

中文摘要

经过多年的发展，企业资源计划（Enterprise Resource Planning, ERP）已在全世界广为流行，成为企业在激烈竞争中取胜的必要条件。但当今 ERP 的发展仍然存在一些问题，这主要表现在一方面多数的 ERP 软件对企业二次开发提供的支持还不够，另一方面只有少数大公司才负担得起 ERP 软件。

本文所完成的支持 ERP 开发的中间件与组件系统——BeaconERP 系统在一定程度上解决了上述问题。该系统适用于中小型企业在其基础上进行二次开发，获得需要的 ERP 系统。

BeaconERP 系统采用分层体系结构，总体分为中间件层与应用层，应用层处在中间件层之上。为了方便用户开发，系统还提供了一个辅助开发插件。

系统中间件部分又被分为三层。核心组件层提供了最为核心的功能，这些功能在大多数企业应用中都需要，该层的组件包括权限控制，Web 架构，日志系统和 O/R 映射。基本组件层提供了更为高级的功能，但它们仍然与具体应用无关，该层组件包括消息系统，报表系统和搜索系统。扩展组件层提供了与具体应用领域相关的组件，它主要靠用户来扩展。辅助开发插件可以图形界面的方式有效的支持用户对配置文件的操作，包括文件骨架生成，文件管理及辅助编辑。

实际的运行效果及随后的测试结果表明，该系统具有易用性，可扩展性，以及较高的性能。可以为中小企业开发 ERP 软件提供有效的支持。

关键词： ERP 中间件 组件 软件架构

ABSTRACT

After many years of development, Enterprise Resource Planning (ERP) has been so popular that it is indispensable for a company to survive in the intense competition. However, there are some problems in ERP systems nowadays, some of them do not provide enough support for secondary development, and some are so expensive that only a few big companies can afford one.

BeaconERP, which is implemented in this paper as a middleware system for development of ERP applications, has solved these problems to some extent. It is suitable for small and middle scaled companies to develop their own ERP applications on top of it.

BeaconERP adopted layered architecture. There are generally two layers, the middleware layer and the application layer. The application layer is built on top of the middleware layer. Besides that, BeaconERP includes a Plug-in to facilitate development.

The middleware layer is subdivided into 3 layers. The Core Components Layer realizes the most significant functions which are necessary in almost all enterprise applications. Components in this layer are: Access Control, Web Framework, Log System and O/R Mapping. The Basic Components Layer provides more advanced functions, but they are also independent of specific applications. Components in this layer are Message System, Report System and Search System. Components in the Extended Components Layer are dependent on specific applications, and they are intended to be extended by clients. The Plug-in can effectively assist users in generating, editing and managing contents of files with a graphic user interface.

Actual implementation and subsequent testing shows that BeaconERP has usability, scalability and high performance. It is capable of giving small and middle scaled companies enough support in developing their own ERP applications.

KEY WORDS: ERP, Middleware, Components, Software Architecture

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得天津大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：樊莉亚

签字日期：2007 年 2 月 5 日

学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名：樊莉亚

导师签名：孙洪英

签字日期：2007 年 2 月 5 日

签字日期：2007 年 2 月 5 日

第一章 研究背景

1.1 ERP 的发展

管理科学思想的不断发展,促使了企业资源计划(ERP)的产生[1],1990年Gartner group正式提出了ERP的概念。此后ERP便展现出飞速增长的势头,据美国权威市场机构AMR Research的研究,1998年全球ERP市场总收入达148亿美元,随着当今企业信息化水平的不断提高,这一惊人的数字必然要继续增长下去。

如今ERP已经涉及到了企业战略决策、企业文化、生产经营,供应链,客户关系等各个方面。ERP的基本思想是不断优化供应链从而不断提高客户满意度[2]。作为集成化的企业管理信息系统,近年来,越来越多的企业采用ERP和其他一些管理思想先进技术[3],提高了企业的整体效率和市场竞争力。ERP发展到今天主要经历了以下几个阶段:

1. MRP(物料需求计划):MRP是在早期订货点法的基础上发展起来的。为了降低库存成本,需要选择合适的的时间进货。太早进货,会加大库存成本,太晚又无法供应生产的需要。供货点法即是一种确定合适的进货时间的方法。但是供货点法只适用于稳定消耗的物料,同时容易造成死库存。为了对其进行改进,提出了MRP的思想,它将产品按结构分层,并将物流需求分为独立需求与相关需求,与订货点法相比,MRP可以更好的确定物料的需求,降低企业成本。

2. 闭环MRP:在上个世纪70年代,MRP发展形成了闭环的MRP生产与控制系统。MRP虽然可以很好的预测物料需求,但是它并没有考虑到企业实际的生产能力,造成了一些计划并不能实际完成。闭环MRP对此进行了改进,引入了企业能力管理的概念,做到了物料计划与生产能力的统一,一定程度上实现了按需定产。同时在这一阶段计算机技术也有了进一步的发展,许多企业开始借助计算机对生产实施闭环控制。

3. MRP II:该阶段开始于上世纪80年代。在闭环MRP的基础上,将资金流引入到MRP中,形成了MRP II。在MRP II中,企业物流与资金流保持高度的一致,可以更全面地反映企业的经营状况。另外MRP II能为企业生产经营提供更为完整而详尽的计划,更能考虑到企业整体的情况,从而可以更好的协调企业内各部门的活动,因此它极大的提高了企业的整体效率。

4. ERP: 开始于上世纪 90 年代, 在 MPR II 的基础上出现了 ERP。它的功能更强大, 一方面是由于新技术如互联网技术, 数据库技术, 图形用户界面的出现; 另一方面是由于管理科学的发展。ERP 在 MRP II 的基础上又加入了知识管理, 人力资源管理, 品质管理, 电子商务, 售后服务, 在线分析等新的功能[1]。ERP 不仅局限于企业内部, 它已延伸到企业上下游的供应商与客户中。ERP 真正实现了企业物流, 资金流与信息流的高度统一, 已成为了集成化的企业管理软件系统。

如今 ERP 已经成为企业在国际化的竞争环境中取胜的必要条件。它可以为决策者提供决策数据, 分析区域的销售情况, 制定销售方案或调整产品结构, 还可以准确的掌握产品生产过程, 找出瓶颈的所在。具体来讲, ERP 的作用包括以下几方面:

- 提供系统信息集成, 从而可以方便的实现数据共享, 使得企业内任何一点都可以方便的获得业务数据和资料。
- ERP 可以更快的响应市场, 为企业赢得宝贵的时间, 从而在激烈的市场竞争区取得先机。
- ERP 将许多管理方面的常规工作自动完成, 将企业管理人员从繁琐而枯燥的常规事务处理中解放出来, 从而可以更专心研究企业生产经营过程中出现的问题。
- 由于 ERP 的精确性, 可以在工作控制方面能够做到分工明确, 适时控制, 对每一环节所存在的问题都可以随时反映出来, 从而为分析企业运行情况提供重要数据。
- ERP 可以从企业整体角度出发, 协调各部门的业务, 使企业的资源得到统一规划和利用, 将各部门联成一个整体, 协调运作。从而使企业整体利益最大化。

综上所述, ERP 的发展经历了由最初的 MRP, 闭环 MRP 到 MRP II 再到 ERP 的阶段, 这个过程中业务流也由最初单纯的企业物流发展到企业物流与资金流相结合, 再发展到目前的企业物流, 资金流, 信息流相结合的形式。ERP 已经成为企业取得成功必不可少的条件。

1.2 问题的提出

近些年来, ERP 系统有了飞速的发展, 提高了企业的信息化水平, 大大降低了企业的管理成本。在国外, ERP 的应用已经带来了巨大的经济效益和社会效益; 在国内, ERP 的实施也进入了全面阶段, 但是效果却不是十分明显。形成这种状

况的原因主要有以下几点:

1. 国外 ERP 企业不了解国内企业的现状,因而产品并不适合应用于国内企业,导致了一些著名 ERP 软件公司的产品在国内会出现水土不服的现象。

2. 国内 ERP 产品大部分都局限于某一领域,很多公司对 ERP 功能的认识还都集中在以财务为中心的企业资源调配上,产品的功能的覆盖面还有问题。

3. 由于不同的企业都有自己的特点,需根据各自企业具体应用要求对已有 ERP 软件进行二次开发。但是就目前的 ERP 软件来看,用户二次开发需要了解软件的内核以及结构,用户获得的开发工具也比较匮乏。

4. 市场上,不论是基于 C/S 架构或 B/S 架构的产品提供给用户的都是比较简单的界面。这不符合以人为本的设计原则,并不能有效地为用户提供他们需要的功能。

基于以上观察,本文设计并实现了支持中小型企业开发个性化 ERP 系统的中间件与组件系统——BeaconERP。本系统正是立足于解决上述问题,该开发平台独立于具体的技术平台,例如操作系统、数据库等,保证了 ERP 应用的开发、部署、维护和升级等方面不受操作系统以及底层技术的限制;该平台提供统一的数据接口和应用模型,易于管理和集成所有的应用,帮助企业跨越技术和组织实现业务流程、数据以及界面的集成;该平台将对新一代的协同工作模式提供支持,在部门与企业内部,以及各种跨地区、跨时区、跨系统、跨网络的团队内部,实现安全、高效、方便的信息交换,从而全面提升团队工作效率,降低协作成本。具体来讲,该系统具有以下特点:

1. 功能完备性:为完成对企业资源进行有效管理这一目标,ERP 系统应当提供各种常见的功能,如报表,日志,权限控制等。BeaconERP 对于 ERP 系统内各种常用的功能提供了相应的组件,从而为企业用户提供了方便。

2. 可扩展性:不同企业有其自身的实际情况,因此对 ERP 也有不同的需求。BeaconERP 提供了丰富的接口,用户可通过配置选项设置不同的属性,或在该基础上对系统进行二次开发,满足企业个性化的需求。

3. 优良架构:为了使用户开发出的系统具有较好的可维护性,可扩展性与可用性,在 BeaconERP 开发过程中使用了多种已经在实践中验证过的设计模式,保证了 BeaconERP 具有良好的软件体系结构。

本系统在分析中小企业对 ERP 系统的市场需求的基础上,将 ERP 技术, CSCW 技术, 中间件技术等结合起来,为 ERP 应用提供一个具有可用性、可扩展性、高性能的开发平台。为 ERP 开发人员提供了丰富的接口和服务,可降低 ERP 应用系统的开发周期和费用。相信将对 ERP 技术在企业中的普及起到积极作用。

全文组织结构如下：第二章介绍系统总体架构以及所使用的设计方案；从第三章到第五章详细介绍了系统各层各个模块的功能及设计方案；第六章给出系统实际的运行效果；最后在第七章给出了全文的总结和对未来工作的展望。

第二章 BeaconERP 系统框架分析与设计

2.1 系统定位

为了解决当前 ERP 系统中出现的问题,需要的并不是一个成品软件,而是一系列的中间件与组件,目的是为了辅助用户在其基础上进行二次开发。它应当提供一些 ERP 应用中的常用功能,用户也可在系统提供的基础上进一步开发自己需要的其它组件。它应当提供一个优良的架构,使得在该框架上开发出的系统也有良好的架构。具体来讲,系统应当满足性能与功能两方面的要求。

性能方面是指系统运行与使用过程中的一些参数,它包括:

模块化:系统各个组件都是独立的模块,在设计过程中应当尽量减少各个模块之间的依赖,这样可得到低耦合,高内聚的软件,同时也利于系统的维护[4]。

可扩展性:系统不仅提供了一组常用功能,同时用户也可以很容易的在此基础上扩展自己的功能,满足自身特定的需求。这就要求系统提供统一的接口,只要满足这样的接口,就可以无缝的加入新的模块。

