网络，并已先后在企业网络上建设了交易系统、办公系统，并开通了互联网应用。因将对安全要求不同、安全可信度不同的各种应用运行在同一网络上，给黑客的攻击、病毒的蔓延打开了方便之门，给我公司的网络安全造成了很大的威胁。作为信息技术中心部门经理及项目负责人，我在资金投入不足的前提下，充分利用现有条件及成熟技术，对公司网络进行了全面细致的规划，使改造后的网络安全级别大大提高。本文将介绍我在网络安全性和保密性方面采取的一些方法和策略，主要包括网络安全隔离、网络边界安全控制、交叉病毒防治、集中网络安全管理等，同时分析了因投入资金有限，我公司网络目前仍存在的一些问题或不足，并提出了一些改进办法。

正文

我在一家证券公司工作，公司在 98 年就建成了与各公司总部及营业网点的企业网络，随着公司业务的不拓展，公司先后建设了集中报盘系统、网上交易系统、OA、财务系统、总部监控系统等等，为了保证各业务正常开展，特别是为了确保证券交易业务的实时高效，公司已于 2002 年已经将中心至各营业部的通讯链路由初建时的主链路 64K 的 DDN 和备链路 33.3K PSTN，扩建成主链路为 2M 光缆作为主链路和 256K 的 DDN 作为备链路，实现了通讯线路及关键网络设备的冗余，较好地保证了公司业务的需要。并且随着网上交易系统的建设和网上办公的需要，公司企业网与互联网之间建起了桥梁。改造前，应用系统在用户认证及加密传输方面采取了相应措施，如集中交易在进行身份确认后信息采用了 Blowfish 128 位加密技术，网上交易运用了对称加密和非对称加密相结合的方法进行身份认证和数据传输加密，但公司办公系统、交易系统、互联网应用之间没有进行安全隔离，只在互联网入口安装了软件放防火墙，给黑客的攻击、病毒的蔓延打开了方便之门。

作为公司信息技术中心运保部经理，系统安全一直是困扰着我的话题，特别是随着公司集中报盘系统、网上交易系统的建设，以及网上办公需要，网络安全系统的建设更显得尤为迫切。但公司考虑到目前证券市场疲软，竞争十分激励，公司暂时不打算投入较大资金来建设安全系统。作为部门经理及项目负责人，我在投入较少资金的前提下，在公司可以容忍的风险级别和可以接受的成本之间作出取舍，充分利用现有的条件及成熟的技术，对公司网络进行了全面细致的规划，并且最大限度地发挥管理功效，尽可能全方位地提高公司的网络安全水平。。在网络安全性和保密性方面，我采用了以下技术和策略：1、将企业网划分成交易网、办公网、互联网应用网，进行网络隔离。2、在网络边界采取防火墙、存取控制、并口隔离等技术进行安全控制。3、运用多版本的防病毒软件对系统交叉杀毒。4、制定公司网络安全管理办法，进行网络安全集中管理。

一、网络安全隔离

为了达到网络互相不受影响，最好的办法是将网络进行隔离，网络隔离分为物理隔离和逻辑隔离，我主要是从系统的重要程度即安全等级考虑划分合理的网络安全边界，使不同安全级别的网络或信息媒介不能相互访问或有控制的进行访问。针对我公司的网络系统的应用特点把公司证券交易系统、业务办公系统之间进行逻辑分离，划分成交易子网和办公子网，将互联网应用与公司企业网之间进行物理隔离，形成独立的互联网应用子网。公司中心与各营业部之间建有两套网络，中心路由器是两台 CISCO7206，营业部是两台 CISCO2612，一条通讯链路是联通 2M 光缆，一条是电信 256K DDN，改造前两套链路一主一备，为了充分利用网络资源实现两条链路的均衡负载和线路故障的无缝切换，子网的划分采用 VLAN 技术，并将中心端和营业部端的路由器分别采用两组虚拟地址的 HSRP 技术，一组地址对应交易子网，一组地址对应办公网络，形成两个逻辑上独立的网络。改造后原来一机两用（需要同时访问两个网络信息）的工作站采用双硬盘网络隔离卡的方法，在确保隔离的前提下实现双网数据的安全交换。

二、网络边界安全控制

网络安全的需求一方面要保护网络不受破坏，另一方面要确保网络服务的可用性。将网络进行隔离后，为了能够满足网络内的授权用户对相关子网资源的访问，保证各业务不受影响，在各子网之间采取了不同的存取策略。

(1)、互联网与交易子网之间：为了保证网上交易业务的顺利进行，互联网与交易子网之间建有通讯链路，为了保证交易网不受互联网影响，在互联网与中心的专线之间安装了 NETSCREEN 委托防火墙，并进行了以下控制：a)、只允许股民访问网上交易相应地址的相应端口。b)、只允许信息技术中心的维护机地址 PING、TELNET 委托机和路由器。c)、只允许行情发送机向行情主站上传行情的端口。d)、其他服务及端口全部禁止。并且在互联网和交易网之间还采用了 SSL 并口隔离，进一步保证了交易网的安全。

(2)、交易网和办公网之间：对于办公网与交易网之间的互访，采用 CISCO2501 路由器进行双向控制或有限访问原则，使受控的子网或主机访问权限和信息流向能得到有效控制，主要采用的策略主要是对具体 IP 进行 IP 地址与 MAC 地址的绑定。

(3)、办公子网与互联网之间：采用东大 NETEYE 硬件防火墙，并进行了以下控制：a)、允许中心上网的地址访问互联网的任何地址和任何端口。b)、允许股民访问网上交易备份地址的 8002 端口。c)、允许短消息访问公司邮件 110、25 端口，访问电信 SP 的 8001 端口。d)、其他的都禁止。

三、病毒防治

网络病毒往往令人防不胜防，尽管对网络进行网络隔离，但网络资源互防以及人为原因，病毒防治依然不可掉以轻心。因此，采用适当的措施防治病毒，是进一步提高网络安全的重要手段。我分别在不同子网上部署了能够统一分发、集中管理的熊猫卫士网络病毒软件，同时购置单机版 KV3000 和瑞星防病毒软件进行交叉杀毒；限制共享目录及读写权限的使用；限制网上软件的下载和禁用盗版软件；软盘数据和邮件先杀毒后使用等等。

四、集中网络安全管理

网络安全的保障不能仅仅依靠安全设备，更重要的是要制定一个全方位的安全策略，在全网范围内实现统一集中的安全管理。在网络安全改造完成后，我制订了公司网络安全管理办法，主要措施如下：1)、多人负责原则，每一项与安全有关的活动，都必须有两人或多人在场，并且一人操作一人复核。2)、任期有限原则，技术人员不定期地轮岗。3)、职责分离原则，非本岗人员不得掌握用户、密码等关键信息。4)、营业部进行网络改造的方案必须经过中心网络安全小组审批后方可实施。5)、跨网互访须绑定 IP 及 MAC 地址，增加互访机器时须经过中心批准并进行存取控制设置后方可运行。6)、及时升级系统软件补丁，关闭不用的服务和端口等等。

保障网络安全性与网络服务效率永远是一对矛盾，在计算机应用日益广泛的今天，要想网络系统安全可靠，势必会增加许多控制措施和安全设备，从而会或多或少的影响使用效率和使用方便性。如，我在互联网和交易网之间设置了放防火墙的前提下再进行了 SSL 并口隔离后，网上交易股民访问交易网的并发人数达到一定量时就会出现延时现象，为了保证股民交易及时快捷，我只好采用增加通讯机的办法来消除交易延时问题。

在进行网络改造后，我公司的网络安全级别大大提高。但我知道安全永远只是一个相对概念，随着计算机技术不断进步，有关网络安全的讨论也将是一个无休无止的话题。审视改造后的网络系统，我认为尽管我们在 Internet 的入口处部署了防火墙，有效阻挡了来自外部的攻击，并且将网络分成三个子网较减少了各系统之间的影响，但在公司内部访问控制以及入侵检测等方面仍显不足，如果将来公司投资允许，我将在以下几方面加强：

- 1、在中心与营业部之间建立防火墙，通过访问控制防止通过内网的非法入侵。
- 2、中心与营业部之间的通讯，采用通过 IP 层加密构建证券公司虚拟专用网 (VPN)，保证证券公司总部与各营业部之间信息传输的机密性。
- 3、建立由入侵监测系统、网络扫描系统、系统扫描系统、信息审计系统、集中身份识别系统等构成的安全控制中心，作为公司网络监控预警系统。

图书馆网络应用体系安全设计

摘要

某某大学图书馆从 85 年引进日本富士通的管理系统开始, 历经近 20 年的信息化建设后, 逐步形成了拥有一定硬件规模、软件资源和一批专业技术人才的现代化图书馆。而自从我校进入“211”以来, 建立数字化的图书馆就成为了我们工作的重中之重。我校数字化建设的主要内容是建立基于千兆主干网络的、提供多种网络服务的网络应用体系。项目建设完成后, 我馆成为该省规模最大的开放式的数字化文献提供和建设中心, 为本省的科技、文化、教育事业的发展提供了强大的资源保障。

2001 年 3 月, 我参加了某大学数字图书馆建设, 担任技术部主管职务, 负责理解学校和省教委对该项目的要求, 根据技术先进、成本适中、充分满足要求为原则, 进行应用需求分析, 对整个网络进行设计, 提出设计方案。由于该项目规模比较大, 提供的服务比较多, 作为校园网络的中心和提供网络服务的核心部门, 我馆图书馆的许多业务需要在网上展开, 各种特色服务所用的平台也是五花八门。这对安全方面的设计提出的较高的要求。我们通过采用保障 Internet 接入的安全、保证干路畅通和合理划分子网、保证软件系统的安全、健全管理机制等措施来设计图书馆网络。这种切合实际、低成本、高技术的设计方案实施后的网络, 其安全性大大加强了, 同时网络的性能并未受到太多的影响。

正文

2001 年 3 月, 我参加了某某大学数字 211 建设图书馆数字化建设部分(分为第一、第二、第三期工程)。第二期工程总额 5500 万元。该工程建设项目分为网络部分、软件部分以及资源建设部分。该项目的目标是, 将地处该城市的东、西、南、北端的各分校区和总校区的网络联连接成为一个畅通的宽带、高速、高性能的校园网; 建立一个文献信息中心, 能为读者提供电子期刊服务、图书书目数据服务、图书光盘点播、视频点播服务, 并且能够拥有部分自己的特色数据资源; 建立一个网络中心提供 Web 服务、邮件服务、全校的办公自动化服务, 它既是全校的应用服务中心、公共数据存储中心, 又是全校网络的管理中心。项目从实现的功能、设备配置的先进性、网络带宽的规模看, 该项目要达到国内一流、省内第一的水平。项目建成后, 该数字图书馆能承担全省的文献检索与开发的重任。我在该项目中担任技术部主管, 其任务是充分理解学校和省教委对该项目的要求, 根据技术先进、成本适中、充分满足要求为原则, 提出和需求分析书和项目计划书; 监督光纤的铺设和楼宇布线; 设备的货到验收; 网络调试(交换机调试、地址分配、访问控制配置、设备网管模块的配置、网络管理平台的配置); 网络应用系统的建立; 数据库的采购; 建立特色的数字化资源等等。进行应用需求分析, 对整个网络进行设计, 提出设计方案。

由于该项目规模比较大, 提供的服务比较多, 各种特色服务所用的平台也是五花八门。同时, 影响网络安全的因数比较多, 通常有人的因数、自然因数、病毒因数。如果对这些主要因数防范不得力, 将影响网络硬件、网络数据传输、数据服务器的安全。因此对其安全设计, 单一的技术或者设备保护难以保证本校网络的安全, 效果往往不理想; 所以我们采用多层次的防护体系来保证网络的畅通和网络数据的安全。我们根据业务数据的流动历经的重要环节, 来提出具体的安全方案。

1 保证接入的安全

俗话说病从口入, 同样一些常有的攻击往往来自外网, 同时使用图书馆所购买的外文资料比如 Web Science、EBSCO 等数据库需要访问外网; 同时, 外网的授权用户需要能够访问我们的数据。所以必须保证接入部分的稳定与安全。在外网联接上, 我们租用 2 条光纤线路(100M)分别接入中国电信和 XXX 大学, 使用 Cisco 的 7500 系列路由器接入, 然后用一台 Sun spark 操作系统为 Solaris 的服务器做防火墙将外网和内网划分开; 同时用其做策略路由服务器, 使用校园网内的用户可以快捷的访问电信网络和教育网络的资源。在外部用户访问上, 我们首先用 Sun spark 服务器防火墙进行 IP 地址过滤掉非法的 IP 地址, 然后通过用户名+密码模式登陆, 才能通过防火墙访问我们内部资源。

2 保证干路畅通和划分子网

网络干路是指各楼的骨干网、骨干网汇接形成的交换中心及联结 Internet 的接入线路。它的不畅通必然导致大规模的网络瘫痪, 外部付费用户访问不到内部资源, 这将造成极坏的影响。

在光纤干路上我们采用光纤连接主校区的各大教学楼、南院校区和 XXX 校区, 主干路使用 3 对光纤做冗余。由于其它分校区离主校区比较远, 光纤连入费用较高, 我们采用拨号接入的方式联结。我们采用 3ComRAS1500 作为拨号服务器, 分配 32 个校内电话号码给该服务器, 用户只需要拨这 32 个其中一个, 通过认证后, 就可以动态分配一个校内的合法 IP 来访问校内的资源。在干路上, 安全的隐患往往来自非技术因素, 由于本市修建大学城, 道路施工比较多, 我们的光纤被人为的挖断; 由于, 某路段起火烧掉部分光纤。对于这种情况, 我们采取紧急修理, 租用吉通的 2M 微波通讯线路和将策略路由指向电信线路的办法来保障网路的畅通。

由于本校应用点多, 地理位置也十分分散, 各大单位为相对独立的机构, 所以需要进行子网划分以便管理。

我们使用两台 3Com 的 CB9000 中心交换机连接各教学单位过来的光纤, 并且将华中分配的 16 个 C 类地址分成为 32 个 VLAN。子网的划分使安全性得到了很大的提高, 同时各单位的网络独立的虚网。同时, 每个教学单位指定专人负责网络地址的分配, 这样盗用 IP 地址的情况大大的减少了。

我们采用 3Com 的 Transcend 作为网管平台, 实施对网络干路和各个子网的监控和进行网络优化。当网络有人恶意下载资源和蠕虫病毒群发邮件时, 我们可以立即找出攻击源而进行相应的措施。

3 保证软件系统的安全

由于操作系统是软件系统的基础, 所以, 保护好操作系统是必须的, 对于 Windows 系统(提供 CNKI 和 VIP 镜像等服务)保护, 我们主要采取取消 Guest 帐号、取消不必要的服务(如远程注册表操作)安装诺顿防火墙和相应的杀毒软件、定时查看有无异常的程序运行等方式来保证其安全。对于 Linux 或者 Unix(提供邮件、网络计费、主页、论坛等服务), 我们关闭许多不必要的端口、定时对系统打包升级等措施来保证其安全。

对于数据库系统的保护, 主要集中在保证数据的完整性、正确性和安全性方面。在数据存储介质方面, 我们采用磁盘阵列、NAS、San 结合 RAID5(无独立校验磁盘的奇偶校验磁盘阵列)方式存储数据。

4 健全的管理机制

管理安全数字图书馆的工作状态在很大程度上取决于是否有良好的管理机制。如果制度合理、管理得当、执行得力就能有效的预防和控制安全事故的发生。反之, 则会引发各种安全隐患。