易用性:用户不需要阅读很多的文档或进行很多的培训就可熟悉并熟练的使用系统。这要求各个模块提供简单明确的接口,同时也要求程序有较高的可读性。

高效率:系统运行时间短,需要的内存少。这与系统能够实现时使用的数据结构与算法有密切的关系。因此选择的算法与数据结构应当有较低的时间复杂度和空间复杂度[5]。

功能方面的要求是指系统提供了 ERP 软件中常用的功能,这些功能应当具有一定的典型性,同时也能很好的反应近几年的技术发展趋势。例如,Web 应用已成为当今企业应用的主流[6],那么在该系统中也应当提供相应的组件,对 Web 应用的开发提供支持;再比如,当今企业中的数据量呈爆炸式的增长,为了能够从这些文档中迅速的检索到需要的信息,应当提供一个搜索引擎,因此也应当有相应的模块支持该功能,需要注意的是,模块需要满足一定的扩展性方面的要求,以便企业定制自己的功能。这样的需求在以前的 ERP 中是没有或者不受重视的,但随着技术与企业信息的发展,这种需求也是应当考虑的。

由以上分析,系统中涉及的功能较多,范围较广,必须有一种好的方式组织这些组件,否则会引起系统的混乱。因此应当采用分层体系结构,将这些组件按照功能的高低处在不同能够的层上,这样不仅可以增加层的重用性,减小模块之

间的依赖性，增加局部依赖性，还可以提高各模块的可替换性[8]。

2.2 系统总体框架

如上所述，BeaconERP 采用分层体系结构模式，图 2-1 是 BeaconERP 系统的总体架构图。由图可见，系统由四层组成，其中中间件分为 3 层，处在系统靠下的部分，它们自下而上分别是核心组件层，基本组件层和扩展组件层。按照自下而上的顺序，各层提供更高级的服务。每一层组件需要调用下一层提供的接口或服务，并被上一层的接口调用，为上一层的组件提供服务。用户开发的应用处在应用层，基于这三层之上，利用它们提供的服务来满足自身特定的需求。为了方便辅助文件与配置文件的编写，系统还提供了一个辅助开发插件。

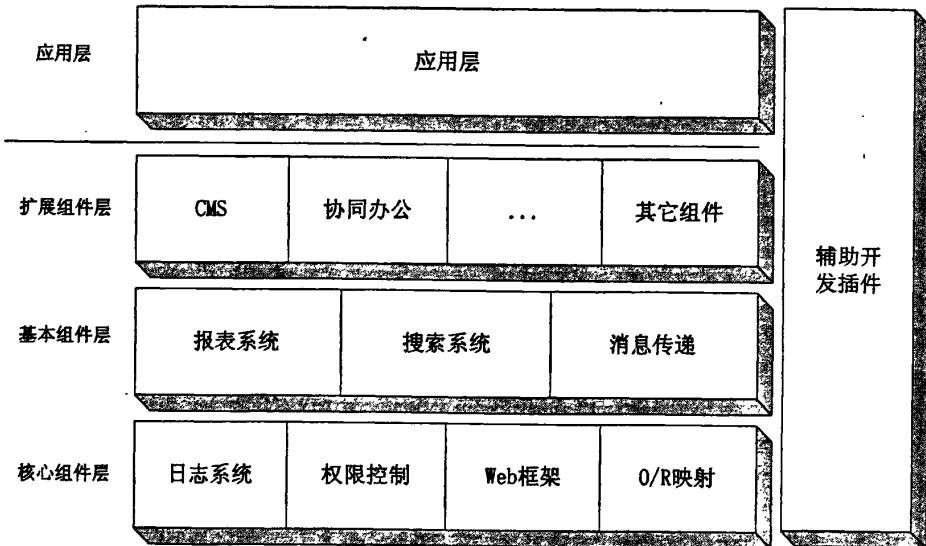


图 2-1 BeaconERP 系统体系结构

BeaconERP 系统的部署结构如图 2-2 所示。系统以 B/S 架构为主，这种瘦客户端的模式具有跨平台，图形界面与客户端无关，无代码泄露，支持多种语言的优点，同时客户端无需安装，无需部署，无需升级。但是对某些特殊的应用，客户端仅有浏览器还不够，对这类应用，BeaconERP 要求在客户端安装一个小的客户端软件，这类应用采用 C/S 模式。总之，系统采用 B/S 为主，C/S 作为补充的运行模式，大部分核心组件部署在服务器上，客户端只有浏览器和一些小的客户端软件。

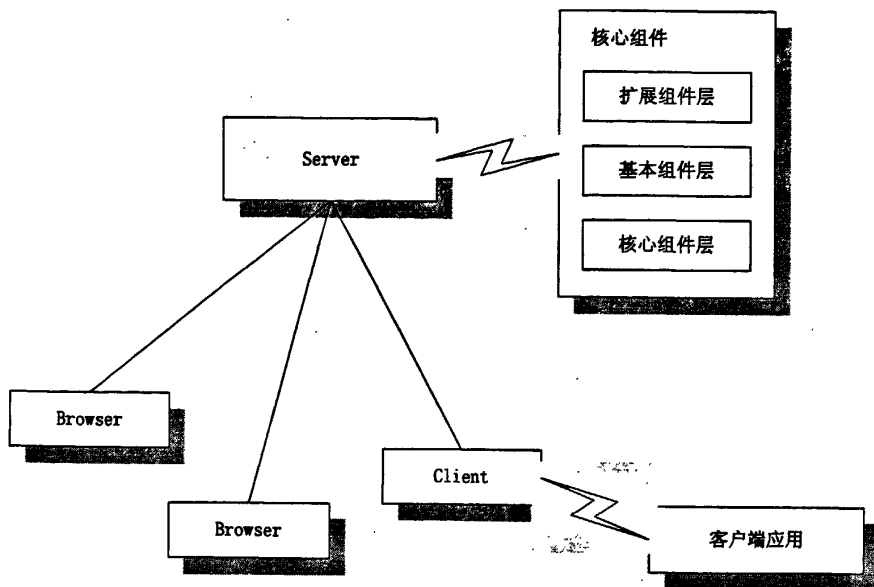


图 2-2 系统部署图

2.3 各层概要介绍

2.3.1 核心组件层

核心组件层提供了整个系统最核心的服务和功能，这些服务和功能不仅在 ERP 系统中普遍应用，而且在几乎任何企业应用中都是必不可少的。BeaconERP 系统核心组件层提供了日志系统，权限控制，Web 框架，O/R 映射这些模块。它们的功能分别是：

日志系统：为系统提供日志信息。该模块可针对不同情况提供不同级别的日志，目前支持的日志级别有信息，警告，错误，致命等 4 个级别。根据日志作用的不同，该模块将日志分为两种类型：数据库日志将日志信息保存在数据库中，使日志信息持久保存，这类日志主要是为了动态监控系统运行状况；文本日志将以文本的方式保存或显示日志，它又可分为控制台日志，HTML 日志，文本文件日志等，这类日志的主要作用是为用户提供调试信息。

Web 框架：近些年，商业应用程序已经由传统的桌面客户端式应用发展到 Web 应用占主流的时代，ERP 应用也逐渐表现出与 Web 与网络集成的趋势[6]。因此 BeaconERP 为了支持 Web 应用提供了 Web 框架模块，该模块的功能有二：一是提供了一个 Web 框架，用户可在此框架的基础上开发出架构良好的 Web 应用，具体来讲，开发出的应用符合 MVC 模式[8]，将业务逻辑从视图中分离出来，

从而使得软件具有更好的扩展性；二是提供了一组实用工具，为开发提供帮助，例如 BeaconERP 提供了一组 JSP 标记库，提供了一些常用的功能，从而简化了显示页面的开发，提高了开发效率。

权限控制：任何企业都必须保证其内部信息被安全的访问，既可以被有权限的用户访问，又可以限制未授权用户对未经授权的资源的访问，这就是对权限控制模块的要求。由于系统中用户数量较大，为每一个用户分配不同的权限将对系统带来巨大的工作负载，因此常用的方法是将用户分为若干组，每一组用户指定一个不同的角色，权限根据用户所具有的角色被分配给用户。权限控制模块即采用了这种基于角色的权限控制[11]，针对用户不同类型的访问操作提供了不同类型的权限控制策略，以最大限度的提高性能。

2.3.2 基本组件层

该层提供一些功能较为基本的模块，这些模块提供的功能在 ERP 系统中具有一定的代表性。与核心组件层相比，该层的功能更为高级更为复杂，但它们与具体应用并没有直接的关系，仍然属于通用功能的范畴。该层在整个系统中有着承上启下的作用。BeaconERP 系统在基本组件层提供了报表系统，搜索系统，消息系统这些模块。

报表系统：报表已成为企业应用中不可缺少的功能，因此该模块提供了 ERP 系统中与报表相关的常用功能，包括报表的生成与报表的管理。用户可方便的对该模块进行配置。用户只需要修改配置文件，就可以控制报表的输出格式，如控制报表输出的样式，报表内容的格式，或重新选择报表的数据源，如数据库的表等等，还可以选择报表的文件类型。目前模块支持 PDF，Excel 两种文件格式的报表。用户还可在此基础上扩展其它文件格式的报表。

搜索系统：该模块的作用是提供一个搜索引擎，对企业中的海量数据进行快速搜索。该模块提供对文档的内容全文检索的功能。目前该模块支持对常用类型文档，如 Word，Excel，PDF，HTML，XML 等的索引，用户也可以扩展该模块，以提供自定义的索引方式。为了满足用户的需求，该模块提供了两种查询方式，用户可以字符串方式提价简单查询，这种方式适用于用户搜索关键字较少的情况；对于用户搜索关键字较多的情况，可以使用系统提供的高级查询功能，BeaconERP 专门针对高级查询制定了一种查询语言（SEL），用户可以使用该语言定制自己的高级查询。

消息系统：根据不同的分类方法，消息可分为许多不同的类型，它们在实际中有着不同的用途。消息系统针对 ERP 系统对消息传递的不同需求，提供了多种灵活的消息传递方式。例如，按照实时性的不同，分别提供了同步消息，异步

消息；按照传输方式的不同，提供了点对点，组播方式消息等。消息系统块可以支持多种网络环境，用户只需修改配置文件，就可以实现不同网络状态下的消息传递。

2.3.3 扩展组件层

扩展组件层位于三层中间件的最上层，提供最为高级的功能，这些功能与具体应用的联系比较紧密，专业化程度也比较强。用户在扩展组件层上扩展自己的特定应用领域的模块，该层可提供的模块有 CMS, 协同办公等。

CMS: 内容管理模块，提供对企业文档内容管理的功能。它使得企业能够迅速跟进大量信息衍生的脚步，从而不需要花费许多时间、人力和物力来处理信息更新和维护工作，这些工作可由 CMS 模块自动完成。

协同办公: 协同办公是办公自动化的下一个方向[12]。因此该模块的主要目的是为了更方便企业内部同一部门内部以及不同部门之间的人员之间进行交流。主要是在基本组件层消息系统的基础上，提供协同办公的常用功能，如文本通信，视频/语音通话，共享白板等等，此外，用户还可以对协作的内容进行保存载入等等。

2.3.4 应用层

应用层是用户在 BeaconERP 中间件层的基础上开发的 ERP 应用，它可分为两部分：基础协作应用是 BeaconERP 平台本身提供的，它提供了某些通用的应用模块或是一些原型 ERP 系统，用户可在原型上逐步演化出自己的系统；用户应用是用户在 BeaconERP 系统上二次开发所得的满足自身特定需求的 ERP 系统。

BeaconERP 在基础协作应用部分提供了一个 ERP 系统销售管理应用。该模块一方面可以演示 BeaconERP 各模块的使用方法，另一方面用户也可以在此基础上逐步演化得到需要的系统。

2.3.5 辅助开发插件

在 BeaconERP 系统多个模块中提供了可扩展的功能，用户只需要修改配置文件或者加入一些很少的代码就可加入新的功能或修改已有的功能。为了对配置文件和其它辅助文件的编辑提供支持，BeaconERP 还提供了一个辅助开发插件，协助用户以图形界面的方式对配置文件进行编辑创建等操作。

综上，BeaconERP 并不是一个 ERP 应用，而是一系列的中间件与组件系统，

用户可在其上进行二次开发,得到满足特定需求的 ERP 系统。系统应当提供 ERP 开发中常用到的功能与服务,这些功能与服务按照高低的不同,分成了不同的层次,每一层依赖于下一层提供的功能完成本层的功能。

第三章 核心组件层

在系统设计中，核心组件层位于中间件最底层，它提供的功能最为基本，与具体应用无任何关联，这些功能不仅在 ERP 的特定领域中有所涉及，在几乎所有的企业应用中都会涉及到。

纵观近些年来企业应用的发展，可以看到如下趋势：对企业动态运行情况跟踪这一要求已被大多数企业应用所接受，同时这也能对系统灾难恢复，数据挖掘等功能提供支持[5]，因此需要一个日志系统，监控企业应用的运行状态。当今任何一个企业应用都离不开权限控制，以防止企业内部信息被未授权的用户访问，这要求系统提供相应的模块，实施权限控制方案，保证企业信息安全。Web 应用已成为下一代企业应用的发展方向，然而从 Web 应用本身的发展趋势来看[6]，开发出一套具有优良架构的 Web 应用并不容易，因此核心组件层需要对此提供支持，使用户在相应模块的框架下开发出架构优良的 Web 应用程序。当今多数企业应用采用三层体系结构[7]，在此过程中经常涉及到类对象与数据库记录的转化，即 O/R 映射，这一枯燥而又必须的工作，核心组件层应当提供相应的组件，使这一转换的过程自动完成。

针对以上提出的需求，核心组件层分别提供了相应的模块，以下分别介绍。

3.1 日志系统

企业运行过程中的日志为企业提供了非常有用的信息，一方面通过对日志信息的数据挖掘，可以为企业的决策者提供有用的信息；另一方面，在系统运行出现故障时，日志可以有限的帮助企业分析原因并提供灾难恢复等功能。因此该模块的主要目的是加入日志，以便监视整个系统的运行情况。

为了满足不同的需要，应当提供多种类型的日志。BeaconERP 日志系统的日志按类型分为四种[13]：控制台日志；文本文件日志；HTML 日志和数据库日志。该模块基于 Apache 开源项目 Log4J[14]，在此基础上提供更高级的日志功能。以下分别介绍各种类型的日志：

1) 控制台日志，输出的日志显示在控制台上。这些信息不做保存，控制台关闭后将消失。这种日志常用在程序的调试当中。

2) 文本文件日志，将日志信息以文本的形式保存在文件中。这类日志由于

持久保存在硬盘上，必须考虑到文件大小的问题：文件不断插入新的日志信息，文件会随着程序的运行会越来越大，如果不及及时清除将会耗尽硬盘空间。为解决该问题，日志系统提供了两个关键属性：**MaxSize** 用来确定文件最大的大小，当文件大小超过 **MaxSize**，新插入的日志将被保存到新的文本文件中；**MaxNumber** 文件数目的最大值，当文件数目超过该值，将以滚动文件的方式保存日志到相应文件，如 `file.txt.1`, `file.txt.2`, `file.txt.3`, ……，`file.txt.MaxNumber` 写满之后之后保存到 `file.txt1`。

3) HTML 日志，将日志信息保存的到 HTML 文件当中。这类日志的主要目的是为了记录 Web 模块的运行信息。从而可以方便管理员或其它高级用户检查系统运行情况及或其它信息。HTML 日志同样有文件大小的问题，因此同样需要设置与文件大小有关的属性 **MaxSize**，和 **MaxNumber**，其作用及相应的命名规则同文本文件日志。

4) 数据库日志，将日志以数据库记录的形式插入到数据库表当中。表的名称由用户指定。如果用户指定的表不存在，模块将自动在数据库中创建相应的表。插入的内容由用户指定，但以下信息是必须的：日志级别（信息，警告，错误，致命），日志创建时间，生成日志的类的全路径名，以及生成日志语句所在的线程。

日志系统体系结构如图 3-1 所示，图中的箭头表示依赖关系。类 `ConsoleAppender` 负责向控制台输出日志，输出方式以追加的方式插入到原日志之后。`RollFileAppender` 类负责向文件输出日志，输出方式同样是追加式。`DatabaseAppender` 类负责将日志插入数据库当中，使用 JDBC 驱动实现与数据库的相连。

`MyConsole`，`MyFile`，`MyHtml`，`MyDatabase` 是具体实现的类，它们分别对应四个日志输出目的地：控制台，文本文件，HTML 文件和数据库。

日志系统提供了方便易用的接口，系统外部模块使用日志模块生成日志也非常方便，只需要按照如下的步骤：

- 1) 通过 `Logger` 类的 `getLogger` 方法获取 `Logger` 类的实例。
- 2) 调用生成的对象的 `addAppender` 方法，确定相应的日志目的地。
- 3) 调用 `Logger` 类实例的插入日志的相应方法，`Logger` 类对不同级别的日志定义了不同的方法，如信息类日志对应 `info` 方法。

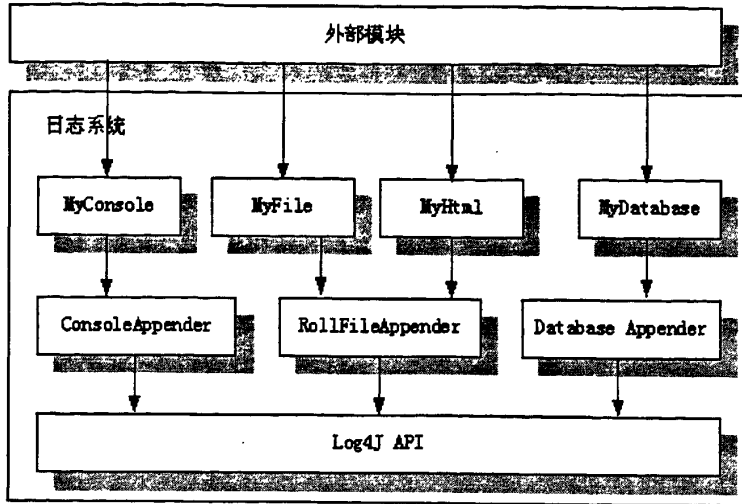


图 3-1 日志模块架构

MyDatabase 实现将日志写入数据库主要是依赖三个成员，分别是：

- 1) DatabaseAppender 类的静态的公共对象 database。
- 2) 在数据库中创建表的 SQL 语句 CreateSQL，它为普通字符串类型。
- 3) 在数据库日志表中插入日志的 SQL 语句 InsertSQL，它的类型为字符串。

所有对数据库的操作是通过 database 成员完成的。外部类使用该成员需要调用 MyDatabase 类的静态方法 getDatabase() 获取该成员。向数据库插入一条日志实际的工作流程如图 3-2 所示。

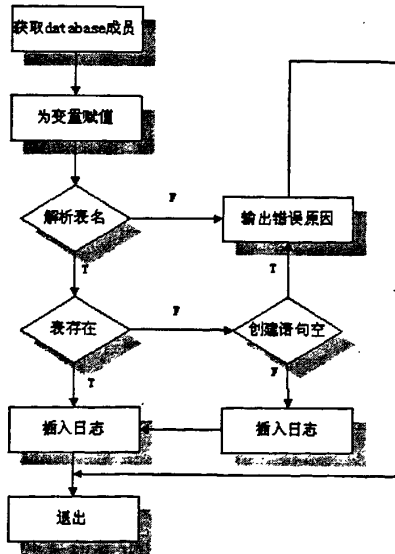


图 3-2 日志写入数据库工作流程图

图中第二步赋值的变量包括：所用的数据库 URL，所用的驱动程序，以及数据库用户名，密码等。插入日志根据 InsertSQL 成员定义的 SQL 语句。在插入之前首先要检查日志对应的表是否存在，如果不存在，需要根据 CreateSQL 定义的 SQL 语句建立日志表，之后便可插入日志信息。

总之，该模块提供了简单的接口，用户只需要很少的步骤即可完成日志生成的功能；模块提供多种类型的日志，可以适用于多种条件，满足用户不同的要求。日志在系统中有很重要的作用，它既是程序员调试程序的重要手段，也可以为分析系统运行状态提供非常有用的数据。

3.2 Web 框架

随着互联网技术的不断发展，Web 应用已成为企业应用的主流。Web 应用进一步反映了软件分布式的趋势，同时增加了软件的互操作与可重用性[6]。目前已有许多开源的 Web 架构，如 Struts[15]，Spring[16]等。BeaconERP 为了辅助 Web 应用的开发，提供了 Web 框架模块，该模块基于 Apache 开源项目 Struts[15]，在 Struts 的基础上进行了进一步的扩展。用户可在此模块的基础上开发出架构优良的 Web 应用。

Web 应用经过多年的发展已由当初的 Model1 模式逐步发展到现在的 Model2 模式[17]。与 Model1 相比，Model2 模式可以业务逻辑层与显示层分离开，因此更符合 MVC 体系结构模式的要求，各部分之间的职责也更加明显，更符合软件工程模块化的要求[4]。Model2 模式已经得到了业界的广泛认可，并已广泛应用到 Web 应用的开发中。由于 Model2 模式的以上优点，Web 架构模块支持 Model2 体系架构下的企业应用的开发。

该模块作用主要表现在两方面：一是为 Web 应用开发者提供了一套框架，在这个框架下开发出的 Web 应用符合 MVC 体系结构模式，因而可以提高软件的质量；二是提供了一套 JSP 标记库，将与页面处理相关的复杂操作封装到 JSP 标签中，从而加快开发速度，提高开发效率。

具体来讲，该模块在 Struts 的基础上进行了三处扩展，分别是：

1. 扩展了 Struts 中的一些类。例如对 Struts 中的 Action 进行了扩展，对不同要求的操作扩展了不同的子类，提供不同的权限控制方法，既保证系统安全，又最大限度的保证了性能，从而可以满足企业应用中特殊的需求。

2. 提供了自定义的 JSP 标签，封装了较为复杂而又常用的功能。例如权限检测标签检测可以从 Session 对象中获取用户的角色和相应权限，如果用户权

限检测通过，继续执行页面中剩余的部分，否则直接转向错误页面。

3. 提供了包含常用功能的类与工具。例如常用的数据验证类，常用的数据转换类，页面菜单生成类等。

该模块的大体工作流程如图 3-3 所示。客户通过浏览器向应用服务器发送请求，请求首先被服务器端控制器接收到。控制器为一个 Servlet，它处理请求的方式是首先是实例化一个负责处理业务逻辑的 JavaBean 对象，该对象对应 MVC 模式中的模型 (Model)。在如今较为流行的三层体系结构当中，业务逻辑通常需要从企业信息系统中存取数据，然后将数据返回给 JavaBean 中相应的成员。在此之后，又将 JavaBean 中的数据转到显示页面 (JSP)，以视图 (View) 的形式返回给客户，根据 Model2 的规范，由控制器负责选择相应的 JSP 页面。最后，页面作为响应返回给用户。

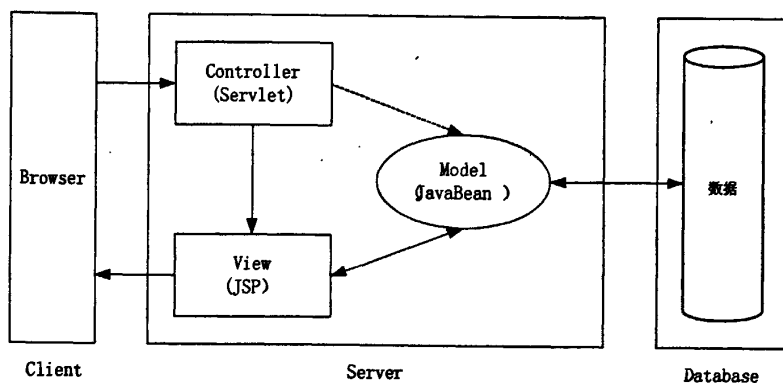


图 3-3 概要工作流程

以上工作过程是一个大体的工作流程，图 3-4 给出了一个较为详细的工作流程。具体步骤如下：

1. 用户通过浏览器向服务器发送请求。
2. 用户请求被交给服务器端控制器 ActionServlet 的实例，ActionServlet 选择相应的 ActionForm，将用户提交的数据封装到 ActionForm 中。ActionForm 有两方面的作用，一是作为数据的容器，另一方面是进行数据验证操作，如果验证通过，进行的第 3 步，否则跳转到错误页面。
3. ActionServlet 类的实例根据用户的请求及配置文件的内容，选择相应的 Action，并将第二步创建的 ActionForm 作为输入交给 Action 进行处理。
4. Action 处理用户请求，这需要访问业务逻辑层，一般需要生成 JavaBean 的对象，由 JavaBean 完成具体的操作。
5. 业务逻辑层 JavaBean 完成操作后将相应的结果返回 ActionServlet，