我们规定工作人员每日必须检查服务器的日志, 通过日志可以清楚的看到有无外来人员登陆服务器, 并且做了何种修改。对重要的数据服务器, 每日必须做异地数据备份。同时管理员的密码必须达到一定的长度并且每周建议修改一次。

一般而言, 网络安全是网络服务效率的保障。没有了数据安全, 网络服务成为无源之水; 没有网络系统的安全, 网络服务将成为无本之木。当然没有效率的网络传输是得不偿失的。所以, 我们要合理的规划好网络及软、硬件的合理利用。

在接入上, 由于全校有 3, 000 多台机器也要通过该做防火墙的服务器接入 Internet, 这将给该服务器造成较大的负担, 同时数据传输量大时, 该服务器处理速度相对将成为瓶颈。我们采取单独使用两台 Spark 主机做代理服务器, 代理服务器同时启动用户认证服务, 由于代理服务器的缓存保证了出校的流量大大减少, 同时认证功能使得非法用户无法登陆网络。这样的冗余设计使得网络的安全性能和效能得到了大幅度的提高。

在子网划分上, 连接各教学单位过来的光纤的 2 台中心 3Com CB9000 交换机之间用两根光纤作一个串口, 这样跨子网的访问无需通过交换能力不强的 Cisco 路由器 7500, 子网间传输数据的瓶颈问题消失了。

在数据存储上, 由于我们提供 CNKI 和 VIP 等服务, 这些服务器的访问量、存储的数据量都十分庞大, 单个的数据检索查询都会占据大量的 CPU 时间。我们使用 SAN 和 NAS 存储大量的数据, 当用户提出一个检索的申请时, 检索服务将在服务器上运行, 当用户提出一个下载的服务时候, 下载的服务由 SAN 和 NAS 去完成, 这样使得 SAN 和 NAS 的效能比磁盘阵列要高。由于 SAN 和 NAS 都可以做 RAID1~5, 所以数据安全性能比普通的硬盘要高。采用新技术和冗余使得我们的网络服务效率和安全性能得到了大幅度的提高。

网络的安全管理不是一个硬件和软件的引进就能得到大幅度的提高的, 他是一个长期的、动态的过程。本项目的网络系统还有许多问题, 一次网络发生了大规模的网络风暴, 从子交换机扩散到中心交换机, 主机与主机之间的数据传输丢包率达到了 75%几乎是不通了, 最后我们发现一个施工的单位将一根漏电的线缆搭在网路上。但是为什么会造成跨虚网的网络风暴, 从技术上我们无法解决, 只能是严格检测各子网和主干网络是否达到屏蔽的要求, 并且将电路和网路尽量分开。

论 XML 技术在 Internet 平台上的应用

摘要

2002 年 10 月,我参与了一个三层在线商城的项目开发,该项目整合了来自不同商家的信息,方便在线用户的查询和购买。

在该项目中,我担任系统分析的工作。在分析设计过程中,我借鉴了 XML 成熟的技术,采用 Java 语言,整个系统由三层组成。在数据层,对于不同的数据库,最后都以 XML 数据的形式来实行数据间的转换和处理。在业务逻辑层,在联机会话的持续时间内,用户的帐户数据在内存中以 XML DOM 形式表示,在表示层,所以给用户的信息首先都封装成 XML 数据,然后用服务器或者客户机附带的 XSLT 转换,根据浏览器的性能将 XML 数据转换为 HTML 在前端显示。

在设计过程中,如设计 XML 的各个基本元素,我应用域分析的方法,在采用 XML DOM 形式的时候,分析比较了其他的形式,在将 XML 转换为 HTML 的设计中,引用了 XSLT。

正文

随着 Internet 信息技术的发展,我所在的公司准备开发一个网上商城,这样各个商家就可以把自己的产品信息在这个网上商城中发布,并且提供了在线购买。也就是开发一个电子商务平台,在这个平台上,整合了来自不同商家的信息,方便在线用户的查询和购买。有点类似现实中的商城,为各个商家提供地方,方便消费者购买。

本人有幸成为这个项目的系统分析员之一,参与了这个系统的设计,并且对系统中的关键实现技术也进行了一一验证。整个网上商城系统由三层组成。在数据层,由于各个商家自己有着不同的数据库来存储自己信息,为了实行信息在同一平台上的共享和处理,我们采用 XML 数据的形式来实行数据间的转换和处理。在业务逻辑层,在联机会话的持续时间内,用户的帐户数据在内存中以 XML DOM 形式表示,在表示层,所有给用户的信息首先都封装成 XML 数据,然后用服务器或者客户机附带的 XSLT 转换,根据浏览器的性能将 XML 数据转换为 HTML 在前端显示。

在数据层,我们面临的第一个的问题就是如何统一认识将要采用的 XML 数据的元素。刚开始是由一个人来定义 XML 数据的元素,但是后来发现,这样定义处理的 XML 元素很难获得别人的认同,并且对于不同的商家,所定义的 XML 元素不具有代表性。于是,在争取到领导的支持后,把以前的 XML 元素设计推倒重来,而且借鉴了国外关于类似设计所采用的域分析的方法,该方法就是一个用于确定网上商城这个域的术语,范围,共性和变性的过程。就这样为了寻求一个统一的 XML 元素的定义,我们成立了一个小组来进行的网上商城的域分析,这个小组有商家代表,数据库设计员,参与多个项目的有丰富经验的程序员和一个专门指导该组域分析的大学专家组成。

在小组会议的开始,我们首先达成一个共识,那就是需要采用一直标准术语来避免交流中产生的误解。于是,我们在刚开始的一周内先确定的关于网上商城的一下公用术语,比如 store (商城),shop (店面),ware (商品)等,作为 XML 最基本的元素。

在小组讨论的过程中,有成员建议对于域分析,分多个阶段进行,每个阶段提交域分析报告,比如第一阶段的域分析报告主要为标准术语以及各个商家信息共性和可变性的表格,第二阶段的域分析报告,对于域分析中的商家可变性的东西进行详细说明。我们采纳这个建议,获得了很好的效果。就这样,通过 3 周的域分析,我们小组最终提交一份完整的关于 XML 网上商城元素的详细表格,由于这些 XML 元素是由各个部分和商家代表讨论处理的,所以很快被整个项目组成员所接收,并且这些 XML 元素在后来的开发中证明是非常完整,能够清楚地反映数据的结构,大大提高了整个的系统开发效率。

在业务逻辑层,由于所有数据库的信息都被转换为 XML 数据结构,所以在处理数据库信息的时候还必须对 XML 数据进行 XML 语法分析,并且将分析出来的结果送往程序。在这儿我们考虑采用 XML DOM (文档对象模型)来保存这些语法分析出来的 XML 数据。比如在联机会话的持续时间内,一个用户的帐户数据首先从传统的关系型数据库中读取出来,转换为 XML 数据结果,并且通过一个 XML 语法分析器,将 XML 数据转换成为 DOM 对象保存在内存中,程序通过 Java 的 DOM API 访问这些对象。在这儿为了更好地选择处理 XML 的技术和方法,我还比较其他两种分析 XML 数据并且保存的方法。

比较了几种方法,发现 DOM 有个缺点就是当它保持的数据非常多的时候,将大量占用内存的存储空间。但是,使用 DOM API 也有一个明显的好处就是简单,它可以通过 Java 程序直接使用一些方法调用 DOM 树上的数据。为此在设计的时候,尽可能地重复调用已经存储在内存中的 DOM 对象上面的数据,避免对于相同的数据有多个 DOM 对象存在。在编码过程中面临的问题不是很多,关键是让程序员熟悉 DOM API 的各种调用方法。

在表示层,考虑到将大量的运算负荷分布在用户端,既用户可以根据自己的需求选择或者制订不同的应用程序以处理数据,我们设计把所有给用户的信息首先都封装成 XML 数据,然后用服务器或者客户机附带的 XSLT 转换,根据浏览器的性能将 XML 数据转换为 HTML 在前端显示。这样的话,服务器只需要准备一次的 Web 内容,尽可能完善,准确地将数据封装到 XML 文件中,而 XML 的自解释性可以使得用户端在接收到 XML 文件的同时也理解数据的逻辑结构和含义,再通过转换程序,XML 可转换为用户所需要的个性的多样的 HTML 显示方式。在设计中我们所选取的 XSLT 是一种用于操作 XML 文档的高级语言,就像 SQL 是操作关系表的高级语言一样。

一个 XSLT 规范本事就是一个 XML 文档,我们通过它的规范,描述出各个用户可以选择的各个 HTML 显示模板,这样客户端程序只需选择好模板,再加上接收到的 XML 数据文件,就可以方便地生成自己个性化的 HTML 文件,并且在浏览器中显示出来。当然在开发中遇到了不少问题,首先由于 XSLT 是一门起点比较高的语言,也比较烦琐,掌握起来比较慢,使得对程序员的要求比较高。这样导致使用 XSLT 语言编写的 HTML 文件的模板过于冗长,作为设计师我一方面是希望模板程序尽可能地包含 HTML 页面所有的功能和美观,一方面又希望模板程序尽可能地短少,简洁,便于修改。

为了达到两种选择之间的平衡,以达到最佳效果,我对于基本模板的设计,采用有经验的程序员进行设计,并且要求尽可能多地写全文档,并且频繁地开展小组会议,对页面设计和美工人员详细讲述模板各个部分和功能。

在 Internet 平台上采用 XML 技术,明显的效果之一对于不同数据库的支持,通过将各种数据转换为 XML 文件,可以实行了数据间的转换,共享和处理。效果二是支持了用户的个性化服务,支持用户在不同的客户端可以个性地选择显示界面。

由于 HTML 在许多复杂的 Web 应用中遇到了问题,为了彻底解决这些问题,必须采用功能强大的 XML 来代替 HTML 作为 Web 页面的书写工具,而 XML 的广泛使用,必定能够推动 Web 的不断发展,开创 Web 应用的新时代。对于 XML 技术在 Internet 平台上的应用,我更关注它的扩展性,既让 XML 包含更加丰富更完整的数据信息,目前公司所接触的首先是软件模型的交换和模型信息的保存,打算在 XMI (XML Metadata Interchange)展开工作。XMI 作为一种试图通过 XML 语言为程序员和其他用户提供一种交换元数据信息的标准途径,是作为 MDA 模型驱动的模式交换的基础,是非常具有意义的工作。

XML 在网上银行中的应用

摘要

网上银行是指在 Internet 上提供银行服务,即银行的客户无须到银行柜台办理业务,可以在家庭、办公室等能够连入 Internet 的任何一处,登录到银行的网站进行交易。这是一种崭新的银行运营模式,具有方便快捷、成本低廉、不受时间地点限制等优点。

本文通过论述的项目是某银行行网上银行系统的 1.0 版本到 2.0 版本的升级和改造,论述了 XML 在 Internet 中的应用。我有幸参加了这个项目,承担在该项目中担当了部分的分析与设计的部分任务。

系统的 1.0 版本存在诸如交易超时、作业比较慢、不能满足客户个性化、技术相对落后等缺点。在 2.0 该项目版本的设计和开发过程中,我们基于 JAVA 技术,采用 J2EE 构架,使用应用了 XML 作为数据交换的标准,。在后台,基于业务数据建立了 XML 数据库,存放签约客户的历史数据,同时在 Web 服务端,我们也应用了 XML,读取 XML 数据库中的数据,同时给客户提供了“个性化”的服务。这些技术的采用,解决了 1.0 版中存在的问题。但是,因为 XML 是一种新的标准,有些地方还不是很完善,在 J2EE 架构下,如何使用 XML 是我们应该一直关注的问题,本文就该问题也有所论述。

正文

2002 年 3 月,我参与了某银行的网上银行系统的升级和改造工作。该系统采取总行、分行两层结构,总行网银中心连接各一级分行,提供信息服务、客户服务、帐务查询和实时交易等功能。网银中心与客户通过 Internet 相连,与分行业务主机通过城市综合网相连。网上银行的客户使用 Browser(浏览器)通过 Internet 连接到网银中心并且发起网上交易请求;网银中心验证后将交易请求返回;分行业务主机完成交易处理,返回处理结果给网银中心;网银中心对交易结果进行再处理后返回相应的信息给客户。

原有系统无论从业务上,还是从技术上通过运行一段时间,存在这很多问题,客户和业务人员意见也很多,诸如部分时间交易超时,作业比较慢,还有无法满足客户的某些个性化的要求,基于 ASP+VB 的应用也不合时宜,这些都是系统需要改造的原因。

新系统整个网上银行应用的开发,全部基于 JAVA 技术,数据的交换采用标准的 XML 协议,应用开发采用 Websphere Studio+Visual Age for Java 等工具进行,采用符合国际 J2EE 标准,系统采用了业界领先的中间件产品(BEA WebLogic)建立网上网上银行系统的交易平台,该系统的主控程序应用 JSP 和 servlet 编写,很好地发挥了多线程机制,大大提高了系统的性能;主要业务逻辑采用 EJB 技术实现,模块化结构利于新业务的开发与布部署;数据库访问符合 JDBC 标准,利用 WebLogic 的 Jdbc pool 提高了数据库访问的效率。

在这里主要谈谈 XML 在该项目中的应用。

XML 是一种具有描述数据功能的语言,它十分适合作为知识表示语言或作为组件及文件格式的表示方法。它还可以让数据在不同的来源中,根据通用的语法规则来处理。而 Java 则是用于 Internet、适合于分布式环境、提供了一个跨平台的语言。XML 和 Java 相结合主要原因是基于 XML 的语法提供了一种灵活的、标准的、健壮的 Java 编程方案。

在该项目中这里,我们统一了 XML 的报文标准,列举了 XML 实例,规定了 XML 头部,XML 根节点,XML 二级节点及具体的报文字段。

在接口中的 XML 报文遵循如下约定:XML 头部如实例所示,不得改变;XML 根结点、二级节点如实例,不得改变;报文必须是包含“\0”作为结束符的字符串;接口格式说明中的字段即指实例中具体的报文字段这一级。对于接口文件,规定所有发送、接收的文件名均为调用方确定,均带文件路径。

接下来谈谈原有系统存在的问题:

因网上银行业务的特殊性,Web 客户端需要连接多个业务种类,多种数据库,跨平台,跨数据库,环节多,这就是原有系统交易缓慢的重要原因之一,而且随着可提供新业务的种类的增多,问题暴露的更加明显;其次 Web 客户端的查询往往对各个应用系统的服务器的负载产生影响,影响了其他业务;同样,对 Web 客户无法提供个性化服务,B2B,B2C 提供的信息和帐页千篇一律,根本无法按照其要求定制;客户端无法进行一些运算,一些简单的比如“还款试算”等还要通过后台应用服务器来进行运算,加重了主机负担。

在具体应用中,比如查询某客户的所有业务是困难的,这具体体现在,如果某签约客户想查询或执行信用卡的历史交易,必须向后台相应的业务的数据库服务器提交数据查询请求,而此时的后台服务器又往往又是银行的实时业务处理机,在高峰期正忙,并且历史交易的数据往往由于年终转换的原因,存在于其它数据表甚至其他数据库和服务器,都给查询带来了困难,只能作到有限的查询,或提供的数据项有限,或甚至根本无法提供,比如有关储蓄业务的有关明细帐页已接近 3000 万条,在高峰期间在其中取几条记录是困难的。另一方面,由于网上银行涉及多个对私和对公的业务,甚至包括资金的清算,一笔交易要跨越多个业务的服务器,这又存在跨不同种类的数据库的问题。