ActionServlet 根据返回的结果选择 JSP 页面向用户显示操作结果。

6. ActionServlet 将操作结果返回客户浏览器。

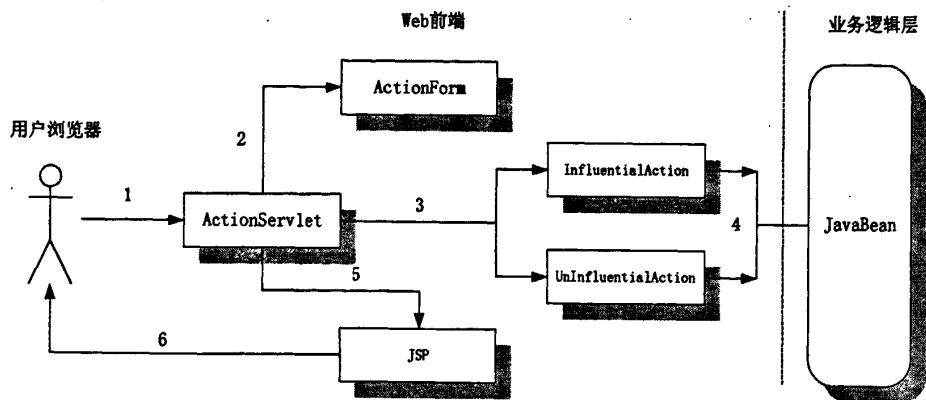


图 3-4 Web 框架模块详细工作流程

总之，Web 框架模块基于当前较为先进的 Model2 模型，它提供了一个框架，用户可在该框架的基础上开发基于 MVC 模式的 Web 应用程序；同时它还提供了一组实用工具，辅助用户开发。

3.3 权限控制

任何企业应用都离不开权限控制，它是保证企业信息安全的有效手段。传统桌面应用的权限控制技术已相对成熟，但 Web 方式的权限控制并不是一件容易的事，它通常需要涉及到服务器配置，公钥/密钥认证等复杂的访问控制技术[9]。权限控制模块的目的就是提供一些常用的权限控制技术，提高客户对权限控制模块的开发效率。

权限控制并不是单方面的控制过程，它涉及两方面基本元素，即主体和目标[10]。主体代表对目标对象访问的人或对象，通常指用户，在 BeaconERP 中指 ERP 系统的使用者；目标代表被访问对象，这里主要是指被访问的资源。由于在实际应用中，多个用户可能会扮演相同的角色，具有相同的访问权限，因此 BeaconERP 采用基于角色的权限控制策略[11]，并对其进行了改进。在模块实现时，采用基于位操作的权限控制策略，提高性能。

一个角色包含一组用户的名字和一组这些用户可以访问的资源。角色和用户是多对多的关系，角色和企业资源之间也是多对多的关系，一个用户可以扮演不同的角色，则他具有的权限是他所具有的各个角色所具有权限的并集。

BeaconERP 提供的框架基于 Web 应用,因此权限控制方案也提供了基于 Web 的权限控制策略。权限控制的工作流程如图 3-5 所示。根据 Web 框架模块的定义,控制流先经过 Action,之后经过 JSP 页面。权限控制也同样依据同样的流程。对于某些 Action,不需要权限检测。一旦权限检测失败,直接转向错误页面。

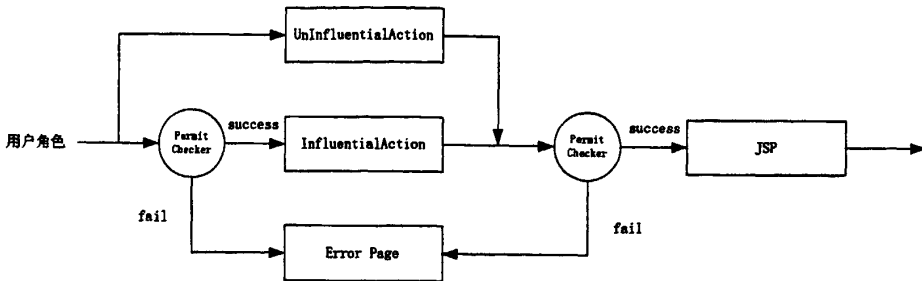


图 3-5 权限控制工作流程

下面讨论对 Action 和 JSP 的权限检测。用户请求的操作可分为两类：第一类操作不会对企业资源产生持久性的后果，例如对数据库表的查询，对某个文件的查看等；另一类操作会对系统资源产生持久性的后果，例如数据库中的记录删除，更新，表的创建，插入，删除等。对于前一类操作，只需要防止相应的信息被用户看到，即使用户执行了这类操作也不会造成信息的泄漏。这类操作不需要对 Action 权限控制，只需要对 JSP 权限控制；而对后一类操作，相应的操作一定要被阻止，同时结果也不能被用户看到，这样才能保证信息安全。因此要对 Action 和 JSP 进行权限控制。对这两类操作在权限控制模块实现过程中分别设计了不同的类。能对企业资源产生持续性后果的操作对应 InfluentialAction 类，该类中定义了相应的检测权限的方法；不产生持续性后果的操作对应 UnInfluentialAction 类，该类没有权限控制的方法。这两个类都是 Action 的子类。

下面介绍权限控制的具体算法。由于是 Web 方式的权限控制，可以从 Session 对象中获得用户的登陆信息，在数据库相应的表中查询获得用户的权限信息，作为一个属性存入用户 Session 对象中，以备之后进行相应的权限检测操作。实际的权限检测操作由 PermitChecker 类完成。检测的规则是：如果检测通过，继续执行下面的操作，如果失败，直接定向到错误页面。

ERP 系统的用户被分成了多个角色，BeaconERP 为每个角色指定了一个权限控制码，这是一个整数，并且是二的整数次幂，如 1, 2, 4, 8 等。用户也被指定了一个权限控制码，它是用户所具有的所有的角色所对应权限控制码之按位或，它的计算公式如(3-1)所示，其中 userCode 是用户的权限控制码，roleCode₁, roleCode₂ ... roleCode_n 是用户所具有的各个角色的权限控制码，n 是用户所具有

的角色的个数。

$$\text{userCode} = \text{roleCode}_1 | \text{roleCode}_2 | \dots | \text{roleCode}_n \quad (3-1)$$

例如，假设 ERP 系统中，系统管理员角色对应的权限控制码为 1，经理角色对应的权限控制码为 2，普通雇员角色对应的权限控制码为 4，客户角色对应的权限控制码为 8，那么具有且仅具有客户和普通雇员角色的用户，对应的权限控制码为 $4 | 8 = 12$ 。

每个被访问的资源也被指定一个权限控制码。资源既可以是 JSP 页面，也可以是能产生持续性后果的 Action。该资源的权限控制码是可以访问该资源的所有角色所对应的权限控制码的按位或。它的计算公式如 (3-2)。其中 resourceCode 是资源的权限控制码， $\text{roleCode}_1, \text{roleCode}_2 \dots \text{roleCode}_m$ 是可以访问该资源的各角色对应的权限码。m 是可以访问该资源的角色的个数。

$$\text{resourceCode} = \text{roleCode}_1 | \text{roleCode}_2 | \dots | \text{roleCode}_m \quad (3-2)$$

继续上面的例子，如果某个资源仅可以被经理，系统管理员这两个角色的用户访问，那么它的权限控制码为 $1 | 2 = 3$ 。

用户和资源的权限控制码定义好之后，就可以进行权限控制了。假设某个用户的权限控制码为 userCode，某个被访问资源的权限控制码为 resourceCode，那么用户能通过权限控制的充分必要条件是：

$$\text{userCode} \& \text{resourceCode} \neq 0 \quad (3-3)$$

其中“&”表示按位与运算。下面再举一例，如果某资源仅可被具有普通雇员和经理角色的用户访问，它的权限控制码为 $2 | 4 = 6$ ，某用户仅具客户角色，它具有的权限控制码为 8，由于 $6 \& 8 = 0$ ，该用户可没有访问该资源的权限；如果某用户同时具有普通雇员和客户角色，它的权限控制码为 $4 | 8 = 12$ ，那么 $6 \& 12 = 4 \neq 0$ ，说明该用户可以通过权限检测。

权限控制模块提供了当前最为流行的基于角色的权限控制技术，同时采用基于位操作的实现，最大限度地提高了性能。系统实现了该机制的基本框架，用户只需要很小的开发工作，就可以很方便的实现自己的权限控制系统。

3.4 O/R 映射

当今多数企业应用采用三层体系结构[7]。如图 3-6 所示，在三层体系结构下，表示层，业务逻辑层与数据层被分隔成相对独立的模块。表示层负责与显示相关的逻辑，业务逻辑层完成具体的应用业务，数据层负责数据的保存与维护。这种结构使得系统各部分的功能相对单一，实现起来较为容易，同时也有较高的灵活性和安全性。在三层体系结构的应用中相邻的层之间交换信息也是非常普遍的。例如业务逻辑层通常都有这样的功能：按照客户的请求，从数据库中查询相应的记录，然后将查询结果返回给客户。这种操作在实现时需要将数据库的记录转化成对应的 Java 对象，再返回给业务逻辑层。这样的操作具有普遍性，但完成它们通常是一件枯燥并且容易出错的事情。



图 3-6 三层体系结构

为了解决上述问题，核心组件层提供了 O/R 映射 (Object/Relational Mapping) 模块。“O”代表“Object”，指 Java 对象；“R”代表“Relational”，指关系数据库。因此从模块名称就可以看出该模块的功能是就是辅助 Java 对象与关系数据库中的记录之间的相互转化。使得一切工作能够自动完成，并且实现细节对上层的程序员是透明的。

图 3-7 给出了 O/R 映射模块在系统中的位置。每一个 Java 类与数据库中的一个表相对应，O/R 映射模块负责它们之间的转换工作。每一个类与表之间的映射关系由一个 XML 文件配置，它规定了字段的对应关系及转换方法。此外还有一个总的配置文件，它指定数据库类型，所用的驱动程序，用户名，密码等。

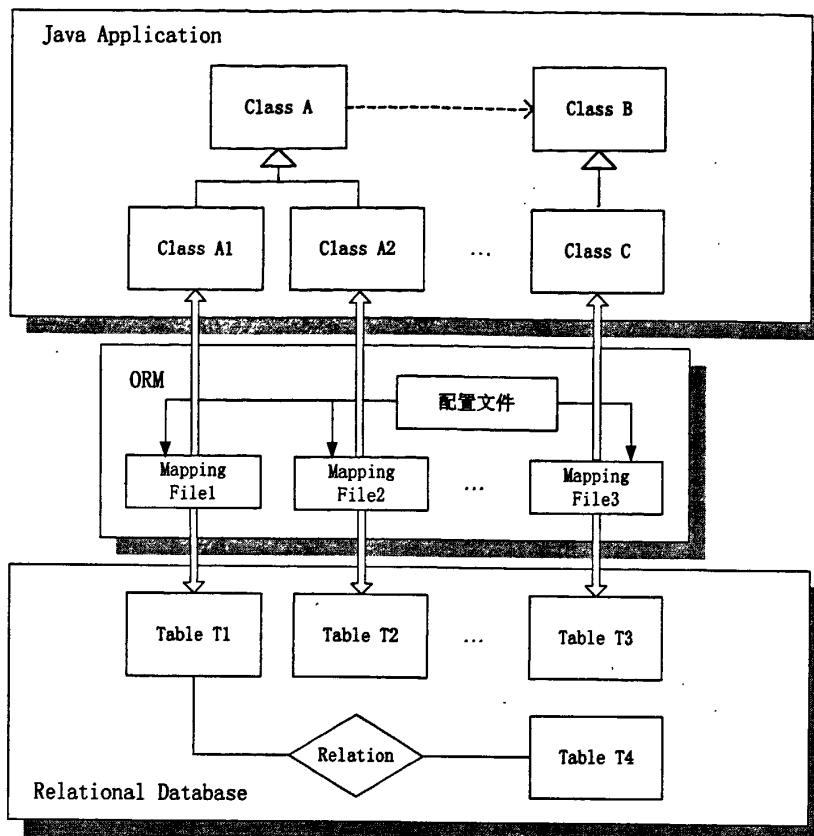


图 3-7 O/R 映射模块

ORM 模块基于 Apache 开源项目 Hibernate。该模块提供了简单的接口，可以很方便的使用其提供的功能。在程序中使用该模块之前需要完成以下工作：

- 1) 设计数据库，设计表及各字段，各字段之间的依赖关系，表之间的相互关系等。
- 2) 设计 Java 类，包括各个类的职责，属性，方法。
- 3) 对每一对 Java 类/数据库表，创建相应的映射文件。
- 4) 编写配置总配置文件。

以上四步完成之后就可以在程序中使用 O/R 映射的功能。在程序中使用 O/R 模块的步骤如图 3-8 所示。第一步创建一个 Configure 类的实例，它负责根据总体配置文件的内容完成 O/R 映射的总体配置。第二步将所有的的映射内容依次添加到 Configure 类的对象中。之后，先要打开会话然后才能开始事务，在事务中，数据库记录的提交回滚操作同样可以作用于 Java 对象，Java 对象与数据库记录之间始终保持一致的关系。当所有操作执行完毕就可以提交事务和结束会话。

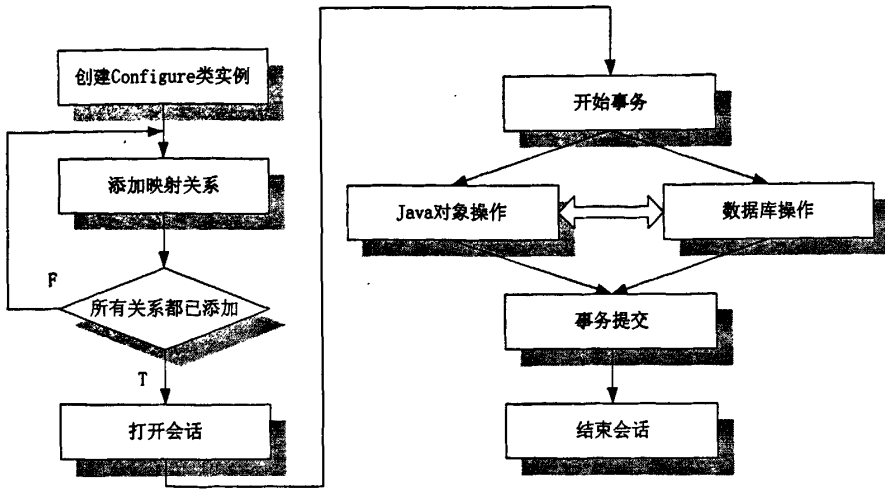


图 3-8 ORM 操作步骤

为了将开发人员从类对象/数据库记录转换这一枯燥而又不可避免的工作中解脱出来，系统提供了 O/R 映射模块。用户只需要修改配置文件，加入相应的属性，这样的转换工作便可自动完成。

总之，核心组件层提供了一些最基本的功能，这些功能在多数企业应用中都会涉及到，在以下章节的介绍中将可以看到基本组件层和扩展组件层的模块都基于核心组件层提供的最基本的功能。

第四章 基本组件层

核心组件层提供了最基本的功能,这些功能还不能完全满足 ERP 开发中涉及到的功能。为了有效的支持 ERP 应用的开发,系统还需要一些更为高级的功能。当今企业生产经营离不开报表,通过报表企业可以对上一阶段工作进行有效总结,统计相应数据,从而为计划下一阶段生产经营提供依据。由此可见,有必要针对报表提供相应的模块,满足企业对报表的相应功能的需求。由于近些年来企业内部文档数量呈现飞速增长的趋势,因此有必要提供相应的功能,快速的从大量的文档中检索出用户所需要的数据,此外搜索功能也是报表管理的一个方面。当今企业应用分布式的趋势越发明显,为了在分布式的系统之间传递事件与数据,系统还应当提供消息传递的功能。

以上功能在基本组件层实现,为每个功能分别提供了相应的模块。基本组件层在核心组件层的基础上提供了更为高级的功能,但这些功能仍然较为基本,属于通用功能。

4.1 报表系统

如今越来越多的企业已经意识到报表在企业进行信息化建设中的重要作用。通过报表,企业决策者能够及时总结上一阶段生产经营中的问题,准确地了解企业的生产经营现状,并根据市场情况确定下一步生产经营方案。报表系统针对这些需求提供了与报表相关的常用功能。

当前主流的生成报表工具主要有两大类:纯 Java 报表工具,和非 Java 报表工具。可以在 Java 程序中调用非 Java 报表工具。非 Java 报表工具产品有很多,它们又可分为两类:一类采用独立报表服务器,如 Crystal Report[18],Brio[19]等;另一类在前端有控件,如数巨报表等。

这两类非 Java 报表工具有各自的优点,同时也存在着不同的问题。独立报表服务器的问题在于:首先独立的报表服务器,成为报表的性能瓶颈;其次无法共享 Web 服务器的集群能力;此外独立报表服务器无法共享 Web 服务器的连接池管理能力;最后独立报表服务器无法统一部署,实施人员除了要部署自己的 Web 应用,还需要部署报表服务器。前端控件方式的问题在于:首先用户部署不方便,每台机器都要安装控件。此外控件会对大报表产生性能问题,因为其报表生成与

计算是在控件中实现的。再次，应用系统无法灵活地通过 API 对报表进行控制，而只能通过控件。最后产品升级比较困难，客户端需要重新下载新的控件。

鉴于以上原因，在 BeaconERP 中采用了纯 Java 的报表工具。又在纯 Java 报表工具中选择了 JasperReport[20]，它是开源报表中最流行的一个报表生成工具。除了最为流行之外，采用 JasperReport 还有一个非常重要的原因就是 JasperReprot 提供了完备的功能，可满足大多数企业与个人对报表的要求。它可将报表内容显示到屏幕上、输送到打印机或输出到 PDF, HTML, XLS, CSV 和 XML 等文件。它完全由 Java 实现，可以方便的在各种 Java 应用中使用。

该模块应当满足各种常见环境下不同用户和系统的对报表的需求，例如，应当满足在 Web 与普通 Java 程序的环境下都可以调用模块生成报表的功能。因此报表系统模块应当提供以下两种生成报表的功能：

1. 通过报表模板。JasperReport 提供了报表模板，用户可以手动修改报表模板，但这样通常效率较低，同时也容易出错。因此，可以使用 iReport 工具以图形化的方式辅助生成编辑报表模板，来定制满足特定需求的报表模板。之后在程序中设置数据源和报表类型，即可生成相应的报表。

2. 通过 Web 服务器。报表服务可以网络的方式提供报表功能。通过这种方式生成报表需要将报表服务部署到 Web 服务器中，之后启动 Web 服务器，用户即可以通过浏览器的方式查看或者打印需要的报表。

图 4-1 所示给出了报表系统的总体架构，由此可以得出报表系统生成报表的工作流程。

1) 解析模板文件：由 JasperCompileManager 解析报表模板文件的内容。报表模板文件基于 XML 格式，生成一种二进制格式的中间文件，解析过程由 compileReport 方法完成。

2) 编译中间文件：对第一步生成的中间结果文件进行编译，该任务由 JRCompiler 完成。编译生成另一种中间文件，这种文件可以被填充数据，之后生成报表。

3) 填充数据：由 JasperFillManager 完成。它需要接收三个输入：一是在第二步生成的可填充数据的中间文件；二是数据源，因为要从数据源提取数据；三是一些生成报表的参数。数据填充完成之后，即进入可打印状态。

4) 输出报表：根据所得报表目的地的不同，分别采用不同的生成方法。

4.1) 对需要打印的报表（如在打印机或控制台上打印），由 JasperPrintManager 负责打印过程。

4.2) 对于需要导出的报表（例如，需要导出到文件的报表），由 JasperExportManager 负责导出到相应的目的地。

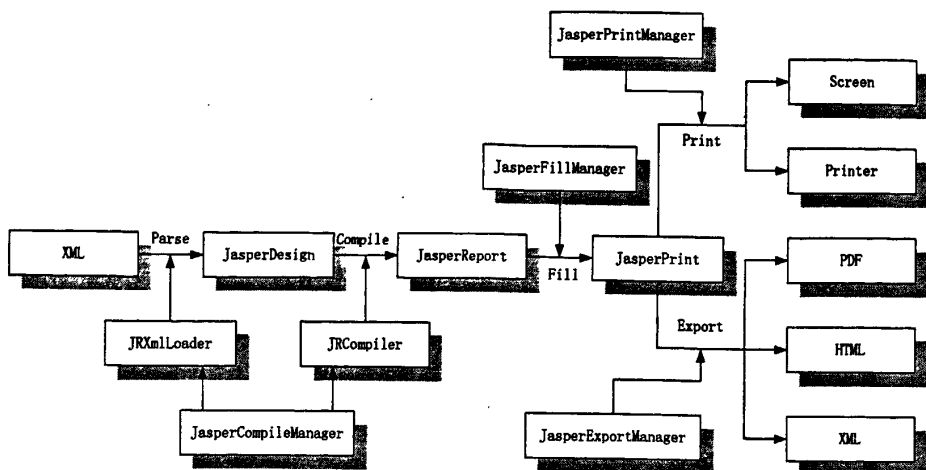


图 4-1 JasperReport API 结构

以上过程步骤较多，较为繁琐，但用户在实际使用时，并不需要这样繁琐的步骤。BeconERP 系统的报表系统模块已经提供了一个框架，代替用户完成以上大部分的操作。例如，用户需要从数据库中获取相应的记录信息，生成报表，那么他只需要修改配置文件，指定数据的来源，和相应的字段，报表系统会自动完成解析编译等工作，生成用户所需要的报表。下面给出了报表配置文件的一个片断：

```

table = User
fields = user_id: java.lang.Integer, name: java.lang.String, birth_date:
java.lang.Date

```

以上片断说明，需要从数据库的 User 表中提取信息，生成报表。具体来讲，报表的内容关于 User 表中的三个字段信息：user_id, name, 和 birth_date，它们的类型分别为整型，字符串和日期。配置文件还可以指定其它更为复杂的信息，如数据源 URL，数据库用户名，密码等。

报表系统提供了企业涉及到报表的常用功能。该模块提供了常用类型报表的生成，同时该模块具有较强的扩展性，用户可以在框架的基础上加入对其它类型报表的支持。

4.2 搜索系统

近些年来，随着存储技术和信息技术的飞速发展，企业内部数据量也以惊人的速度增长。除此之外，企业每天生产经营中要生成大量的报表与日志信息。如何管理与利用这些信息就成为摆在企业面前的一个问题。要解决这些问题通常要求从大量的数据中快速检索得到用户需要的信息，即搜索功能。搜索功能又可分为两类：一类是精确匹配搜索，要求搜索的结果与查询关键字完全匹配，数据库的查询语句可以提供这种功能；另一类是模糊匹配搜索，不要求结果与查询条件完全一致，只要信息与查询条件相关即可，满足这种需求需要一个搜索引擎。