所以,将历史数据分离及整和是必然的,我们也曾考虑过将数据进行归类,建立一个类似“历史数据服务器”(或是数据仓库)上,但该历史数据仍需建立在某种数据库上如 INFORMIX-ONS,仅作到了数据的集合,没有治本,投入大,费时。

所以为了解决原有系统中存在的问题,我们在新系统中我们建立专用的转换服务器,作为“缓存”,目的仅是为了“脱离”原有的依赖,减少联机处理,这样,有关历史的交易就不用分别直接连到所需业务的数据库或服务器,我们采用 XML 格式进行中间的转换。这种所谓的 XML 数据库其实就是文档的集合。我们用了一台 IBM PC SERVER 来存储 XML 数据,具体就是用 UNIX 下的文件系统来存储和检索。将部分数据转换成 XML 文件,包括所有已签约的客户的历史明细帐及全部卡号(未来新开户时就不用再去信用卡主机进行校验了),同时 XML 文件的存储按文件系统,并对 B2B 和 B2C 进行了分类。

这里我们使用了 Informix Web DataBlade 工具,按照我们自己定义的数据结构进行了批量转换,基本与网银后台数据库服务器的数据库表结构相符,这样做的目的是为了便于更新,且对原有有关调用数据库的数据逻辑改动不大,将数据转换成了 XML 文件格式,对于其他非 Informix 数据库的数据,我们则采取了先将数据倒入导入至 Informix 数据库,再进行这种转换。

有关当日的账务,我们在日终将当日发生的流水帐进转换,存入了数据库,作为了历史交易的追加。在这里 Informix Web DataBlade 可以通过一个简单的 SQL 接口产生动态的 XML 数据和文档,用于日终更新。这样实际就上在数据库与应用服务器之间采用 XML 作为信息缓存。

建好了 XML 数据库,那么又如何来应用呢?接下来我们谈谈这方面的问题。

首先读取 XML 数据。具体的方法为:一个页面通过服务器端对象与 XML 数据源相连,将信息转换成数据抽象,接着用 JSP 元素显示数据。这样我们就使用了 XSLT 转换程序转换 XML,。在这里应用到的 XSLT 技术,XSLT 是 W3C 小组制定的一个转换语言规范,它可以用来将 XML 数据转换成 HTML、PDF 或其它 XML 格式。具体过程就是首先定义了 XSLT 模板,然后进行转换,除了模板需要定义外,这个解析的过程是还是比较容易的。其实 XSLT 与 XML 的关系,就好象 SQL 与表格化的数据一样。只是在 JSP 中,需用 scriptlet 或自定义标签中编程激活一个 XSLT 处理器来进行转换。在 XML 数据源上使用一个转换程序,或者是抽取数据或者是创建新的格式,这个转换程序可以使用许多不同的机制来实现并且通过自定义标签来访问它。

接下来 XML 应用就是使同样的数据可以以不同的浏览方式出现在浏览器中,而这些数据并不需要从再次从 Web 服务器上下载。其实这类应用的就是早期的 Web 上的另我们头疼的“动态表格”。

对于 B2B 的用户,可根据企业要求的,制定不同的往来对帐单,约十多种,以满足其对帐的需要,对其所属职工的集体办理的有关代发工资,缴存公积金,还个人贷款的有关帐务,都有良好的支持,在单位就可了解到,随时的变动,并可随时打印,不用再跑银行进行询问,取帐页。将来甚至直接提供信息化较高的企业 XML 数据,这也是未来交换数据的标准,也是应用 XML 的美好憧憬。对于 B2C 的个人用户,可根据其所开办的业务的不同,来对其提供“个性化”的服务,可提供该人名下的所有帐户的信息,如存贷款信息,各种缴费的信息,并可根据其需求方便的进行排序,筛选,组和,打印。甚至可以自己定制格式,改变了以往显示单一,多业务罗列,操作复杂的状况。Java 提供的 JSP 为多种基于 Web 的用户产生基于 XML 的标记语言的问题,也就是我们要达到的“个性化”界面。

在具体应用中,我们是用从 JSP 页面产生的 XML, XML 在 Web 界面层的应用得益于 JSP 技术的发展。同样,也有两种方法,一种直接将 XML 数据源集成到 JSP 的界面中去的方法是,将 XML 加载到 JavaBeans 组件中,然后在 JSP 中直接引用这些 JavaBeans 组件。最大好处是使我们的程序代码集中在一个地方(对 Java 技术而言,一般是指在“类”中),清晰,易于管理和修改。另一种方法是直接将 XML 数据转换成 Web 显示内容的另一种方法是使用 XSL 和 XSLT,将 XML 数据映射成 HTML(或 WML 等)的逻辑由 XSL 样式表(XSL StyleSheet)来定义。样式表描述了每个特定 XML 数据实体应该怎样转换成界面数据实体(如 HTML 表格、内联标记等),采用一套自定义的 JSP 标记并引用某个 XSLT 处理程序,也就是前面提到的模板。

通过比较综合了这两种方案方法,我们作出了选择,选择了后者,XSLT 方案的伸缩性要好一些,而且具有更好的可管理性。在这种情形下,我们的转换逻辑是编写在一个 XSL 样式表中,而不是在 Java 代码中。这意味着当需要修改界面时,大多数情况下只是编辑样式表或者 HTML,代码不受影响,不涉及程序编译的问题,业务人员经过简单的培训就可以修改,这对我们很重要。

在网银项目中由于 XML 的应用,解决了我们许多历史问题,也使银行的 B2B 和 B2C 业务进行了有效的整合,更好的为客户服务。

尽管 XML 是未来网上数据格式的标准,可在具体应用中还存在许多问题:

尽管我们由于网上银行所要求的数据格式不是很多,可建立 XSLT 的模板并不是一件容易的事,不是很好写,主要是由于表中嵌套太多,尽管是文本形式,阅读和修改很实际上是复杂的。

另外,这种基于文件系统的后台历史数据按 XML 存储的方案还有待探讨,尽管 XML 的确与数据库有相似之处,但对于索引、安全机制、数据完整性等特性是不具备的。在实际应用中,我们目前由于签约的客户数量不是很多,不足万人,尽管我们只应用了一台服务器来存储数据,目前还没有什么问题,但随着签约客户的增多,不知会不会有问题。看来 XML 还是要发展,想取代数据库还是需要一段时间的。

还有,我们未将 XML 封装进 JavaBean,却将部分应用逻辑由 XSLT 来实现,这种方法是否合适还有待探讨,从某种程度来说是不符合惯例的,不仅仅是 Web 服务器负载的问题,它给编程人员和维护人员带来的问题有些还是不可预计的。

我想,未来 XML 会成为数据交换的标准的,我们也考虑逐步将我们现有的数据转换成 XML 格式。接下来,我们还将在这些数据的基础上,建立一个统一的,用 XML 实现通用 WEB 报表的系统,在这方面已有了成功的案例。

现在,大部分数据库支持 XML 格式的数据查询和转换,包括 SQL Server 2000,ORACLE,IBM DB2 等大型关系数据库,估计会越来越多,我想以上问题也会很快得到解决的。

论软件三层结构的设计

摘要

随着中间件与 Web 技术的发展,三层或多层分布式应用体系越来越流行。在这种体系结构中,将应用功能分成表示层、功能层和数据层三部分。

本人在去年参加了一个备件流程管理项目的开发,在此项目中担任需求分析和结构设计等工作。结合需求分析结果和该单位的实际情况,在该项目中我们采用 C/S 和 B/S 的混合模式,客户端使用的是 Delphi 和 FrontPage 进行开发,中间件我们采用的是 COM+,使用 Delphi 进行开发,后台使用 SQL Server 数据库。本文详细描述三层结构的设计过程,重点讨论中间件的设计过程和在设计实施过程中碰到的一些问题以及解决的方法,文章最后说明了采用三层结构带来的效果,以及可以改进的地方。

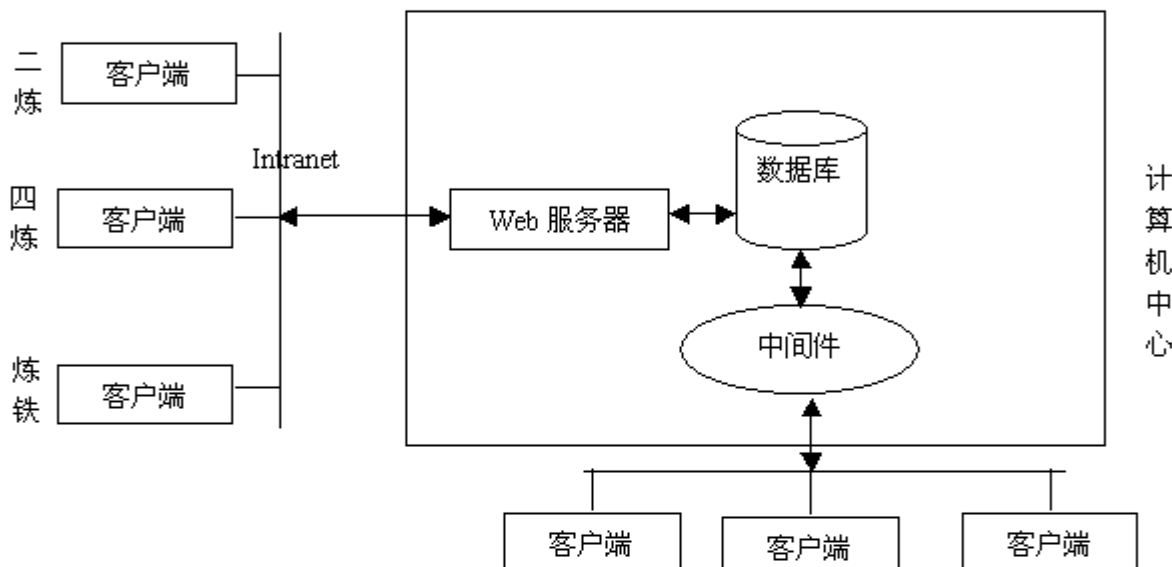
正文

2003 年 8 月,我参加了某钢铁公司备件采购项目,主要负责该项目的需求分析和结构设计等工作。该钢铁公司原先的备件采购全部是人工完成,手续复杂,办事效率低下,更重要的是无法提供及时有效的信息供企业领导参考。该公司一方面需要通过此项目来缩短采购时间,提高办事效率,另一方面该公司已经建成比较完善的局域网,需要在内部网上公开采购结果,并要做出统计分析,供全公司员工和领导查询参考。该项目涉及部门主要包括该集团公司下的仓储公司,供应公司,招标公司,劳企部等。

根据我们做出的需求分析以及各种体系结构的优缺点,我决定采用 C/S 和 B/S 混合的体系结构来开发。对于仓储公司等部门的需求,需要对数据进行更新处理,采用 C/S 结构可以更快更好的开发且数据处理速度更快,可以更好的满足要求。对于领导和员工的查询需求,我们采用 B/S 模式。采取这样的结构可以很好的满足用户需求,且容易开发和维护。由于都是在 windows 平台上使用,因此在开发工具的选择上,我们使用 Delphi 来开发仓储公司等部门的客户端和中间件,使用 FrontPage 来开发网页,连接在其内部网上提供查询服务。中间件我们采用的是 COM+,进行逻辑处理,数据层使用 SQL Server。

以下详细介绍三层结构的功能分配和物理分布,描述三层结构设计的过程,讨论在设计实施过程中碰到的一些问题以及解决的方法,文章最后说明采用三层结构带来的效果,以及可以改进的地方。

对于客户端,B/S 结构仅提供查询功能,使用 Internet Explorer,各个子公司、二级部门都可以通过内部网使用。C/S 结构提供日常操作和管理界面,承担着整个系统的数据录入及数据维护工作,使用 Delphi 开发,它是系统数据的入口,使用频繁,安装在仓储公司、供应公司等单位;中间件和数据库以及 Web 服务器都放在集团公司的计算机中心,便于维护管理。中间件负责根据客户端要求从数据库中取得数据,并在进行处理后提交到客户端显示;后台使用 SQL Server 数据库,数据集中在数据库服务器进行管理,方便数据管理和分析,保证数据安全。其物理分布如下图:



对于 C/S 和 B/S 结构,我们分别使用不同的工具来开发客户端。C/S 结构的客户端我们使用的是 Delphi 来开发,对于 B/S 的客户端我们使用的开发工具是 FrontPage,采用 VBScript 脚本语言来开发。客户端只用来显示结果,所有的逻辑处理我们全部放在中间件中。在客户端只需引用中间件提供的接口即可,这样一个方面使逻辑处理集中,便于维护,另一个方面只向客户端提供处理后的数据,可以减少网络流量,加快反应速度。

对于中间件的设计是我们工作的重点。Microsoft 的 MTS/COM+不但能够稳定地执行应用系统的企业对象,以服务客户端的请求,更能够提供在分布式环境和异质数据库之间保护数据的事务管理能力。这让分布式应用系统能够稳定、可靠地在复杂的环境中正确地执行。再加上 MTS/COM+能够有效地利用各种系统资源,增加中介软件的执行效率,因此使用 MTS/COM+作为中介软件的应用系统能够提供合理的执行效率。

经过考虑,我们决定使用 COM+作为中间件来开发。如何设计出合理的中间件关系到项目的成败。根据项目的特点,我决定根据不同的部门以及各个部门的需求来开发 COM+组件。因为各个部门有不同的数据表,中间件主要处理的是每个部门各自数据处理和内部网上的查询以及统计分析处理,所以为每个部门设计了数个 COM+组件,分别用来处理数据维护,查询操作以及统计分析工作。Delphi 提供了 MTS 的向导可以帮助程序员开发 COM+应用系统,程序员可以直接在 COM+数据模块中放入 ADOExpress 组件来存取数据库,提高了程序员的工作效率。

但在开发过程中,也出现了不少的问题。比如在仓储公司维护更新工作的设计中,使用到多个数据库表,有些程序员直接使用 MTS/COM+数据模块,然后在 MTS/COM+数据模块中加入 TADOConnection 连接数据库,再分别使用 TADOQuery/TADOTable 和 TDataSetProvider 组件连接到这些数据表。

从功能上说没有什么问题,但我发现这样设计存在一些问题。其中最大的问题就是执行效率的问题。这可以从数个不同的角度来看,首先程序员把所有的数据存取组件放在一个 MTS/COM+数据模块中,因此当客户端建立这个 MTS/COM+数据模块时需要花费许多的激活时间。第二个问题是如果客户端只需要使用其中的一个数据表的数据,那么客户端仍然需要花费多余的时间建立不相关的对象。最后,对于 MTS/COM+提供的 Pooling 机制而言,这样设计系统架构也是不好的,程序员应该尽量利用 MTS/COM+提供的数据库连接 Pooling 的功能。因此最好是把每一个 TADOConnection 的连接放在不同的 MTS/COM+数据模块中,因为这些数据表都位于一个相同的数据库中。

为了解决这些问题,我重新设计了这个架构。为充分利用 MTS/COM+的数据库连接 Pooling 功能,并且避免客户端建立不必要的对象,我使用多个独立的 MTS/COM+数据模块,其中分别放入 TADOConnection、TADOQuery 和 TDataSetProvider 以连接到不同的数据表。这样的设计虽然比直接使用单一的 MTS/COM+数据模块来得麻烦,但是无论在执行效率、系统的延展性以及资源的共享性上都比单一的 MTS/COM+数据模块好得多。