目前商用搜索引擎的发展已较为成熟，如 Baidu, Google 已经在全世界广泛使用。近几年搜索引擎的发展趋势是个性化，即对不同用户不同应用背景提供不同的功能。由于企业有其自身的特殊背景，针对企业的搜索引擎也应当是一个个性化的搜索引擎。目前有一些针对个性化搜索引擎的研究，但是这些搜索引擎往往是根据特定的搜索内容而加入个性化的内容，如[21][22]，它们还不能适应于企业的特殊背景。BeaconERP 的搜索系统模块提供的个性化功能可在一定程度上满足企业个性化的需求。

搜索系统的整体架构如图 4-2 所示。该模块总体分为两部分：索引子模块（图中左边部分）负责解析文档，创建索引，维护索引等。查询子模块（图中右边部分）完成实际的查询，它接收用户查询请求，解析用户查询请求，将查询请求提交给查询器进行查询，最后收集查询结果，将结果返回给用户。索引以文件的方式存储在硬盘上，是这两个模块共同操作的对象，同时也是他们之间的桥梁，它的存放位置可由配置文件配置，它的内容根据企业内部文档的变化而动态改变。下面分别介绍各个子模块。

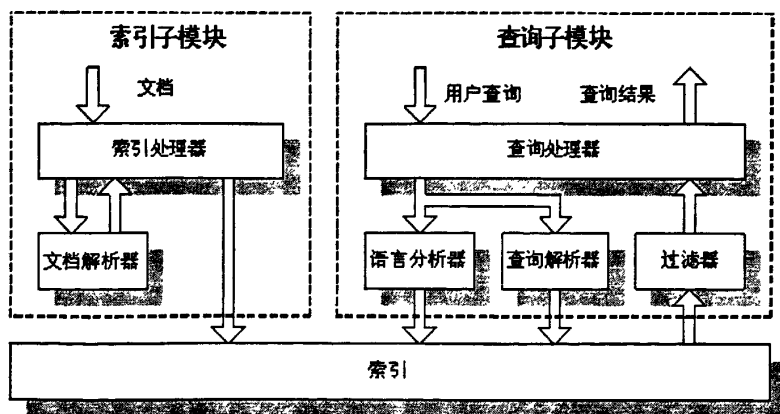


图 4-2 搜索模块架构图

4.2.1 索引子模块

索引子模块负责创建维护索引。图 4-3 给出了该模块的体系结构。创建索引只需在系统启动时进行一次，而维护索引需要随着系统文件的更新而不断重复。该子模块的总控制器是索引处理器，在系统启动时，它创建索引，之后启动索引更新线程动态维护索引。索引更新线程在后台运行，定期更新索引，更新的间隔可由用户设定。更新索引主要需要根据文件系统的变化及时对索引进行更新 [23]，具体来讲需要处理以下三种情况：

1) 修改过的文档：通过比较文档的最后修改时间与上次更新索引的时间，以判断文档是否进行过修改。若没有修改过，则无需进行任何操作，否则，应当将文档原来的内容从索引中删除，将更新后的内容重新写入到索引中。

2) 新建的文档：索引更新线程维护着一个列表，记录着所有经过索引的文档的全路径名，如果某个文档不在该列表中，说明它是新建文档，就应当对其解析，将其内容加入到索引中。

3) 已删除的文档：不仅需要将其内容从索引中删除，还要从索引更新线程所维护所有经过索引的文件列表中将该文件对应的项删除。

图 4-3 中虚线框中的部分直接对索引进行操作。解析一个文档，并将其内容写入索引的步骤如下：将文档被交给文档指派器，由文档指派器根据文档的类型选择一个适当的文档解析器，对其进行解析，指派规则可通过配置文件设定，文档内容解析之后，由索引器将内容写入到索引中。BeaconERP 系统中，文件类型的判定根据文件后缀名。不同文件类型由不同的解析器解析，但所有解析器涉及的操作大致包括以下三个：

1) 将文件内容从二进制格式转换成文本格式；

2) 从文本流中过滤掉无用信息，如果滤掉 HTML, XML 文件的标签等，提取有用的元信息，如从 PDF 文件中提取作者，标题等；

3) 将处理过的文本流划分成一个个标记 (Token)。

并不是所有的解析器都需要完成以上三个步骤，例如 HTML 文件本身就是文本格式的，因此 HTML 解析器不需要第一个步骤。

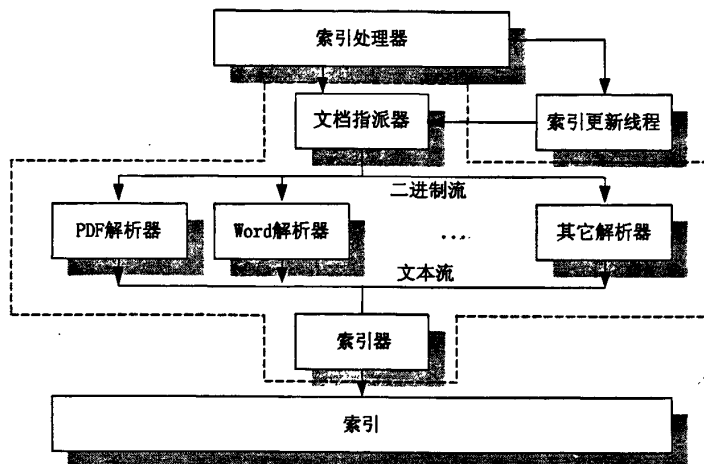


图 4-3 索引子模块体系结构

文档指派器根据指派规则为不同类型的文档分配不同的解析器，指派规则可以通过配置文件配置。为了方便用户解析常见类型的文档，BeaconERP 提供了大多数常用文件类型的解析器，但实际当中往往还有企业内部定义的文件格式，BeaconERP 对这一类文件也可提供支持，用户可以扩展 BeaconERP，开发自己的文档解析器，从而对企业特定的需求。

BeaconERP 系统搜索模块目前提供了 Word, PDF, Excel, HTML, XML, TXT 等常见类型文件的解析器。无论是 BeaconERP 系统还是用户自己提供的解析器，都必须实现 BeaconERP 提供的 API 接口 FileExtractor 及其相应的方法，才能加入到系统框架中。

4.2.2 查询子模块

查询子模块负责实际的查询工作，它的结构如图 4-4 所示。它的总控制器是查询处理器。查询子模块的主要工作步骤如下：

- 1) 查询处理器接收到用户查询请求。
- 2) 判断查询请求的类型，执行相应的操作：
 - 2.1) 以字符串方式的请求提交给语言检测器。
 - 2.2) 以查询表达语言 (SEL) 方式的请求提交给查询解析器。
- 3) 用户请求被解析后封装成查询对象然后交给查询器。
- 4) 查询处理器负责将与用户权限对应的过滤器与查询器绑定。
- 5) 由查询器执行查询操作，将查询结果返回给查询处理器
- 6) 查询处理器向用户返回查询结果。

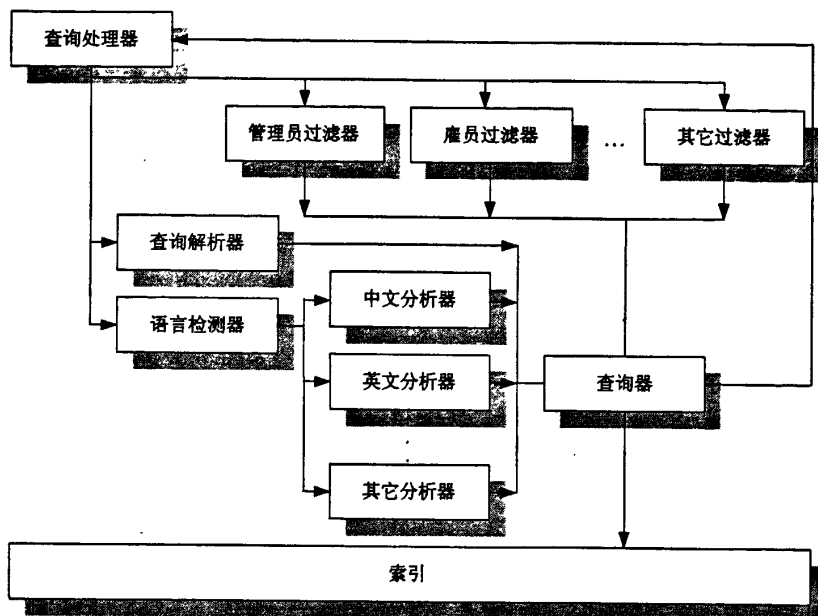


图 4-4 查询子模块结构图

各部分详细介绍如下：

1) 查询处理器

在一次查询操作中，首先由查询处理器负责接收客户请求。客户请求可由两种方式提交：一是普通的查询字符串，这适合于大多数的简单查询；二是通过查询表达语言（SEL）提交，这种方式主要适用于高级查询。查询处理器根据查询请求提交的方式，选择相应的模块进行处理。字符串方式的查询提交给语言检测器；SEL 方式的请求提交给查询解析器。

在查询对象被提交给查询器之前，查询处理器还应当获取用户的权限信息，创建相应的过滤器，将过滤器与查询器绑定，这样可以预先过滤掉用户不具有访问权限的资源，从而保证了企业内部的信息安全。一次查询操作的最后，查询处理器从查询器获取查询结果，返回给用户。

2) 查询解析器

如今的大多数商用搜索引擎都提供了两种类型的查询：一种是简单查询，这种查询只根据需要提供查询关键字组成的字符串就可以准确的表达；另一种类型是高级查询，它提供了更复杂的功能，例如查找所有作者为“Jack”，内容中包含“Program”的文档，这种类型的查询仅通过字符串表达并不是一件容易的事，需要更复杂的表达方式。当前大多数搜索引擎提供了高级搜索的功能，但这大多是硬编码实现的，如 Google 和 Baidu 的高速级搜索功能都已经硬编码在网页中，用户无法定制自己的高级查询功能。

为了克服以上缺点, BeaconERP 提出了查询表达语言 SEL (Search Expressing Language), SEL 基于当前流行的 XML 技术, 易学易用, 用户可以非常方便的创建与编辑。下面的片断给出了一个 SEL 的完整例子, 由此可见, SEL 查询由多个查询项以及各查询项之间的逻辑操作组成, 一个查询项代表一个查询条件, 查询项的逻辑操作表明该查询项是必需的, 可选的还是禁止的。

```
<?xml version="1.0"?>
<queries>
  <query operation="and" field="content">
    shell
  </query>
  <query operation="or" field="content">
    script
  </query>
  <query operation="not" field="content">
    documentation
  </query>
</queries>
```

例如片断中的第一个查询项为:

```
<query operation="and" field="content">
  shell
</query>
```

该查询项内容为“shell”, field 属性为“content”, 表示文档的内容字段中含有“shell”, operation 属性为“and”表示该查询项是必需的。因此, 该查询项表示匹配的文档中必须在内容字段含有关键字“shell”。查询操作字段对应的值及其代表的含义如表 4-1 所示。

表 4-1 operation 字段值及其含义

字段值	含义
and	必须
or	可选
not	禁止

查询处理器将以 SEL 方式提交的查询请求提交给查询解析器, 查询解析器解析 SEL 中的内容, 将其封装成查询对象提交给查询器。该模块对 SEL 的解析基于开源项目 Dom4J, 由 Dom4J 提供的 XML 解析引擎解析 SEL 的内容, 之后将 SEL 的内容封装成查询对象。

3) 语言检测器/分析器

随着全球经济一体化趋势的逐步加强, 与以往相比, 当今企业处在国际化的环境中, 企业内部需要使用多种语言, 因此搜索引擎应当提供一种框架[24], 对不同语言提供搜索功能。对不同语言提供搜索首先要能够检测出查询来自哪一种语言。语言检测器的作用正是根据查询字符串的内容检测字符串所对应的语言。之后, 就可以根据语言类型将查询字符串交给相应的语言分析器, 对不同的语言进行不同的分析。

语言检测有手动监测和自动检测两种方式, 手动检测由用户确定查询字符串使用的语言, 这种方式需要人工干预, 自动化程度较低。自动检测由系统检测查询字符串对应的语言, 这种方式自动化程度较高, 但是由于许多语言都有相同的字符集, 例如多种语言都使用拉丁字母, 因此通过字符集判断文本所属语言并不容易。当前最先进的研究也仅仅是在预先知道范围的情况下, 检测字符串对应的语言。

大体上讲, 语言自动检测的方法分为两种, 第一种方式通过文本的脚本图像来检测, 根据脚本图像的字体形状或光学密度的分布规律来检测; 第二种方式根据字符串的字符编码来检测, 例如通过判断字符串使用了 UTF-8 字符编码还是 GB-2312 字符编码可以区分查询串是英文还是中文。第二种方式对于检测像中文和英文这两种编码方式差别较大的语言有较高的效率和准确性。

目前, 搜索系统模块实现了通过字符编码的方式自动检测中英文两种语言, 用户可在框架上加入扩展以支持其它的检测方式。

语言分析器负责分析查询字符串的内容, 将其封装成查询对象提交给查询器。语言分析涉及到同义词替换, 常用词过滤, 归一化处理等操作, 这些操作与特定语言相关, 因此需要对不同语言实现不同的分析器, 此外还应当允许用户扩展自己的语言分析器。BeaconERP 目前提供了中英文两种语言的分析器, 同时提供了可扩展的特性, 用户可以根据自己的需要编写其它语言分析器, 加入到系统中, 以对其它语言的查询请求进行分析。

4) 过滤器

在权限控制模块曾经介绍过, 使用 ERP 系统的用户在企业中有不同的角色, 他们对企业资源具有不同的访问权限, 因此在查询过程中也会涉及到权限控制的问题, 只有实施正确的权限控制策略, 才能有效保证企业信息安全。

权限控制的根本问题是谁可以或不可以对那些资源进行访问。如今大多数的企业应用都采用基于角色的权限控制策略，如前所述，BeaconERP 也提供了这种方式。在搜索系统这一模块，权限控制是通过过滤器实现的。

过滤器的功能类似于一个基于权限信息的查询，它对实际查询的范围进行限制，只有满足过滤条件的文档才在查询范围之内，即只有用户具有访问权限的文档才不被过滤掉。与查询对象不同的是，过滤器与查询器是预先绑定的，这样可以避免将文档归入查询结果之后再过滤掉，从而提高了性能。

目前 BeaconERP 系统中提供了几个常见角色的过滤器，但这并不能保证在所有情况下都能满足企业的要求，因此模块提供了扩展机制，用户可编写自定义的过滤器，以满足特殊的需求。

与过滤器相关的工作流程如下：

- 1) 用户查询请求与登陆信息提交给查询处理器。
- 2) 查询处理器根据用户登陆信息获取用户具有的角色，并由此获得用户的权限信息。
- 3) 查询处理器根据用户权限生成相应的过滤器对象，并将其与查询器绑定，查询时即可完成信息过滤。

综上，搜索模块提供了 4 方面的扩展性：用户可以扩展文档解析器，以提供对企业内部文档类型的索引；可以加入语言分析器对其他语言进行分析；可以编写 SEL 文件定制自己的高级查询；也可以实现过滤器满足企业个性化的权限控制策略。

4.3 消息系统

现代企业管理往往需要涉及其上下游多方面、多行业的支持和协同，而其供应链、客户等一般来说在地理上是分散的，因此企业外部的环境是分布式的。另一方面，企业内部各部门之间采用的应用模块可能是也可能是异地、异构的，也就是说，企业内部环境往往也是分布式的。消息系统正是为了满足这样的需求而设计的。消息系统应当能够适应异构的环境，具有跨平台性，适应多种网络传输环境[10]。Java 语言对网络提供了很好的支持，因此实现的中间件和组件能比较好地满足如上要求。

企业 ERP 所连接的各部分之间一般并不是紧密耦合的，这可以使模块在发生变动时尽量减少对其它模块的影响，但另一方面，这需要不同模块之间传递消息以相互协作。消息系统可以为这种松散耦合体系结构的应用提供支持。

4.3.1 传输模式

从不同的角度，可对消息进行分类，例如根据收发消息的设备，可将消息分为手机消息，电邮消息等；按照接收与发送消息在时间上的跨度，可将消息分为同步消息与异步消息；按照消息的内容，可将消息分为文本消息与二进制消息。在此，将根据消息接收与发送方的数量，将消息分为点对点消息与组播消息。

1. 点对点消息只有一个发送者和一个接收者，日常涉及到的大部分消息都属于这种方式。这种方式对网络带宽的消耗较少，但它提供的功能也相对较弱。一般主机的 IP 地址属于点对点方式的 IP 地址，只需要将消息的目的地址设置为目标主机的 IP 地址，就可将消息唯一地发送到目标主机，实现点对点的消息传递方式。

2. 组播传递方式的消息有一个发送端和多个接收端，在发送者发送消息之后，所有的接收者都可以接收到同样的消息。这种方式提供了较为强大的功能，但对网络带宽的消耗也较大，因为路由器需要转发一些根本没用的数据包与信令消息。在目前的 IP 协议中，D 类 IP 地址保留为组播传送的地址，它的范围是 224.0.0.0~239.255.255.255[25]，只需要将消息的目的地址设置为相应的组播地址，则凡是加入到该组播组中的主机都可接收到相应的消息。点对点方式与组播方式的区别如图 4-5 所示。

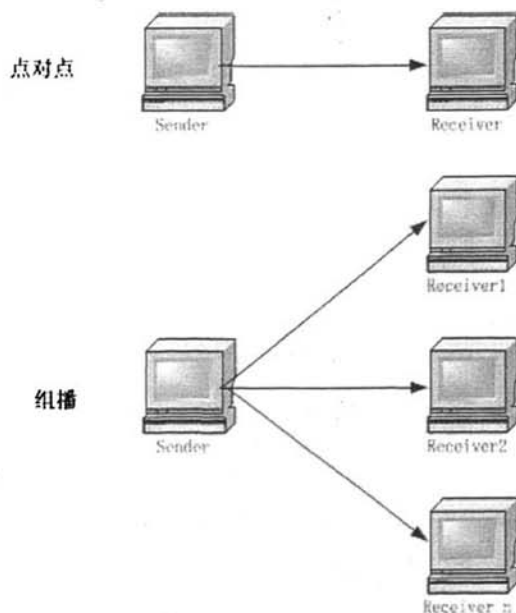


图 4-5 点对点方式与组播方式的区别

按照消息发送与接收在时间上的分布,将消息分为同步消息与异步消息。这两类消息在实际当中也有广泛应用,例如手机通话属于同步消息,而手机短信属于异步消息。

1. 同步消息:消息的发送与接收在时间上的差别较小,通常时间差仅为网络延时,发送方与接收方同时进行消息的收发操作,当消息发送/接收成功,发送/接收方可以继续其它的任务。同步消息的实时性较强,但过多的同步消息会造成系统吞吐率的下降,因为消息的发送与接收方都必须阻塞等待直到消息的收发完成。

2. 异步消息:消息发送与接收的时间差较大,通常消息发送方将消息发送完成较长时间后接收方才能接收到消息。这种方式不要求收/发双方阻塞等待直到消息发送的完成,发送方只须将消息放入缓冲区中,既可继续其它任务,接受方在空闲的时候将消息从缓冲区中取出,即完成了一次消息传递过程。由此可见,异步消息对系统吞吐率的影响较小,但另一方面,它需要缓冲区存放消息,从而需要较大的内存空间。

按照消息的发送方与接收方的不同,还可以将消息分为人-系统消息,系统-系统消息等。人-系统消息收发方一方为人,即用户,另一方为系统,这种消息通常为由用户操作引发的事件;系统-系统消息的发送者接收者均为系统,不同的系统既可以在同一台主机上,也可以分布在地理位置不同的主机上,通过网络进行通信,系统-系统消息通常代表了系统不同模块之间的一次协作或交互。本模块针对不同消息的特点,分别提供能够了不同的实现,从而能够满足用户不同的需求。

4.3.2 系统实现

消息系统基于开源项目 JGroups[26],可以适应不同的网络环境下的消息传递,只需要修改配置文件即可。本模块实现了各种消息传递模式:点对点/组播模式,可靠/不可靠的消息传递模式等。组播方式基于 IP 组播,但为了进行可靠的组播,在 IP 组播上加入其它功能,防止丢包,重传或乱序。此外,还提供了对通信群组的管理功能,对于群组通讯中的每一个发送/接收者,可以很容易的获得通信群组各个成员的信息。

消息传递依赖于一个称为通道的组件(如图 4-6 所示),通道封闭了底层的通信细节,如丢包重传,包重复等等。需要消息传递的模块只需要设置通道的相应属性即可在该设置的假设下进行消息传递。

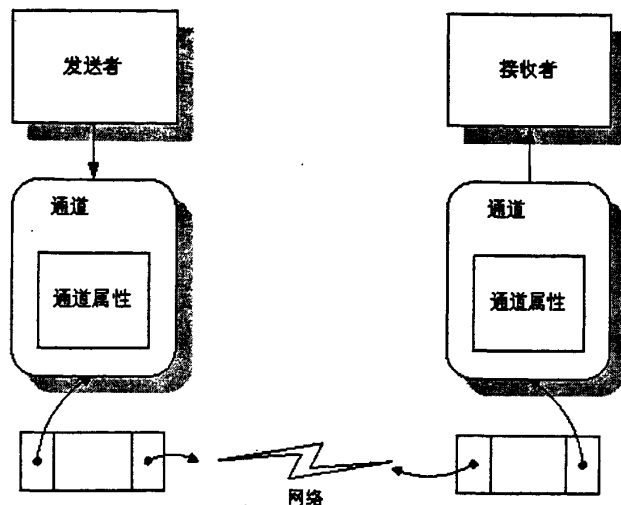


图 4-6 消息传递模块实现

使用通道收发消息也同样简单。在使用之前应当先编写好配置文件，配置文件基于 XML 格式，配置的内容包括网络地址，端口等。具体配置属性名称及涵义如表 4-2 所示。

表 4-2 消息传递模块配置文件

名称	作用
mcast addr	组播地址
mcast port	组播端口号
ucast send buf size	点对点发送缓冲区大小
ucast rcv buf size	点对点接收缓冲区大小
mcast send buf size	组播发送缓冲区大小
mcast rcv buf size	组播接收缓冲区大小
ip ttl	IP 包生存期

使用通道收发消息需要以下几个步骤：

- 1) 创建通道对象，在该模块中，通道对象的创建须遵循工厂模式[27]。
- 2) 载入配置文件，对通道进行配置。
- 3) 加入道通通信群组。
- 4) 绑定发送/接收消息的对象。
- 5) 开始通道监听。
- 6) 之后，发送/接收消息的对象即可通过通道收发消息。

由以上内容可见，消息系统是为了企业与内外部分布式模块之间协作而设计

的，它提供了多种消息传递模式，能适应多种网络环境，从而可满足多种消息传递环境的需要。

本章介绍了基本组件层的模块及其相应的功能，这些功能功能较为高级，但仍属于通用功能，与具体应用无关，与具体应用相关的功能位于它的上一层——扩展组件层。

第五章 系统其它组件

本章介绍系统中剩余的组件，包括三部分：扩展组件层，应用层和辅助开发插件。

5.1 扩展组件层

为了有效支持 ERP 功能的开发，仅靠核心组件层与基本组件层提供的通用功能还不够。ERP 已由当初的制造业发展到几乎各个领域，因此不同行业的 ERP 系统也有不同的功能，例如制造业和金融业的 ERP 系统功能就有很大差别，这就要求对不同行业提供不同的功能，这些功能位于系统扩展组件层，该层的模块主要由用户根据自身所处行业所需要的功能来扩展。

扩展组件层依赖于核心组件层和基本组件层提供更为高级的功能，这些功能与具体应用的关系较为紧密。系统也提供了一些基本或常用的功能，但这一部分主要靠用户对其进行扩展。用户可根据自己的实际情况，开发扩展组件层的组件，并加入到系统中，不断扩展 BeaconERP 提供的功能。例如，系统可以提供内容管理（CMS），协同办公等功能。