为了更好的设计开发中间件,我对使用 COM+做中间件来开发应用系统需要注意的一些问题做了总结:

- 1、对于客户端而言,应该尽早取得需要使用的 MTS/COM+对象,并且在最后使用完毕之后再释放取得 MTS/COM+对象。
- 2、避免激活不必要的事务 Context。对于 MTS/COM+对象而言,应该让事务管理越晚发生越好,并且在执行完必要的工作之后立刻调用 SetComplete/SetAbort 和 EnableCommit/DisableCommit 方法释放占据的资源。尽量把相关的 MTS/COM+对象放在同一个套件组件中,尽量减少不同套件组件之间 MTS/COM+对象的调用。
- 3、在 MTS 中使用 STA 线程模型的对象,在 COM+中使用 Neutral/Rental 线程模型的对象。
- 4、尽量利用 MTS/COM+对于数据库的 Pooling,而不要使用主从架构的方式来利用数据库 Pooling。通过使用这些方法在开发过程中我们大大改进了系统的执行效率。

数据层使用一台数据库服务器,通过企业局域,分布在仓储公司等单位的 workstation 利用客户端软件进行数据录入及维护工作,在各个二级分厂、单位的客户端都可以通过内部网进行查询浏览。我们采用的是 SQL Server 数据库,功能强大,使用方便,完全满足系统的要求。

最后通过项目组成员的努力,我们按期完成了任务。采用这样的体系结构,完全满足要求,并且安全可靠、容易维护、扩展方便、结构模块化、易操作。经过用户一段时间的使用,基本上没有什么问题。在后期的维护中,我们对中间件和客户端分别做了少许修改,但二者之间没有互相影响,这些充分体现出多层结构的优越性。当然在其中也存在一些问题。比如在系统的执行效率和延展性上,给人的感觉是多层应用系统的执行速度有点慢,还有组件的设计上还存在些问题。这些问题在了解 COM+的设计方法和系统需求后才能更好的解决。

论软件三层结构的设计

摘要

随着市场的建立和发展,卫生行业面临了很多问题,一些制约卫生事业发展的矛盾和问题日益显现,因此,国家卫生部要求各医院采用信息化管理。前不久,我所在的部门承担了一个医院管理系统的设计和开发,医院希望以此来转变医院现有的运行机制,提高服务质量。该系统除了目前常见的计费系统、电子病历外,还包括门诊医生工作站、住院医生工作站、护士工作站等分系统。考虑到需要通过 Intranet 实现功能,并有部分的 Internet 功能,本项目平台最后采用了 Java 平台。我在项目中主要负责项目的前期规划,即选择合适的开发方案,并建立部分的数据流,在系统实施过程中推动其顺利前进。此系统开发成功后投入运行,获得医院相关工作人员的好评。

正文

前不久,我所在的部门承担了一个医院管理系统的设计和开发,医院希望以此来转变医院现有的运行机制,提高服务质量。客户是一个市级医院,医院很早就开始从事信息化管理,但主要是针对计费这一块,后来,对其进行了改进,加入对病人电子病历的管理和采集,这样当病人二次就诊时,可以很容易地得到病人的既往病史。但随着系统的运行,院方希望对现有系统进行改进,为了更好得为病人服务,医院考虑加入一些其它的分系统,比如门诊医生工作站、住院医生工作站和护士工作站等等。因此我所在的部门承接了该 HIS 的开发,开发的成果是一个典型的 Java 技术在 Intranet 上的应用。

在开发前期,首先要设计出详细的系统功能规范,这一部分所花费的时间很少,因为卫生部在 2002 年曾经颁发了一个有关医院管理系统功能规范的通知,我们参考了该规范,很快确定了各分系统以及每个分系统的基本功能。但在选择合适的系统平台上有一番讨论,考虑到医院原有系统在有些地方运行良好,是否有必要将原有系统淘汰重新设计,另外新的分系统到底采用何种平台结构也是需要考虑的问题。

医院原有的计费系统和电子病历系统数据流向范围比较固定,主要集中在交费处和挂号处,一旦引入了新的系统,必然要将数据流向医院的各个部门。医院的 Intranet 已经实施,因此首先考虑采用 B/S 架构体系,旧系统的数据模型尽可能保留。在系统的软件平台上,我们考虑使用 Java 平台,可以让数据在整个系统安全、有效地流动;另外现在也有很多的 HIS 系统可供我们参考,虽然往往是单机版的系统,但其中的数据模型有很好的参考价值。医院的现有网络系统和操作系统多种多样,这就要求我们选择的软件平台必须具有开放性、平台无关性。而在不同的系统上安装相应的 Java 客户端虚拟机并不困难,最后,在项目组的讨论和征求客户意见下,项目组采用了此方案。

在项目中,我们这样设计 Java 架构系统,将系统分为三层:

(1) 表示层采用 Jsp 实现页面输出,这也是用户直接访问层,表示层接受来自网络浏览器的 HTTP 请求,然后返回给客户端浏览器可以显示的 HTML 页面;

(2) 中间件层用 Java 实现对数据库的访问,考虑到数据的分布特点,我们使用了数据库连接池技术;

(3) 数据库层用 SQL Server 实现数据库的管理和存储过程。

JSP 以其执行的高效性和使用的方便性,已成为近年来大家首选的因特网开发技术,JSP 是一种页面开发技术,它以 Java 为其服务器端语言,结合 Java Script 作为其客户端语言,能方便地实现页面的表示。选择 Jsp 作为前台语言,是考虑到它的平台无关性,能够兼容其他的操作系统和数据。利用 Jsp 可以将 HTML 文件很方便地发送到客户端 Web 浏览器,同时也可以支持一些非 HTML 格式的文档发送。我们在网站中的用户层页面设计中,广泛地参考了现有的一些成功案例,在节约设计和开发的费用的同时也得到了用户的认可。比起单纯的 Servlet 技术,Jsp 在页面元素上也更为丰富,在这里,我们为今后客户页面定制做了部分尝试,设计了一个比较简单的模块,用户可以通过选择控件来动态生成页面提供打印,当然在这部分我们设计中,可提供的数据库字段是固定有限的,灵活性没有很好体现。

我们大量工作主要放在中间件层,由于系统中的医生工作站分系统是主要部分,因此数据库的负载成为一个不得不考虑的问题。综合考虑,我们决定采用数据库连接池技术,因为基本上每个医生都对应一个客户终端,无论是病人资料的查询或录入,或者医嘱的处理,都涉及到对数据库的频繁读写,服务器对于数据库连接的频繁打开和关闭必然导致性能下降。一方面,我们预先考虑数据库的连接量,在系统初始阶段建立相应的存储空间,当数据库连接打开和关闭时都对该连接池进行处理;另一面,我们也使用了高速缓存技术,对某些固定的 SQL 查询结果,例如药品查询、药性禁忌等,将结果采用缓存存储,以加快对数据库的访问,降低了服务器端的负载,提高性能。值得一提的是,为了尽可能得避免纠纷,医院要求采用医生签名制度,我们设计了电子签名,采用加密算法保证各环节产生数据的有关人员不能抵赖。

数据库层我们选择了 SQL Server,程序员比较熟悉此平台的开发和设计。在前期,我们到医院做了大量的走访工作,了解整个数据流,同时广泛参考现有的系统,虽然对医院行业不是很熟悉,但数据层开发遇到的问题的并不是很大。在概要设计阶段,我们使用了一些计算机辅助开发工具,这也加快了详细设计的进程。比如使用了 Sybase 的 PowerDesigner 工具,在概要设计中规划了 E-R 图和各个分系统以及分系统之间的数据流图,然后让工具直接生成后台数据库中的基本表结构,大大提高了开发效率。

整个项目实施完成后,医院相关工作人员反映良好,但其中也暴露出了一些问题,值得改进。

首先,用 JSP 编程时容易导致系统信息的扩散。如果有人恶意攻击服务器,程序执行将出现异常。这时,就会在页面上打印出相应的错误信息。这些信息会暴露出这台服务器的路径信息。这个漏洞往往会被别人利用,程序员对此也一头雾水,下一步我们考虑从服务器入手,采用通用的异常说明界面,解决该问题。

其次,在设计上,可以使用更多的辅助开发工具和建模工具,比如 Rational Rose,可以利用它的代码自动生成功能,来大大提高开发速度;在用户界面定制方面,我们希望通过不断的实践,加强其灵活性,尽可能把可供选择的字段扩大到整个数据库。

最后,大量使用 Java 技术,势必对服务器的负载过大,在今后的开发中,除了现有的数据库连接池和缓存技术,我们考虑使用更多的数据库脚本来替代部分 Java 代码,来高速实现逻辑,相较而言,数据库脚本的执行速度较有优势,结合结果缓存,应该对服务器的性能有较大的提高。

上面是我在今后的系统设计和开发中需要注意和加以改进的地方,也是我今后应该努力的方向。

论软件三层结构的设计

摘要

我所在的单位是国内主要的商业银行之一，作为单位的主要技术骨干，2003年1月，我主持了远期结售汇系统的开发，该系统是我行综合业务系统 XX2000 的一个子系统，由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 结构作为该系统的软件体系结构，在详细的设计三层结构的过程中，我采用了字符终端为表示层，CICS TRANSATION SERVER 为中间层，DB2 UDB 7.1 为数据库层，并采用了 CICS SWITCH 组，并行批量的办法来解决设计中遇到的问题，保证了远期结售汇系统按计划完成并顺利投产，我设计的软件三层结构得到了同事和领导的一致认同和称赞。但是，我也看到在三层结构设计中存在的一些不足之处：比如中间层的负载均衡算法过于简单，容易造成系统负荷不均衡，并行批量设计不够严谨，容易造成资源冲突等。

正文

我所在的单位是国内主要的商业银行之一。众所周知，银行的业务存在一个“二八定理”：即银行的百分之八十的利润是由百分之二十的客户所创造。为了更好地服务大客户，适应我国对外贸易的蓬勃发展态势，促进我国对外贸易的发展，2003年1月，我行开展了远期结售汇业务。

所谓的远期结售汇就是企业在取得中国外汇管理局的批准后，根据对外贸易的合同等凭证与银行制定合约，银行根据制定合约当天的外汇汇率，通过远期汇率公式，计算出交割当天的外汇汇率，并在那天以该汇率进行成交的外汇买卖业务。远期结售汇系统是我行综合业务系统 XX2000 的一个子系统，它主要包括了联机部分、批量部分、清算部分和通兑部分，具有协议管理、合约管理、报价管理、外汇敞口管理、帐务管理、数据拆分管理、报表管理、业务缩微和事后监督等功能。

我作为单位的主要技术骨干之一，主持并参与了远期结售汇系统的项目计划、需求分析、设计、编码和测试阶段的工作。由于银行系统对安全性，可靠性，可用性和响应速度要求很高，我选择了三层 C/S 结构作为该系统的软件体系结构，下面，我将分层次详细介绍三层 C/S 软件体系结构的设计过程。

1、表示层为字符终端。我行以前一直使用 IBM 的 VISUALGEN 2.0 附带的图形用户终端来开发终端程序，但在使用的过程中，分行的业务人员反映响应速度比较慢，特别是业务量比较大的时候，速度更是难以忍受。为此，我行最近自行开发了一套字符终端 CITE，它采用 VISUAL BASIC 作为开发语言，具有响应速度快，交互能力强，易学，编码快和功能强大的特点，在权衡了两者的优点和缺点之后，我决定选择字符终端 CITE 作为表示层。

2、中间层为 CICS TRANSATION SERVER(CTS)。首先，我行与 IBM 公司一直保持着良好的合作关系，而我行的大部分技术和设备都采用了 IBM 公司的产品，其中包括了大型机，由于 CICS 在 IBM 的大型机上得到了广泛的应用，并在我行取得了很大的成功，为了保证与原来系统的兼容和互用性，我采用了 IBM 的 CTS 作为中间层，连接表示层和数据库层，简化系统的设计，使开发人员可以专注于表示逻辑和业务逻辑的开发工作，缩短了开发周期，减少开发费用和维护费用，提高了开发的成功率；其次，对于中间层的业务逻辑，我采用了我行一直使用的 VISUALAGE FOR JAVA 作为开发平台，它具有简单易用的特点，特别适合开发业务逻辑，可以使开发人员快速而准确地开发出业务逻辑，确保了远期结售汇系统的顺利完成；

最后，由于采用了 CTS，确保了系统的开放性和互操作性，保证了与我行原来的联机系统和其他系统的兼容，保护了我行的原有投资。

3、数据层为 DB2 UDB7.1。由于 DB2 在大型事务处理系统中表现出色，我行一直使用 DB2 作为事务处理的数据库，并取得了很大的成功，在 DB2 数据库的使用方面积累了自己独到的经验和大量的人才，为了延续技术的连续性和保护原有投资，我选择了 DB2 UDB7.1 作为数据层。

但是，在设计的过程中我也遇到了一些困难，我主要采取了以下的办法来解决：

1、CICS SWITCH 组。众所周知，银行系统对于安全性，可靠性，可用性和响应速度要求很高，特别是我行最近进行了数据集中，全国只设两个数据中心，分别在 XX 和 YY 两个地方，这样对以上的要求就更高了，为了保障我行的安全生产，我采用了 CTS SWITCH 组技术，所谓的 CICS SWITCH 组，就是一组相同的 CTS，每个 CTS 上都有相同的业务逻辑，共同作为中间层，消除了单点故障，确保了系统的高度可用性。为了简化系统的设计和缩短通讯时间，我采用了简单的负载均衡算法，比如这次分配给第 N 个 CTS，下次则分配给第 N+1 个 CTS，当到了最后一个，就从第一个开始；为了更好地实现容错，我采用了当第 N 个 CTS 失效的时候，把它正在处理的业务转到第 N+1 个上面继续处理，这样大大增加了系统的可用性，可以为客户提供更好的服务；此外，我还采用了数据库连接池的技术，大大缩短了数据库处理速度，提高了系统运行速度。

2、并行批量。银行系统每天都要处理大量的数据，为了确保白天的业务能顺利进行，有一部分帐务处理，比如一部分内部户帐务处理，或者代理收费业务和总帐与分户帐核对等功能就要到晚上批量地去处理，但是，这部分数据在数据集中之后就显得更加庞大，我行以前采用串行提交批量作业的办法，远远不能适应数据中心亿万级的数据处理要求，在与其他技术骨干讨论之后，并经过充分的论证和试验，我决定采用了并行批量的技术，所谓的并行批量，就是在利用 IBM 的 OPC (Tivoli Operations, Planning and Control) 技术，把批量作业按时间和业务处理先后顺序由操作员统一提交的基础上，再利用 DB2 的 PARTITION 技术，把几个地区分到一个 PARTITION 里面分别处理，大大提高了银行系统的数据处理速度，确保了远期结售汇系统三层结构的先进性。在并行批量的设计过程中，我考虑到批量作业有可能因为网络错误或者资源冲突等原因而中断，这样在编写批量程序和作业的时候必须支持断点重提，以确保生产的顺利进行。

由于我软件三层结构设计得当，并采取了有效的措施去解决设计中遇到的问题，远期结售汇系统最后按照计划完成并顺利投产，不但保证了系统的开发性、开放性、可用性和互用性，取得了良好的社会效益和经济效益，而且我的软件三层结构设计得到了同事和领导的一致认同与称赞，为我行以后系统的开发打下了良好的基础。

在总结经验的同时，我也看到了我在软件三层结构设计中的不足之处：

首先，负载均衡算法过于简单，容易造成系统的负荷不均衡：由于每个业务的处理时间不一样，有的可能差距很远，简单的顺序加一负载均衡算法就容易出现负荷不均衡，但是如果专门设置一个分配器，则增加了一次网络通讯，使得系统的速度变慢，这样对响应速度要求很高的银行系统来说也是不可行的，于是我决定采用基于统计的分配算法，即在收到请求的时候，根据预先设定的权值，按概率，直接分配给 CTS。

其次，由于批量作业顺序设计得不够严谨等各种原因，容易造成资源冲突：在远期结售汇系统运行了一段时间之后，数据中心的维护人员发现了，系统有的时候会出现资源冲突现象，在经过仔细的分析之后，我发现，由于每天各个业务的业务量大小不一样，顺序的两个作业之间访问同一个表的时候便会产生资源冲突，另外，在 OPC 作业运行的过程中，操作员提交的其他作业与这个时间的 OPC 作业产生也有可能产生资源冲突。对于第一种情况，可以在不影响业务的情况下调整作业顺序或者对于查询作业运用 DB2 的共享锁的技术，而第二种情况则要制定规范，规定在某时间断内不允许提交某些作业来解决。为了更好地开展系统分析工作，我将在以后的工作实践中不断地学习，提高自身素质和能力，为我国的软件事业贡献自己的微薄力量。

论软件开发平台的选择与应用

摘要

本文讨论选择新软件开发平台用于重新开发银行中间业务系统。银行中间业务系统是指银行通过与企事业单位、机关团体的合作,为客户提供金融服务的系统。X 省农行银行的原中间业务系统软件开发平台是以 UNIX 系统为操作系统,使用的数据库是 Sybase,采用二层的 C/S 结构,使用 DB—Library, T—SQL 编程。随着业务的不断发展和软件开发维护工作的剧增,该软件开发平台表现出工作效率低,开放性差,开发出的产品不易管理等突出性的问题。为了解决原软件开发平台的不足之处和基于该银行长远发展目标的要求,我们引入新的软件开发平台 OSP PrePbranch。在文中阐述了选择软件开发平台的原则:要求开放性好,可复用性高、开发出的软件易于管理、风险可控、技术能与发展主流趋势相一致并易于掌握,并总结了应用新软件开发平台开发银行中间业务系统所带来的优势。

正文

银行的中间业务系统是指银行依托自身的电子化平台、网点网络和人员等资源,通过与企事业单位、机关团体的合作,为客户提供代收代付、资金转帐结算、业务代理和咨询、增值理财等金融服务的平台。X 省农业银行的中间业务系统主要包括由五大类业务组成,即批量业务,如代发工资、国库统发工资、批量代收代付;单笔缴费业务,如柜台代收学费、行政管理费等;通信类业务,主要是指要与外单位进行实时通信的业务,如固定电话缴费系统,手机缴费系统等;增值服务业务,如代客理财、银证资金转帐和结算;其他业务,如业务咨询和代理等。

该行原中间业务系统是 C/S 结构,采用 DB—Library/C 编程,使用的数据库是 Sybase,以 UNIX 作为操作系统。该开发平台在开发中间业务系统是工作效率比较低,开发周期比较长,复用性差。针对上述开发平台的实际情况和存在的主要问题我们引入新的国内某公司的基于中间件技术的软件开发平台“OSP PrePbranch”用于重新开发该银行的中间业务系统。在整个项目期间我参与了软件开发平台的选型,系统的分析,开发和设计,测试的工作。

在选择 OSP PrePbranch 平台时,我们主要遵循了以下原则:利用选择的平台开发出的新系统能和基础业务系统有机集成的原则,选择的开发平台易于掌握的原则,选择的开发平台开放性好、复用率高的原则,选择的平台易于管理的原则。

应用新软件开发平台开发出来的系统要能够和基础业务系统有机集成是引入软件开发平台的基本条件,它也是选择软件开发平台的原则之一。OSP PrePbranch 软件开发平台是一个独立的开发平台,使用该软件开发平台开发的中间业务系统能方便通过基础系统的接口直接调用,不需要另外开发接口调用中间业务系统,而且 OSP PrePbranch 厂商进行了现场演示和提供了正式的书面文档。这样,通过简单的调用能实现基础业务系统和中间业务系统的有机集成。

选择的开发平台要遵循易掌握性原则。引入一项过于复杂或难以掌握的软件开发平台是毫无意义的。基于中间件技术的 OSP PrePbranch 软件开发平台采用的操作系统是 unix,使用 C/C++和嵌入式的 T—SQL 语言为开发工具。C/C++、SQL 它们都是该银行软件工程师比较熟悉的编程语言,因此该软件开发平台易于掌握。使得该银行选用该软件开发平台在较短的时间就能掌握它,应用它。

选择的开发平台要遵循开放性好、复用率高的原则。OSP PrePbranch 系统提供的每个原子交易都定义了规范的接口说明,调用时入参、出参的类型、个数、顺序方面都有明确的定义,使用的场合范围;如文件的上传/下传服务就有详细的定义,上传文件名,传输方式,存放的目录等方面都有详细的说明。在开发时只要根据实际情况和应用说明文档就可以实现透明调用了。OSP PrePbranch 软件开发平台在应用层提供了网关(GateWay)服务,使得中间业务系统与协作单位的通信不需要银行开发人员自己开发,而由 OSP PrePbranch 直接提供;OSP PrePbranch 还提供了通信的包格式的定义,通信连接方式的定义,通信格式加包头,通行打包,通信格式去包头,通信格式解包等原子交易。

除了上述 OSP PrePbranch 系统本身具有良好的开放性,提供大量可复用的原子交易外。在此次开发新系统时根据银行中间业务系统的特点还自制了大量可复用的原子交易,如帐户的入帐、帐户明细的查询、帐户性质的判断、卡折之间的转帐、各人帐户和对公帐户之间的转帐等,把这些模块加载到 OSP PrePbranch 系统的构件库中,与 OSP PrePbranch 本身的原子交易一起提供服务。对自制的原子交易也提供了规范的接口说明,参数调用说明,应用场合等规范性文档。它们和 OSP PrePbranch 软件开发平台本身直接提供的原子交易对于所有的程序员是透明的,程序员使用时只要查阅规范文档就可以直接使用,而根本不需要了解它们的具体实现的细节。有了这的开发平台,在开发新系统时只要根据不同的业务品种编写必需银行核心功能模块,然后从 OSP PrePbranch 构件库中选择所需的原子交易就可以快速开发出金融交易。这样提高了软件的复用率,该变了过去重复开和重复测试的大量工作,极大地缩短了开发地周期,大大加快了研制进程。因此,无论 OSP PrePbranch 软件开发平台本身,还是后继的开发都保持了良好的开发性,较高的复用率。

选择的开发平台要遵循易于管理的原则。该银行中间业务系统改造之前客户端前台软件升级时,采用手工作业方式,工作量大,不便于管理。新系统中把各网点终端信息收集在一起,保存中间业务系统应用数据库中;若软件需升级更新时,把升级软件放在中间业务平台的指定目录下,每天网点开启中间业务系统时,前台软件自动执行版本更新升级程序,保证网点及时、自动地从后台取回最新版本软件并投入生长运行。因此,新系统极大地减少了软件升级所带来地工作量,使得中心机房可对各网点运行的软件进行统一升级、版本控制、集中管理、方便管理。

由于选用的软件开发平台 OSP PrePbranch 是基于中间件技术的,它由前台开发子系统,主控守护进程 ubridge,后台开发子系统和数据库连接与管理子系统组成的。中间件技术是当前信息技术发展的一种主流技术,因此 OSP PrePbranch 软件开发平台在相当长的一段时期内保持技术领先的优势。

OSP PrePbranch 软件开发平台具有良好的开放性,复用性高,它所采用的是先进的中间件技术。特别是 OSP PrePbranch 系统的开放性好,可复用性高方面在应用该软件开发平台开发中间业务系统时,可以方便的使用 OSP PrePbranch 提供的开放的接口,大量现成的原子交易,在具体开发时只要根据业务需要编写必须的核心代码就可以快速开发出金融交易,缩断了软件开发的周期,提高了工作的效率,提高了软件的复用性,加快了研制开发的进程。使得该银行能在较短的时间内开发出新的金融产品,提高了银行的市场竞争力。

任何一种软件开发平台的选用都具有一定的风险,特别是在金融业中引入新的软件开发平台要谨慎从事。首先,它是一个基于中间件技术的成熟的产品,在此之前已经在国内金融业有多个成功应用的先例;其次,该软件开发平台大部分是采用 C/C++语言开发的,它易于掌握,因此就不存在选用一个不易掌握或不能掌握的平台;再次,该软件开发平台实施成本低。因此我们从软件开发平台的成熟性,易掌握性,实施成本等多方面进行了论证和考察,它所带来的风险是在可以控制的范围之内的。

当然,我们在选择 OSP PrePbranch 软件开发平台主要是基于该银行的实际情况的需要。在不同的应用领域有不同的选择,如企业应用集成可以选择 EJB 规范的 J2EE 平台,它具有开放性,平台应用的无关性等特点。但是无论那种软件开发平台,它们都应该朝着开放性、平台无关性、可移植性、能够实现异构集成等方向发展和演变,同时它们应该代表着先进的技术发展方向。

论基于构件的软件开发

摘要

本文以我主持的某商业银行交易监控分析系统项目为实例，探讨了作为开发方公司基于构件技术开发项目碰到的问题以及解决的方法。文章首先解释了基于构件技术开发软件的基本概念，认为目前大多数开发单位的产品在存在重复的功能模块，而重复的开发工作，直接导致了项目周期以及成本不必要的增加，针对这一问题，提出了应该及时整理已有的系统，形成企业构件库，针对性的选择构件，从而基于构件开发新的软件项目，在保证软件产品质量的前提下，缩短项目周期和开发成本，最终使企业盈利。

我在项目中担任了开发方的项目经理，自始至终参与了整个项目的建设，自 2008 年 3 月项目启动至 2008 年 10 月验收历时 8 个月，系统至今运行稳定，取得了客户的一致好评，项目能够保证质量的前提下迅速完成，最终节约了成本，很大程度上得益于基于构件开发软件的应用。

正文

目前，银行间竞争已日趋激烈，降低银行的服务成本，提高对客户的服务质量，已成为银行竞争的关键。随着服务种类的增多、日交易量的增大，针对这些交易数据进行有效的管理变得迫在眉睫。首先，需要将各种交易实时监控，保证能够对异常交易进行及时处理，提高客户满意度。其次，需要对对这些交易数据进行统计分析支持高层管理人员的决策，提高银行的竞争力。基于上述需求，2008 年初，某市商业银行招标建设一个交易监控分析系统，我所在的单位中标后，我有幸参加了该项目的建设，承担了开发方的项目经理一职。

交易监控分析系统主要包含 4 个子系统：基于 B/S 结构的管理平台，处理多种不同报文格式交易信息的数据转换平台，基于 B/S 结构的联机分析处理平台，数据维护子系统。

众所周知，基于构件的软件开发是一种自底向上的、基于包装好的构件来构造应用系统的方法。它主要包含构件的检索与提取，理解与评价构件，修改构件，组装构件，应用部署等工作。

目前软件行业间的竞争程度趋于白热化。建设方更倾向于选择已经有成功案例或者有类似项目成功案例的开发方。我认为，对于开发方来说，大多数开发单位的产品在存在重复的功能模块，而重复的开发工作，直接导致了项目周期以及成本不必要的增加，如何利用已有的软件项目构造新的系统，而不仅仅是将其作为投标的筹码变得越来越重要。针对这一问题，我在采用及时整理已有的系统，形成企业构件库，针对性的选择合适的构件，加大对已修改构件的管理力度等方法，有效的实施了基于构件的软件开发工作。

形成构件库是基于构件开发软件的前提。我们在已经成功实施的项目中，抽取一些共有的模块作为单独的组件，封装其内部操作，对外提供一致的调用接口。我们分析发现，一些常用的模块例如登录模块，只需要很少的改动就可以复用到新的系统中，对于一些看似不同的模块，例如查询银联交易流水，查询 POS 交易流水，查询设备状态信息等，如果对数据库操作的方式去分类的话，都离不开增、删、改、查四个操作，因此，我们对表示层，业务层，数据层进行了面向接口的整理工作，封装成一个组件。使用 Hibernate 技术管理数据库操作，对于表示层需要列表显示的对象，通过参数的方式传递到业务层。在本项目中，我们的工作就集中在了 Model 层对象设计上，建立 Model 层对象与数据库的映射关系之后，直接使用构件组合待查询的交易信息，极大的减少了系统的重复性开发工作。

针对性的选择构件是基于构件开发软件的关键。构件的选择有多种途径，一是从构件库中提取符合要求的构件，二是从市场上购买现成的构件，三是根据特殊应用需求开发。在本次项目中，我们选择的构件来自于第三方和企业构件库。

考虑到银行的前置机，核心主机都是 Unix 操作系统，而部分外围服务运行与 Windows 或 Linux 平台，所以我们采用了与系统无关的 J2EE 平台构架。

关于系统与后台主机的通讯，DCOM 和 CORBA 都是适合于服务器—服务器间通信的成熟规范，由于选定了开发平台，我们选择使安装 Java 自带的 ORB 进行通信。后台主机将处理的交易信息使用 UDP 协议发送到监控系统，从而实现了交易数据的获取。获取的报文需要经过转化为监控平台统一格式才可以使用，因此，我们需要数据转换服务，如果选择第三方的构件，面临的问题是无法对其进行源码级的修改，所以我们选择自主开发新的构件用于数据转换服务，平台处理 XML 格式的数据，数据转换服务实现 XML 报文与银行后台报文的转换。

几乎所有的信息管理系统都包含了登录模块，这一块我们直接从构件库中提取，业务层，数据层都不需要修改，只需要更换表示层中的界面图片即可。系统使用验证客户端 IP 的方式，限制可访问系统的客户端，避免非法连接。用户权限使用角色，用户组等方式进行管理，便于权限分配。这些组件都是我们经过长期使用并且不断完善的模块，可直接从构件库中提取。

加大对已修改构件的管理力度是为今后的构件开发软件项目做好准备。在每个项目中，我们都或多或少的生成一些新的功能模块。在本次项目中，对数据维护子系统，就是新开发的一个组件，它负责定期进行对应用系统中的数据进行抽取，清洗工作，并将操作结果存储到数据仓库中，用于支持决策分析，其接口使用 XML 定义。对于这样的新开发的构件，经过测试之后，我们将其按照企业目前的标准进行严格定义，形成相关的文档，一起录入到企业构件库中，以便复用于后续的项目。对于一些修改过的构件，如数据转换模块，我们对它也进行了详细的描述，指定版本号以及各个版本之间的差异，便于以后针对不同的情况使用不同版本的构件。

在项目中，我们也发现了一些不足之处，例如要使已经完成的构件能有效的支持项目开发，对构件的管理与维护，员工对构件的理解程度都是不容忽视的问题。通过本项目的实施，我们了解到了仅仅是对企业软件构件进行严格定义并且有效的管理还不能保证员工对其理解程度一致，在项目开发过程中，开发人员通常从构件库中检索获取构件，而随着构件的增加，不同的构件也可能存在部分的功能冗余，要在合适的系统上使用合适的构件就对开发人员提出了较高的要求。针对这一问题，我们除了采用培训员工的办法，还定期以知识竞答的方式组织员工进行知识竞猜，创造一种积极的氛围，促使员工对企业构件库有较高等度的理解。

综上所述，经过整个项目组精心准备和严密实施，项目如期完成，自 2008 年 3 月项目启动至 2008 年 10 月验收历时 8 个月，系统至今运行稳定，得到了用户的一致好评。回顾项目的实施过程，我体会最深的是，随着基于构件技术开发软件的成功实施，我们在享受它带来的便利的同时，也要注重企业内部的构件积累。从通用性的角度来看，企业新开发的构件不如经过市场验证的第三方的成熟构件，从行业的角度来看，企业开发的构件能够满足其业务领域的大部分开发工作，这正是第三方构件无法做到的。感觉自己在结合第三方构件与企业内部构件进行软件开发的能力还有待加强，我将在后续的项目中努力做好这些工作。

论软件开发平台的选择和应用

摘要

本文从一个行业 MIS 系统的开发实践,讨论了软件开发平台的选择和应用。首先,作者从项目的实际情况确定了软件开发平台的一些原则:技术成熟兼一定先进性、高效集成的开发工具、开发人员熟练掌握等,随后就系统平台、软件开发平台、数据库平台的选择作了详细论述。之后,作者就开发过程中就保持系统开放性,对数据导入导出、与 P3 软件集成、WEB 查看权限采取了相关措施;就保持系统先时性,提到了多种软件技术合成及 VPN 技术两项措施。最后,作者对近期商业应用软件开发平台的主流--微软的 .net 及 J2ee 进行了介绍,比较了其优缺点,对今后本部门在软件开发平台的发展方向作了一定的评估。

正文

我公司是大型电源建设项目的专业建设公司,曾以总承包方式承建设了多个电厂,在工程建设过程中逐步建立了一系列完整、科学的工程管理体系,与此相应的是逐步建立电力建设项目工程管理信息系统(简称电建 MIS),原版本的不足之处是应用模块不多,且多以简单文本为主。

2002 年,浙江省某规模为 4 台 60 万千瓦机组的发电厂二期工程建设上马。该工程作为一个大型能源投资项目,将有力地拉动地方经济的增长,并且将为浙江和华东地区的经济和社会发展提供强大的能源支持和保障。工程由多方共同出资建设,由我公司承担工程总承包建设任务。借此工程建设的契机,我公司决定重新开发电建 MIS 3.0 版。主要模块拟包括办公自动化、施工总平管理、合同管理、物资管理、质量管理、安全管理、图档管理、公用信息管理、综合查询(包括 WEB 查询)、系统维护,基本涉及我项目部的各个职能部门。

软件采用二层 C/S 与三层 B/W/S 相结合的方式。其中,B/W/S 方式用于查询和浏览,C/S 方式用于主要数据录入和维护,采用 C/S 和 B/S 结合的混合体系结构,较好地满足系统功能的需求,并符合可持续发展的原则,使系统有较好的开放性和易扩展性。

软件采用我公司和外部软件公司合作开发的形式,版权由我公司独家所有。因多种原因,与我们合作开发的软件公司有 2 个。我是软件开发的负责人。

在开发平台的选择上,我们考虑了本 MIS 项目的特点:该项目要求在规定的时间内(6 个月)内完成,并要求有较高的质量,且要为以后的进一步开发提供基础。

基于此,我们对选择开发平台的原则达成共识:

- 一、技术上成熟且具有一定的先进性;
- 二、有高效、集成的开发工具;
- 三、应为开发人员熟练掌握。
- 四、软件平台提供商对该软件平台的后续支持能力。

首先,在系统平台的选择上,有两点考虑:1、公司现有的各级软件系统都是基于微软 Windows 系列平台的,且公司没有在日后使用其它平台的打算;2、微软的 Windows 平台完全能满足开发、运行该类 MIS 系统的要求。因此确定新开发的 MIS 系统也就是基于此平台的。

对二层 C/S 在开发平台的选择,考虑采用微软的 VB6.0 或 Sybase 的 Powerbuilder8.0。前者是公司前两版 MIS 系统的开发平台,后者则更为现有的开发人员所熟悉。两者在开发上技术都很成熟。在开发界面的亲和性上,VB 做得较好;而对从数据库设计到编程的全过程而言,Powerbuilder 具有更好的集成性,用其集成工具 Powerdesign 设计的逻辑数据库可以很方便地生成物理数据库。从当时情况看,无论是微软,还是 Sybase,对各自软件平台的后续支持能力都较好。最后综合各种情况,选择了 Powerbuilder,其最重要的一点是开发人员的熟悉程度。

对三层 C/S 在开发平台的选择,考虑采用的比较方案是微软的 IIS+ASP 组合及 Apache+PHP 组合。开发人员的经验还是在 IIS+ASP 上,对于 Apache+PHP 少有实践,基于此,选用第一种方案。

在数据库平台的选择上,有微软的 SQL Server 和甲骨文的 Oracle 可以选择。前者是前两版 MIS 的数据平台。而 Oracle 数据库是目前公认的最优秀的大型数据库,和微软的 SQL SERVER 相比,它具有更好的稳定性和安全性---这通常是企业用户最关心的两个特性。开发人员对两种数据库都有相应的开发经验。最终选用了 Oracle 数据库平台。

在开发中考虑了系统的开放性和先进性。在开放性方面,有以下措施。

1. 针对原有大量已积累数据(大多是 EXCEL 格式的)的导入,及系统中数据可能的导出,设计了专门的导入导出模块。使得操作人员一方面从繁重的初始数据录入工作中解脱出来,一方面随时可以导出系统中的数据作个别分析。并预留了与公司总部及其它协作单位进行数据交换的接口。

2. MIS 系统与 P3 软件(大型项目管理软件)要求做到集成。为此设计了专门的模块,通过 ODBC 方式进行后台的数据交换,使得两套软件做到无缝集成:在 P3 软件中操作的结果可以在 MIS 软件中反应出来,而在 MIS 软件中相应模块中输入的数据也能在 P3 软件中反应出来。这样做的好处是大多数工程师无须操作 P3 软件即可输入有关工程进度的数据,并通过 P3 软件的功能自动计算出相应的数据并得到横道图;而少数工程师仍可在 P3 环境工作,而工作结果同样反应在 MIS 中。

3. B/S 部分的使用对象不仅仅局限于项目部的员工,还包括公司本部人员及众多协作单位的人员。而由此相关的权限及安全问题必须考虑,即不同身分的人员能查看的权限是不一样的。ASP 在此方面的功能相对较弱,为此在 B/S 的服务器端专门设计了基于 COM 技术的身份认证 ActiveX 组件。

在先进性方面,有以下措施。

1. 采用了多种软件技术的合成,如在 MIS 系统中合成了工作流技术,在工作流技术中调用了微软 OFFICE 的宏技术。工作流技术上考虑与系统更好地集成,采取了基于数据库的方案,自行开发了相关的模块。其最终效果是在 MIS 中实现了办公自动化的功能,极大地方便了使用者。

2. 采用了先过的 VPN 技术,使得部分外出员工可以通过 Internet,借助 VPN 访问内部的 MIS 系统,从而完成远程办公。

系统运行后收到了良好的效果,用户对该软件的满意程度较前两版有较大的提高。该系统随后应用于多个电厂建设项目的管理中。

近期商业应用软件开发平台的主流越来越集中在微软的 .net 及 J2ee,其共同特性是更多地支持面向 Internet 的开发。.net 的优势是支持多种开发语言,开发人员容易掌握,产品集成好,总体成本较低,不仅在开发的时候容易介入,在日后维护时也更能掌握主动权,其缺点是事实上只支持 Windows 平台。J2ee 的优点是跨平台,可以选择多家公司的相关产品,但开发相对复杂,且只支持 JAVA 语言。Powerbuilder 虽然有最新的面向分布式及三层 B/s 结构的新版本推出,但终不及前两者。对于象我们在非软件企业的 IT 部门,在开发新的软件系统时,如采取自行开发或与其它软件公司合作开发的方式,.net 应该是一个比较好的选择。

论软件的可维护性设计

摘要

2008年3月1日至12月20日，我参加了“数据安全访问平台”项目的开发，担任系统分析员的工作。该项目是某行业用户“数据中心二期”建设的主要内容，目标是：建立数据统一访问接口及其使用标准，规范、约束和审计数据应用访问数据库的行为，对数据应用提供强制审计的技术手段。

由于系统交付后，存在较长维护期，同时系统存在升级与扩展的情况，因此本项目对系统的可维护性设计要求较高。本文结合作者实践，讨论了从软件设计上提高可维护性的方法和措施：通过模块化设计方法和提高设计文档质量，改善软件的可理解性；通过提供测试接口和采用测试框架工具，改善软件的可测试性；通过动态库加载和针对接口编程的方法，提高软件的可扩展性。最后分析了采用方法的效果。

正文

一、项目概述

2008年3月1日至2008年12月20日，我参加了“数据安全访问平台”项目的开发，担任系统分析员的工作。“数据安全访问平台”是某行业用户“数据中心二期”建设的主要内容。在一期建设中已建成数据的统一存储和统一分发框架。但主要存在以下问题：无法获得应用用户对数据库的操作日志；开发人员对数据库的使用不规范，查询的结果集过大，导致数据库的性能大幅下降；应用直接使用数据库的登录数据库，存在着一定的安全隐患。“数据安全访问平台”的目标是：建立数据统一访问接口及其使用标准，规范、约束和审计数据应用访问数据库的行为，对数据应用提供强制审计的技术手段。

该项目具有较高的业务需求风险和技术风险。由于没有成熟系统做为参照，该项目需求不是很明确，而且系统涉及甲方多个利益相关方，各方对系统的安全和审计功能、运行维护、可靠性、性能和易用性有者不同的观点，某些观点之间还存在冲突。同时系统作为“数据中心”的基础设施之一，所有的应用系统都要通过本系统完成数据库访问。系统的可靠性和性能直接影响到应用系统的正常运行。整个系统分为6个子系统，包括JDBC驱动封装子系统、ADO.Net驱动封装子系统、WebService接口子系统、管理配置网站、存储子系统（SQL Server 2005数据库，存储配置信息）和监控子系统（数据库网络协议分析与连接控制）。

二、项目中的可维护性设计工作

1、系统的可维护性需求

本系统有较高的可维护性需求。首先，系统作为数据中心应用的基础平台，数据中心的新建应用系统必须依赖于本系统，系统具有较长生命周期；同时合同中规定系统正式上线后，公司需要提供1年的免费维护。其次，本项目中的接口子系统是基于JDBC5.0和ADO.Net2.0实现的，随着JDBC与ADO.Net的升级，接口子系统也需要升级。然后，监控子系统对Oracle和SQL Server数据库的网络包进行解析，由于商用数据库网络包格式不公开，我们在解析时会有遗漏；用户要求，当发现未能解析的数据包时，需要及时添加到监控子系统中。最后，由于没有成熟系统做为参照，该项目开始时需求不是很明确，在开发过程中用户提出了一些新需求，由于工期的限制，经过与用户协商，这些新需求不在本项目中完成；用户准备视系统的运行情况，设立新的项目完成这些剩余需求。因此可维护性是本系统的一个重要质量指标。

为了增加系统的可维护性，减少维护人员理解和修改系统的难度，我们在本系统的设计上，不仅仅关注系统的功能需求实现，而且重视系统的可维护性需求。决定软件可维护性的因素主要包括：软件的可理解性、可测试性和可修改性。在整个系统设计过程中，我们都注重改善系统的可理解性、可测试性和可修改。

2、改善软件的可理解性

本系统涉及的问题域有一定的复杂性，如果将整个问题域的复杂性完全暴露给维护人员，维护人员很难理解整个系统。因此，首先，我们将整个系统划分为功能独立的六个子系统；其次，在各个子系统设计时，我们都采用了模块化的方法，即将内聚性高的业务逻辑合并封装成独立的模块；最后，重视设计文档质量，对设计文档做了内部审核，确保文档清晰准确地描述了设计。

监控子系统是本系统的关键部分之一。公司原有Oracle数据库监控子系统，本项目中需要在原有监控系统中增加SQL Server的监控功能。监控子系统的流程是网络协议包的截获、数据库协议解析、监控审计处理模块。由于原监控子系统只是处理Oracle，其关注功能的实现，三个模块划分的不清晰。例如，使用了过多全局变量在模块间传递数据；功能分布不合理，导致模块之间相互依赖；一个源代码文件包含两个相关功能代码。

本项目中，除了添加SQL Server的监控功能外，我们还改善了子系统的设计，增强各个模块的内聚性，降低模块之间的耦合性。首先，各个功能的相关代码处于不同的代码目录；其次，三个模块都做成了单独的静态库，总控模块负责调度各个库，并且明确了各个模块对外提供的接口，模块之间的调用都通过接口完成。

通过这种方式将系统进行清晰的划分，维护人员可通过对模块接口的学习快速了解子系统的运行流程，当需要时再对某模块做进一步分析。

3、改善软件的可测试性

维护人员对代码进行修改后，必须进行测试，才能保证软件的质量。并且，用户对系统的可靠性要求很高。因此，在软件设计的整个过程中，我们都考虑了测试的问题。

首先，各个子系统的内部模块必须是单向依赖，对出现循环依赖的模块，我们采用调整功能分布，抽取公共模块等方面消除循环依赖。其次，对于接口子系统，我们需要对某些模块内部进行深入的测试，而由于模块接口的封装性，无法直接访问内部数据；对于这样的情况，我们在设计这样的模块时，专门提供了测试接口。最后，开发中采用了CppUnit、NUnit和JUnit测试框架，通过测试框架来组织测试驱动程序。

4、改善软件的可扩展性

监控子系统采用网络监听的方式获取数据库访问的信息，这种方案的优点是不给业务系统增加性能负担，缺点是由于商用数据库协议不公开的，虽然我们经过大量的测试，但是肯定是有遗漏的。

为了当发现未能解析的数据包时，及时添加到监控子系统，同时不重新编译系统，我们进行了可扩展性设计。我们采用了动态库加载和针对接口编程的方法。监控子系统启动后，会加载指定目录下的动态库（符合名字规范），这些动态库都实现了规定的接口。子系统当收到一个网络包后，会依次询问所有的动态库是否能解析该网络包。这样只需要将新开发的动态库拷贝到制定目录，并重新启动监控子系统就可以完成系统的升级。

三、总结

在系统的维护阶段，发现了较严重缺陷2个与8个一般缺陷，同时，发现了监控子系统不能解析的5种协议包。维护阶段的开发人员与开发阶段有了较大变化。由于在系统设计上重视可维护性，软件进行模块化设计，提供了完备的设计文档，维护人员可以较快的定位与解决问题；由于考虑了系统的可测试性，提供了回归测试集，维护人员可以运行回归测试验证软件质量；由于考虑监控子系统的扩展性，维护人员可以及时增加了新的协议包解析功能。综上所述，由于在设计中考虑了软件的可理解性、可测试性与可扩展性，很大程度上提高了系统的可维护性。

论异构数据库的集成

【摘要】

本文讨论了**数据集市项目的数据集成方法与过程。该系统在 2008 年 12 月启动，在 2009 年 5 月正式上线使用。该系统是以 oracle 系统为主要的数据库，同时集成 DB2 系统中的数据。每天的话费清单系在 DB2 数据库中存储，通过 ETL 调度程序把 DB2 中的数据汇总并把结果写入到 ORACLE 数据仓库中。本文首先讨论了建立数据集市项目异构数据库的两个数据库系统的背景以及用户对该项目的需求。接着讨论了使用 Perl 技术来集成两个数据库中的业务逻辑的过程，并说明了该技术在集成过程中出现的问题，如：数据分层，ETL 调度程序改造，以及参数化 SQL 处理等问题。最后讨论了该集成方法的优点和缺点，并对改进该项目提出了优化 Perl 技术的设想。在本次的项目开发过程中，我主要担任了系统分析与设计的工作。

【正文】

数据集市项目是在 2008 年 12 月启动的，我所在的部门主旨了该项目的开发过程，我有幸在该项目中担任了系统分析与设计的工作。该项目的实施是基于以下的背景的。在 2005 年 12 月，我们部门已经组织了开发经营分析系统，该系统已经建立了数据仓库系统，把 BOSS，财务数据，DSMP，MIS 等项目的业务操作数据作为数据源，通过清洗，转换，装载等方式把所有的清单一级的数据全部集中到该系统中了，该经营分析系统是基于 DB2 的数据库开发而成的，把数据仓库分为了 STAGE 层，ODS 层，EDS 层，REF 层。STAGE 层是数据清单的原始数据，ODS 层是经过日汇总的数据，EDS 层是经过维度统一化后的数据，REF 层是维度数据层。每一层数据之间的业务逻辑是通过 ETL 调度程序实现，该程序主要调用的是数据库中的存储过程或者 Shell 脚本完成每层中的数据汇总的过程。对于数据集市项目，我们比较有优势的就是该项目是基于上面的系统进行扩展而成，使用经营分析系统的经验和技能，我们可以快速开发完成。但是该项目由于企业的需求以及供应厂商等问题，最后企业选用了 ORACLE 数据库作为该项目的建立主要数据库，在该项目中必须把所有的 EDS 层的汇总数据记录在以 ORACLE 的主的数据库中以供企业的每个地市使用。

在上面的背景与需求下，对于采用两种数据库系统集成问题是我们完成该项目的主要障碍。为了完成该项目的开发，我们对原有的系统和新开发系统的做了一些技术和方法的调整。这些方面主要有如下方面：

一、重新规划数据层次。通过我们对用户需求的分析，对数据分布的理解，觉得了在集成异构数据库层的时候，我们采用尽量保护数据的原则。我们按照之前的数据分层方法，在 ORACLE 的数据仓库中，重新增加了两个层次的数据，分别是 DM 层和 REF 层，其中 DM 层的数据是 EDS 层数据的较大的粒度汇总过程，主要的来源数据是 DB2 的 EDS 层。通过调度业务逻辑程序，实现如下的数据汇总过程，STAGE 层记录数据库清单的原始数据，没有经过任何的转换和变化，而 ODS 层主要是针对时间和地市维度进行的第一级汇总的数据，该数据基本上是以天为单位的历史数据。EDS 层是对其他维护进行统一化后的汇总数据，该数据主要是对 ODS 数据进行转换和汇总的过程结果。DM 层则是在 EDS 层的数据上，通过企业需要的业务逻辑，如每月统计数据，平均值，指标考核值，预测值等业务逻辑，把 EDS 层的数据汇总到 DM 层的数据中，并且通过特定的数据库权限和视图的方法，把各个地市需要查询的数据汇总到该层的数据表中。通过这样的数据划分后，我们就等到的很好的数据分层结构，为下面进行异构数据库业务逻辑集成奠定了重要的基础。

二、ETL 调度程序的异构数据库处理。在规划了数据分层后，我们遇到的另外一个难题是 ETL 调度程序如何在这两个数据库中调度处理的过程，按照我们原先的设想是通过 Shell 程序来调度两个数据库中不同的存储过程，完成汇总数据的过程。但是这样的方法显然会出现异构数据库中很多的调度问题，而且会出现在调度程序中出现 Shell 再调用 Shell 等问题，使得业务逻辑混乱而无法以后的维护结果。为了缓解这样的问题，我们大胆的采用了 Perl 技术作为业务逻辑层的脚本处理平台。通过该平台，我们只要处理不同数据库的 SQL 就可以了，对于连接数据库以及调度的脚本的开发，调试，测试等提供了重要的保障，另外 Perl 程序另外一个特点就是可以面向对象，这样只要我们开发一些公共的模块，通过对象的形式，就很容易把业务逻辑的 SQL 嵌入到 Perl 程序中，从而解决异构数据库集成的问题。

三、异构 SQL 的函数处理。在搞好可以统一调度的问题后，数据仓库基本上能够通过调度程序跑出对应的数据了，但是我们开发人员在开发的过程中非常的头疼，因为他们比较熟悉的 DB2 函数，在 ORACLE 中要寻找其他的替换方法，并且某些业务逻辑还要书写两套的 SQL 语句来处理业务逻辑。这样的妨碍了我们快速开发的目的。为了解决该问题，我们设计了一套通过的 Perl 函数库，该函数库式以 SQL-92 标注为基础，通过 DB2 中我们经常使用的函数集合的汇总过程，我们通过参数的输入设计方法，把按照我们标准编写的 SQL 语句转换成数据库中真正执行的 SQL 语句，通过正则表达式以及 SQL 字符串特定分解的过程，转换成特定数据库的 SQL，并通过 Perl 函数执行该业务逻辑。这样，我们完成了异构数据库集成的中的 SQL 异构的处理。

通过以上的三个种处理过程，我们最终解决了数据集市项目中的异构数据库问题，并且在 2009 年 5 月份完成了该项目的验收，把真正的数据通过经营分析系统的 ETL 调度程序下发到各个地市中，获得了用户的一致好评。同时由于该系统集成的时候充分考虑了开发人员的使用 SQL 习惯，所以开发的效率比较高，比使用存储过程的开发过程减少了三分之一的开发时间。但是该系统还是存在很多的问题，首先是异构 SQL 的函数还不是很多，并且对于一些特殊的函数，由于需要考虑到两个数据库转换等问题，曾经出现果效率的瓶颈问题，后来经过多方的考虑，把某些功能定义为只能有一个数据库使用的方法来解决。另外，在 Perl 执行的业务逻辑中，我们还是比较陌生，虽然开发处理了很多的通用模块，但是没有定义比较好的接口，如果以后需要改进的时候，可能会出现大面积的修改工作量。我们建议在 Perl 开发平台上，应该多做一些规划，如果能够把 Perl 模块做成数据库连接构件，SQL 执行构件，函数转换构件等等构件化的形成后，并且定义良好的接口形式，这样会使数据集市项目在集成异构数据库上更加的好，同时可以集中解决一些性能的问题，以优化系统的运行效率。

论基于 UML 的需求分析

摘要

2008 年 3 月 1 日至 12 月 20 日，我参加了“数据安全访问平台”项目的开发，担任系统分析员的工作。该项目是某行业用户“数据中心二期”建设的主要内容，目标是：建立数据统一访问接口及其使用标准，规范、约束和审计数据应用访问数据库的行为，对数据应用提供强制审计的技术手段。由于该系统是所有应用的基础平台，对系统的可靠性与性能有较高要求，同时由于没有成熟的现有系统作为参照，该项目存在较高的风险。本文结合作者实践，讨论了在项目中基于 UML 的需求分析。我们使用用例图描述用户与系统的交互；使用类图描述系统的核心概念；使用部署图描述系统的网络部署；使用活动图描述系统的应用流程。由于采用了 UML 中的多种技术，使得我们能从多个方面完整的把握需求，有效的保证到了需求工作的质量。最后，分析了需求工作中存在的问题和改进的方法。

正文

一、项目概述

2008 年 3 月 1 日至 2008 年 12 月 20 日，我参加了“数据安全访问平台”项目的开发，担任系统分析员的工作。“数据安全访问平台”是某行业用户“数据中心二期”建设的主要内容。在二期建设中已建成数据的统一存储和统一分发框架。但主要存在以下问题：无法获得应用用户对数据库的操作日志；开发人员对数据库的使用不规范，查询的结果集过大，导致数据库的性能大幅下降；应用直接使用数据库的登录数据库，存在着一定的安全隐患。“数据安全访问平台”的目标是：建立数据统一访问接口及其使用标准，规范、约束和审计数据应用访问数据库的行为，对数据应用提供强制审计的技术手段。

该项目具有较高的业务需求风险和技术风险。由于没有成熟系统做为参照，该项目需求不是很明确，而且系统涉及甲方多个利益相关方，各方对系统的安全和审计功能、运行维护、可靠性、性能和易用性有者不同的观点，某些观点之间还存在冲突。同时系统作为“数据中心”的基础设施之一，所有的应用系统都要通过本系统完成数据库访问。系统的可靠性和性能直接影响到应用系统的正常运行。整个系统分为 6 个子系统，包括 JDBC 驱动封装子系统、ADO.Net 驱动封装子系统、WebService 接口子系统、管理配置网站、存储子系统（SQL Server 2005 数据库，存储配置信息）和监控子系统（数据库网络协议分析与连接控制）。

二、UML 在需求工作中的应用

项目组采用一个精简 RUP 开发模型指导项目的整个开发流程。在初始阶段主要采用用户访谈、用户调查和联合讨论会捕获用户需求，进行初步分析，完成《需求规格说明书》初稿，通过用户评审；在细化阶段，对需求进行了细化，并在采用原型法验证用户需求，完成《需求规则说明书》更新稿，通过用户评审，作为项目验收的依据。

项目开发中，我们采用了统一建模语言（UML），并使用了 Rational Rose 工具。在需求工作中，我们主要使用了 UML 中的用例图、类图、活动图和部署图。

1、用例技术的应用

整个需求开发都是围绕着用例技术开展的。首先，我们明确了系统的利益（查书）相关方、确定了系统边界和建设目标，以及主要功能特性。由于系统涉及甲方多个利益相关方，包括：应用科、维护科、网络科、信息中心领导和应用开发方，他们对系统的安全和审计功能、运行维护、可靠性、性能和易用性有者不同的观点，同时本系统与已建成的网管系统和单点登录系统存在着交互关系，这给确定系统的目标和边界带来一定的难度。项目初始阶段开始，我们先通过用户访谈、用户调查了解了各个用户对系统的观点；随后，我们召开了联合讨论会，会上我们描述了各个用户的观点，以及之间可能的冲突，供各方进行充分的讨论，会议最后，信息中心的领导统一了各方的意见，对系统达成一致的目标，并明确了系统主要的功能特性，确定本系统与网关系统和单点登录系统的关系。

其次，我们建立用例模型，并细化关键用例。明确了系统的目标和主要功能特性后，我们采用用例模型对需求进行分析。整个系统包括六个用例包：访问接口包用例包、审计信息管理用例包、用户认证信息管理用例包、数据库连接资源管理用例包、访问控制信息管理用例包、数据库连接监控管理用例包和审计日志管理用例包，共包含 55 个用例。对大部分用例我们只描述了一个基本正确流程，只对 5 个关键高风险用例进行了细化描述，包括：前置条件、后置条件、基本流、扩展流和相关约束。这项工作完成后，我们编写了《需求规格说明书》初稿，提交用户进行了审核。经过修改后，通过用户评审，作为初始阶段的主要成果。

最后，细化用例模型。在细化阶段，我们细化描述了所有的用例，完成《需求规则说明书》更新稿，通过用户评审后，作为项目验收的依据。

2、类图的应用

用例技术描述了系统需求的动态结构，但对于需求特性和用例中出现的概念，并没有统一的分析。我们使用类图描述系统的核心概念。本项目中涉及的核心概念主要包括：逻辑连接、用户、应用、受限角色、认证 SQL 语句、参数取值范围的约束和查询结果集的限制。在分析核心概念时，我们主要关注：概念之间的关联关系，是否具有相同的生命周期，和具体的对应关系（一对多、多对多和多对一）。

3、部署图与活动图的应用

由于整个系统包含多个子系统，各个子系统部署在不同的节点，需要考虑用户的网络结构是否能支持。在需求阶段，我们也描述了整个系统的部署。其中监控子系统比较特殊，由于其是通过分析 Oracle 数据库的网络数据包进行工作的，因此监控子系统必须接入核心交换机的镜像端口。我们将部署图与网络科进行了确认。

我们使用活动图描述系统的应用场景。由于本系统对应用系统的开发增加了一层管理，因此应用系统的开发方与信息中心存在一个交互流程：应用开发方首先使用本系统的开发版进行本地开发，并填写基本配置数据；然后，在用户的生产环境中部署应用，并提交基本配置数据；最后，信息中心的管理员对配置数据进行修改后，应用在生产环境中才能运行。我们使用流程图描述了整个过程。用通道表示应用开发方和信息中心，并描述了各个通道内的流程以及通道之间的交互。

三、总结

由于没有成熟系统做为参照，该项目具有较高的需求风险。由于在整个开发过程中使用了 UML 的用例图、类图、部署图和活动图，这使得我们能从多个方面完整的把握需求，有效的保证到了需求工作的质量。

本项目的需求工作中，我们也遇到的一些问题，主要是维护 Word 文档与模型的一致性。我们使用 RationalRose 建模，使用 Word 写文档，通过截图的方式将模型插入文档，因此需要手工维护两者的一致性。这种方式较繁琐，容易出错。今后的工作中，我们准备使用 Rational 公司的 Soda 工具，自动将 Rose 中的模型插入到 Word 文档中。

论软件的性能优化设计

摘要:

本文结合我 2008 年在***人民银行实施的 E 户通电子转账系统的经历,就软件的性能优化设计进行了详细讨论。在系统的设计中针对实际应用环境主要从以下几方面对系统性能进行了优化设计:(1)选择当前成熟的三层 c/s 体系结构进行开发,以减轻数据库服务器的负担;(2)优化数据存取策略以降低系统运行对网络带宽的要求;(3)对客户端应用进行优化,如减少数据请求量、相关的处理利用多线程进行以提高系统的响应速度。通过以上优化设计,系统满足了企业百万级以上主题数据库处理要求,系统开发获得了成功。最后对系统中存在的不足进行了简要的总结,如未考虑查询的需求等等。

正文:

近年来,各银行的信息化发展非常迅速,直接带动了银行中间业务的发展。银行随着多年的建设,已经在中间业务上建立了多个业务系统:如行内的通存通兑系统、定期借记系统、定期贷记系统等。由于这些系统都是由各银行的内部需求为主导建设的,只能支持单个银行系统内部的服务,难以实现银行间的服务。如以银行代收水电费为例,如工行能代收水费,农行能代收电费,老百姓如果需要同时交纳水电费,就必须同时在这两家银行开户,十分麻烦。与此同时,银行因开展业务的需要,不得不与不同的收款单位联网,如水厂,电厂等。数量一多,银行也很麻烦,安全控制越来越复杂。老百姓和银行都迫切要求进一步提升服务质量,实现银行中间业务联网,合理规划和利用银行的总体资源。为此,***人民银行领导经过研究决定于 2005 年利用自身技术力量开发设计 E 户通电子转账系统,作为银行间的电子转账业务支撑平台,实行 7X24 小时连续运行,满足银行间的跨行转账业务、定期借、贷记业务,并支持各商业银行在此基础上建设新的中间业务。我作为工程实施组组长,组建了 12 人的开发队伍,全程参与了项目建设。项目从 05 年 7 月启动,06 年 4 月结束,历时 9 个月。

E 户通系统涉及到与相关银行、相关企业的联网,支持批量和两种业务处理模式,支持定期借记(如定期代收水电费)、普通借记(如支票)、定期贷记(如代发工资)、普通贷记(如银行间转账)、银行柜面现金缴费(如老百姓缴纳路费等)、客户签约管理、网上业务受理等。除了满足以上的业务功能需求以外,还要应对如下性能方面的需求:系统的大部分业务工作都由分散全市的银行网点和联网企业受理,相应要求系统适合于分布式的应用环境并提供较好的性能,要能支持 500 个并发实时处理;必须在较小的网络带宽占用下完成日常业务处理,在使用 64KB 的 DDN 与我系统连接,要求实时业务处理(128 字节/笔)要求在 20 秒以内完成;对于批量业务数据采取包处理制,最大 800 笔一包(约 150KB),要求对包中每笔数据进行 MAC 验证正确后转发至相关银行,每包处理时间不能超出 45 秒;要求查询时,最迟必须在 15 秒内返回结果;资金清算,报表统计与生成工作等,不能影响其它业务的正常进行。

根据银行的业务和管理特点及上述提出的对软件性能方面的要求。在对软件的性能优化设计中我和系统分析人员进行了系统分析:因为该项目的硬件投资较为宽裕,可以购买性能较好的服务器。因此,系统分析主要集中在数据库的设计和编码问题。对于 500 个并发问题,考虑到目前主要商业银行已经实现数据集中,各银行网点的交易数据是通过其管辖行发起,交易量会很多,但是银行总量不会很多,因此交易模式拟采用长连接模式,一方面是保证交易效率,一方面是保证交易可靠性,简化设计。由于是异构环境开发,最好采用成熟的应用服务器;对于实时交易时间问题,数据包的大小不是问题,关键是合理安排交易各区段的时间划分问题,考虑到受理方的处理能力,应至少保证它有 10 秒的处理时间;对于批量系统的时间要求,主要在于两个方面:一是 MAC 验证的时间;二是对包内数据的区分和路由转发。其中后者是重点,需要在网络实环境下进行验证。对于查询性能要求,考虑到主要是对交易历史的查询,应适当分离当前库和历史库。

鉴于以上的分析,我们在系统开发中做了以下的设计和安排:

首先我们选择了目前在数据库开发方面非常成熟的 BEA 公司的 WEBLOGIC 应用服务器模式,数据库服务器采用 SYBASE EAServer。这样,系统具有良好的兼容性与可扩展性的同时,也为系统的性能改善提供了可能。

其次,在系统中我们根据业务处理分散在异地的特点,采用了三层 c/s 体系结构,通过三层 c/s 体系结构的设计可以将数据库的访问及业务处理逻辑移到应用服务器进行。由于系统应用特点,我们在应用服务器的设计中主要采取了长连接池和负载均衡的技术来提高系统性能。对于客户的新数据库连接我们首先去查找连接池中是否有可用的连接,如果没有才则建立之,否则直接利用连接池中的连接来处理客户端的请求,这样做可以有效降低系统因建立与数据库的连接而带来的性能影响,同时也可以较好的节约系统资源。我们针对系统主题数据库容量上百万及日常业务处理繁重的情况,充分利用 BEA WEBLOGIC 应用服务器系统提供负载均衡的功能,如在业务处理高峰期系统将会自动通过调配服务器组的利用率,提高了应用服务器的响应能力。同时,在系统开发中我们采用了大量的存储过程对业务进行处理,既简化了业务逻辑变化带来的维护工作量,在提升数据库服务器的处理能力也提高了系统的性能。

第三,合理确定实时交易的时间区段。对于此类交易,我方只是转发,处理的重点在于受理方。受理方可能有各种原因会导致时延。因此我们设定受理方有 10 秒的受理时间,若超过此时间,我方将会自动提示发起方业务超时。为防止业务多次超时引起系统等待造成效率下降,我们在程序中设置了阈值,对达到阈值的网点实行交易限制。

第四,对于批量交易处理问题,在***研究所的大力支持下,我们找到了合适的网络 MAC 加密机,能够满足系统对 MAC 运算的时间要求。我们要重点解决批量业务交易包问题。为此,我们需要第一手数据。我们模拟了一批数据,进行网络实环境的测试,我们发现,在数据包满载且包内数据目的地为不同银行的时候,在交易高峰期,很容易超时。一方面是受理银行可能超时,一方面是中心端可能超时。因此,经过仔细分析,我们确定以下组包原则:按照业务种类和单一受理行组包;业务发起方只需要关注业务包是否发送到中心,而不必关心是否到受理行。这样,一方面可以简化中心端的操作,另一方面是发起方等待时间短。

第五,合理规划设置查询服务,专门设置查询服务器,提供对当前日以前各交易的查询活动。这样,既可以满足客户的需求,又可以有效防止对交易的影响,客观上提高了数据库的处理能力。

第六,我们在客户端主要采取 GUI 方式与用户进行交互,使用多线程技术对用户输入进行处理,如用户输入完相关资料在提交时,系统利用线程来进行处理,这样系统主线程就不会因等待处理结果而停顿。另外,系统在查询中也大量使用了线程技术,系统首先利用线程对用户要求的资料进行查询,待结果出来后再交给主线程将结果显示出来。在显示浏览表格或用户查询结果时并不是一下将所有的结果都从服务器提取过来,而是每次传输 60 条记录,屏幕上显示 20 条,还有 40 条暂存在本地,如果用户继续查看则进行“翻页”操作以提高系统的响应速度。

另外,我们将批量业务处理、报表处理等大作业量尽量在系统闲时进行处理,一方面可以缓解业务高峰期的性能压力,另外还可以实现系统性能。

通过综合利用以上技术,系统于 2009 年 4 月投入运行时就取得了较好的效果,实时交易时间一般不会超过 3 秒,最长 15 秒;系统峰值处理能力远超过设计要求,经受住了 5 倍数据量的考验;批量交易数据处理一般在 15 秒内就得到了应答。客户查询交易一般在 6 秒左右得到应答。系统主要性能到了设计要求,得到了银行和企业的好评。

尽管系统建设取得了成功,但是我也看到了不足之处。设计中只考虑了满足联机交易均衡负载和提高性能,但是在实际应用中,发现银行客户对交易的查询是大量的,设计的前瞻性不足,余地不大,容易引发服务器堵塞,需要进一步考虑查询优化。

此次系统的顺利实施为我在中、大型软件性能设计方面积累了较多的经验,为我以后的工作提供了很好的帮助。同时,软件技术的日新月异也促使我要不断更新自己的知识结构,为应对不同体系结构的软件分析与设计做好准备。

论软件架构的选择与应用

摘要:

2006 年 5 月,我所在的公司承担了某省社会保险管理信息系统的开发工作,我在该项目中担任系统架构设计师职务,主要负责设计应用系统架构和网络安全体系架构。该系统以 IC 卡为信息载体,完成劳动和社会保险的主要业务管理,即“五保合一”管理,包括养老保险、医疗保险、劳动就业和失业保险、工伤保险、女工生育保险。整个业务流程十分复杂,牵涉面相当广泛。

本文以社会保险管理信息系统为例,讨论了软件架构的选择和应用。整个系统采用具有四层的层次式软件架构的设计思想,在业务管理层的设计中,采用了互连系统构成的系统的架构,把整个业务管理系统划分为八个从属系统。各从属系统的架构可以相同,也可以不同。整个系统的开发工作历时 19 个月,目前,该系统已经稳定运行 1 年多的时间。实践证明,这种架构设计有效地降低了维护成本,提高了系统的开放性、可扩充性、可重用性和可移植性。

正文:

2006 年 5 月,我所在的公司承担了某省社会保险管理信息系统(以下简称为“SIMIS 系统”)的开发工作,我参加了该项目前期的一些工作,担任系统架构设计师职务,主要负责设计应用系统架构和网络安全体系架构。

1. 项目概述

SIMIS 服从于国家劳动和社会保障部关于保险管理信息系统的总体规划,系统建设坚持一体化的设计思想,总体目标是建立比较完备、高效、与劳动和社会保障事业发展相适应、与国家经济信息系统相衔接的劳动和社会保险管理信息系统,实现劳动和社会保险管理体系的技术现代化、管理科学化。

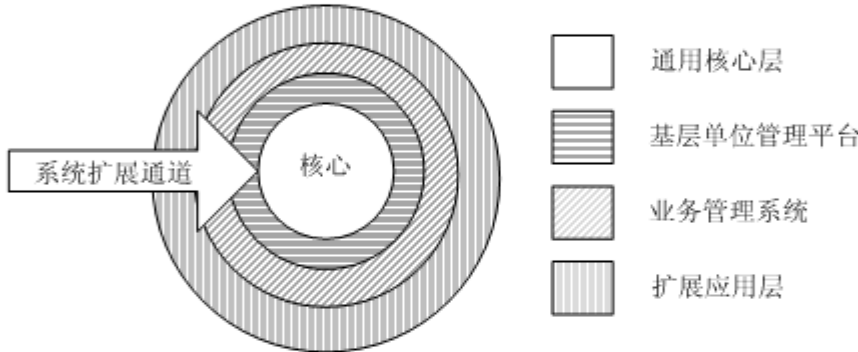
SIMIS 系统以 IC 卡为信息载体,完成劳动和社会保险的主要业务管理,即“五保合一”管理,包括养老保险、医疗保险、劳动就业和失业保险、工伤保险、女工生育保险。整个业务流程十分复杂,牵涉面相当广泛。SIMIS 系统由省、地市、县三级组成,网络纵向覆盖全省各级劳动和社会保障机构,横向与财税、银行、卫生、邮政、企事业单位联网,是一个典型的广域网络系统;系统设计按照社会保险与个人账户相结合的模式,以养老保险为重点,并以此为全省劳动和社会保险管理信息网络主干网络,带动劳动力市场等其他社会保险业务管理信息系统建设。

2. 架构设计

虽然国家劳动和社会保障部对整个业务有一套规定的指导性流程,但是,在调研的过程中,我们发现各市、县或多或少地存在使用“土政策”的情况,正是这种“土政策”致使软件在设计阶段具有很多的不确定性需求。另外,我们还考虑到将来用户需求可能会发生变化,为了尽量降低维护成本,提高可重用性,我们引入了层次式软件架构的设计思想。

SIMIS 系统采用层次式软件架构的基本出发点在于,这种软件结构不但能够满足不同规模的用户(县级、地、市级)的需求,可以方便地在最小的完成基本功能的基本系统和最大的完成所有复杂功能的扩充系统之间进行选择安装,而且通过逐层功能扩展的方法来进行软件实现,有利于程序设计和构件开发。同时,一定级别的抽象层可以作为一种知识积累,对于同类软件的快速开发有着很大的作用。

根据调研的结果,我们把 SIMIS 系统设计成具有通用核心层、基本应用层、业务管理层和扩展应用层四个层次的层次式软件体系结构,如图所示。



通用核心层完成的是软件的一些通用的公共操作,这些操作能够尽量做到不与具体的数据库和表结构相关。基本应用层是 SIMIS 系统数据采集的主要来源;业务管理层是对基本应用层的进一步扩展,主要完成 SIMIS 系统的业务管理,管理内容涉及劳动者个人、企业和其它劳动组织的微观信息,能够实现数据的初步汇总。业务管理层与其内包含的两层一起构成了 SIMIS 的典型应用系统。扩展应用层是在典型应用系统的基础上扩充了一些更为复杂的功能,如对政策决策提供依据和支持,对政策执行状况进行监测、社会保险信息发布及个人账户电话语音查询系统等。

SIMIS 的设计和开发重点放在业务管理层上,在这一层的设计中,我们采用了互连系统构成的系统的架构,把整个业务管理系统划分为失业保险管理、养老保险管理、医疗保险管理、女工生育保险管理、工伤保险管理、工资收入管理、劳动关系管理、职业技能开发管理等八个从属系统,所有的从属系统共用同一个数据库管理系统,每个从属系统作为单独的系统,由不同的开发团队进行独立开发。

从属系统可以自成一个软件系统,脱离上级系统而运行,有其自己的软件生命周期,在生命周期内的所有活动中都可以单独管理,可以使用不同的开发流程来开发各个从属系统。

各从属系统的架构可以相同,也可以不同。例如,在开发“养老保险管理系统”这个从属系统时,选择了正交软件架构。我们将整个系统设计为三级正交结构,第一级划分为八个线索,每个一级线索又可划分为若干个二级线索,每个二级线索又可划分为若干个三级线索。一条完整线索如图所示。

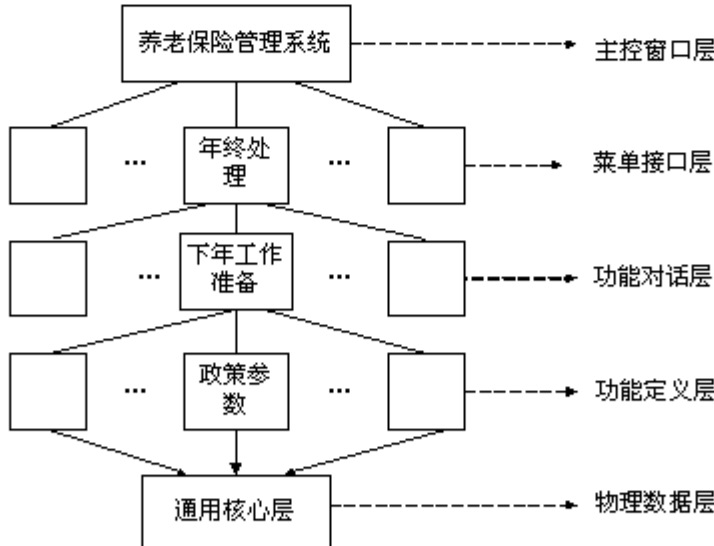


图 完整的一条线索结构

每一条线索完成整个系统中相对独立的一部分功能,所有线索是相互独立的,即不同线索中的构件之间没有相互调用。由于采用了正交结构的思想,在系统开发时,我们分成若干小组并行开发,视开发难度情况,每个小组负责一条或数条线索,由一个小组来设计通用共享的数据存取构件。由于各条线索之间没有相互调用,所以各小组不会相互牵制,大大提高了编程的效率,缩短了开发周期,降低了工作量。

虽然各个从属系统相对独立,可以进行并行开发,但我们尽量注意了软件重用,以节约开发成本,加快开发进度。例如,“基金收缴”是失业保险、养老保险、医疗保险、女工生育保险和工伤保险五个从属系统中都要进行的操作,且其操作流程大致一样,只是基金收缴的参数有所区别。我们就只安排养老保险从属系统的开发团队开发一个通用构件,提供参数接口供其他从属系统使用。

3. 总结

在 SIMIS 系统的架构设计中,我们引入了层次式软件架构的设计思想。根据调研的结果,把 SIMIS 系统设计成具有通用核心层、基本应用层、业务管理层和扩展应用层四个层次。SIMIS 的设计和开发重点放在业务管理层上,在这一层的设计中,采用了互连系统构成的系统的架构,把整个业务管理系统划分为八个从属系统。每个从属系统作为单独的系统,由不同的开发团队进行独立开发。各从属系统的架构可以相同,也可以不同。

以上设计有效地降低了维护成本,提高了系统的开放性、可扩充性、可重用性和可移植性。但 SIMIS 的设计和开发也存在一些不足。例如,由于采用了互连系统构成的系统的软件架构,使资源管理开销增大,各从属系统的开发进度无法同步等。又如,由于采用了 B/S 和 C/S 结构混合的异构结构,使外部用户修改和维护数据时,速度较慢,较烦琐,数据的动态交互性不强等。