- **CMS**：内容管理模块，随着企业内部数据量的迅速增长，对文档内容的协同管理，分布的要求也越来越高，因此对内容管理的需求也不断增加[28]。在这种新的背景下，企业中的文档经历一个新的由内容创建，内容管理，到内容分布的新的生存周期[29]，其中内容管理处核心的地位。本模块在基本组件层搜索系统，报表系统的基础上，提供对企业文档内容管理的功能，对文档内容进行存储，索引，分类等[30]。如果没有内容管理系统，对企业应用的内容管理，基本上都是靠手工维护，面对着千变万化的海量信息流，这几乎是是不可能的事，而有了 CMS，这些任务可以由计算机自动完成。只需修改配置文件，剩下的事就交给 CMS 去处理。
- **协同办公**：该模块主要是在基本组件层消息系统的基础上，提供协同办公的常用功能。它为企业提供一个统一的管理平台和畅通无阻的管理通道，通过信息交互、业务流程及按需应变的软件应用方案，增强企业管理执行能力，从而为用户提供了一个全方位的信息获取、共享和使用的

协同平台。进而提高企业工作效率，同时也提升了企业的协调能力。

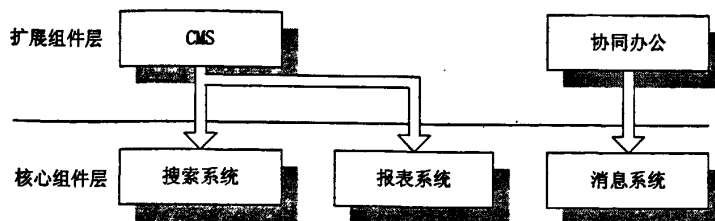


图 5-1 扩展组件层在系统中的地位

扩展组件层与核心组件层个模块之间的依赖关系如图 5-1 所示，其中图中箭头表示依赖关系。由图可见，内容管理模块主要依赖于搜索系统和报表系统两个模块；协同办公主要依赖于消息系统模块。

当今企业各部门之间的相互关系不同，它们之间的协同方式也有所不同，协同办公应当包括与日常办公有关的各种系统[31]。因此，BeaconERP 提供了多种方式的协同办公方案，用户可以根据实时性，交互性，等要求选择相应的协同方式。不同的协同方式包括：

- 文本通信：本文通信方便用户以文本方式进行交流。这种方式属于同步消息，实时性较强，同时使用也较为简单。
- 语音/视频会议：这种方式为用户提供了可视化的交流方式。它的具体功能包括语音会话，视频会话，录音，录像，拍照等。
- 共享白板：该方式为用户提供了一种类似于会议室黑板的协作方式，用户可以在画板上绘制各种图形，如文本，矩形，椭圆等等。会议结束时可对画板上的内容进行保存，会议开始进行时，可以对上一次的内容进行载入。

5.2 应用层

三层中间件的目的是为了有效的支持 ERP 应用的开发，这些应用位于应用层，是具体的 ERP 应用组件。它可分为两部分：一类是基础协作应用，它由 BeaconERP 提供，主要目的是辅助用户开发自己的应用组件；另一类是用户应用，这是用户在 BeaconERP 系统上二次开发得到的 ERP 系统。它们的关系如图 5-2 所示。

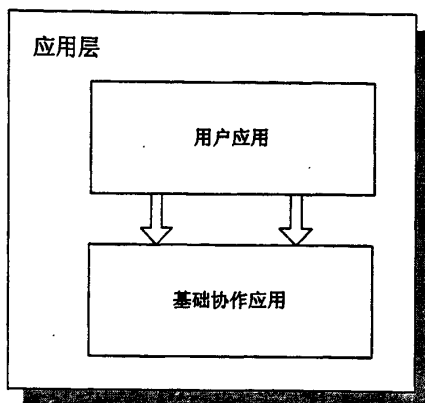


图 5-2 应用层

BeaconERP 在应用层基础协作应用部分提供了一个 ERP 系统销售管理模块的原型。该模块有两方面的目的：一是用户可以参考该原型的源代码从而进一步了解 BeaconERP 中间件系统的使用方法；二是可以在该模块的基础上增量式开发，最后演化到自己的 ERP 系统。

5.3 辅助开发插件

当今的许多系统对用户辅助开发提供的支持还不够。例如，提供了可配置的功能，用户只需要修改配置文件即可完成对功能的配置。但有时这并不是一件容易的事，多数配置文件格式较为复杂，内容也较为庞杂，如果有一处编写错误，将会对系统造成不可预知的后果。

BeaconERP 也面临着同样的问题，许多组件都是通过修改配置文件来扩展系统功能或修改系统能够原有功能，如 O/R 映射模块的映射文件；还有一部分功能需要用户手动编写一些文件来实现，例如搜索模块中的高级搜索文件等等。这些文件大都基于标准 XML 格式，且通过 XML 标签的名称即可猜出其作用，但这对于不熟悉 XML 的用户并不方便，同时手动编写文件也容易出错。

为了解决以上问题，一个比较好的办法是提供一个图形界面，辅助用户完成配置文件的编写。因此，BeaconERP 提供了一个辅助开发插件，可以方便用户对配置文件和其它辅助文件的操作，提高用户开发效率。该插件图形界面是一个 Eclipse 插件[32]，基于比较新的 SWT[33]控件库，它以 Perspective 和 View 为核心[34]，提供的功能包括以下几方面：

文件管理：对文件进行管理，包括浏览文件，打开/关闭文件，创建，删除，重命名文件等。

文件骨架生成：插件辅助生成文件的框架，用户只需在框架的基础上填入或修改相应的内容即可。

文件编辑：用户可以修改文件，同时插件提供了一些辅助编辑的功能，如关键字的高亮显示。

本章介绍的组件虽然占用的篇幅较少，但它们的作用仍然是至关重要的，这些组件直接与最终的 ERP 产品相关，因此从某种意义上讲，这些组件的质量直接关系到最终 ERP 产品的成败。

第六章 系统实现及评价

作为一个中间件系统, BeaconERP 系统的主要设计目的是为了辅助用户进行二次开发, 从而得到满足企业特定需求的 ERP 软件。因此, 为了测试系统的实际效果, 主要采用两种方式进行: 一是在中间件的基础上开发一些 ERP 示例应用, 以验证所用到的组件, 这主要是通过系统应用层组件的运行效果来体现; 二是在各组件的基础上编写一些测试程序, 以检验系统性能, 可用性等指标。对系统的评测主要通过以下几方面进行:

- 可扩展性: 是否可以较为容易地在系统原有组件的基础上添加新的模块与功能, 以满足新的需求。
- 可配制性: 通过修改配置文件的内容, 就可以重新对系统进行定制, 用户不需要编写新的代码, 就可以满足自己的功能。
- 可移植性: 系统是否不依赖于底层特性, 如所用操作系统, 数据库等, 是否可在不同平台之间移植。
- 易用性: 系统是否提供了友好的界面与良好的接口, 使用户可以较为容易的熟悉系统使用系统。
- 可靠性: 是否没有错误, 或者在有错误的情况下可以正常运行[35]。
- 运行性能: 主要是程序运行的时间与空间特征。

6.1 应用层组件

BeaconERP 在应用层提供了一个 ERP 系统销售管理模块的原型。该模块将系统中的用户定义为四种角色, 它们分别是雇员, 客户, 经理与系统管理员。系统为它们分别提供了不同的功能。它们的用例图如图 6-1 所示, 雇员, 客户, 经理与系统管理员都是特殊的角色, 他们都从同一个角色——用户扩展而来, 因此他们有一些通用的功能, 如个人信息查询, 登陆注册等。

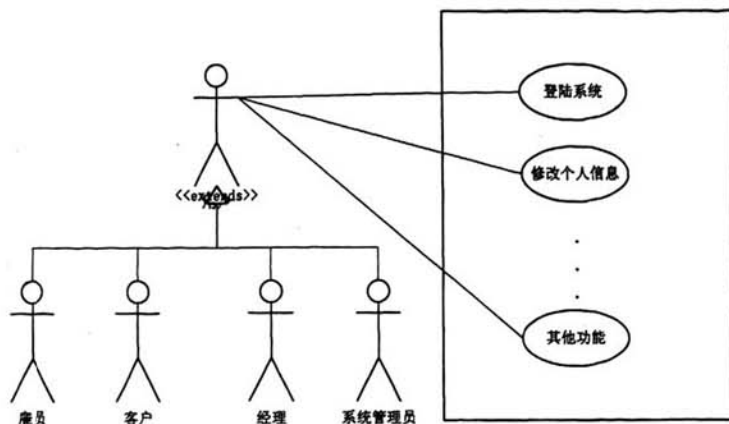


图 6-1 总体设计用例

实际运行效果如图 6-2 和 6-3 所示。首先进入系统登陆界面(如图 6-2 所示)。用户输入正确的用户名密码后,点击“提交”按钮即可登陆。没有注册的用户可以点击下面的“注册”按钮,录入相应的用户信息进行注册,新注册的用户默认角色为普通客户。

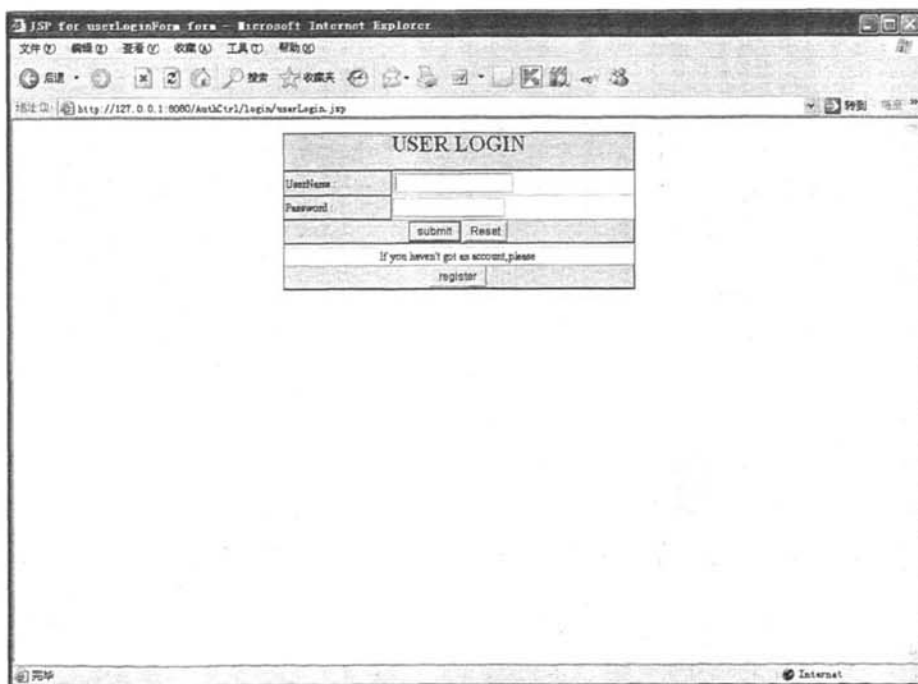


图 6-2 系统登陆界面

登陆后的系统界面如图 6-3 所示。用户可从左侧根据用户所具有的权限选择

相应的菜单，再从菜单中选择相应的功能。如果用户权限不够，权限检测将不能通过，从而导向到错误页面。各角色所具有的功能包括：

客户：系统提供的功能包括创建新的订单，查看自己提交的所有订单，了解新提交订单的当前状态，对于没有经过处理的订单，客户还可以选择修改订单或撤销订单。

雇员：系统提供的功能包括查看所有客户提交的订单，确认同意订单或确认不同意订单。对于已经过经理确认的订单，若货物已经备齐，雇员可以选择发货将货物发送给相应的客户。

经理：在系统中的功能包括：查询雇员信息，查询客户信息，雇用新雇员，解雇职员和确认已同意订单等。查询可包括通过单个关键字（姓名，地址，电话等）查询，也支持多关键字复合查询。

系统管理员：功能包括用户密码的找回，授予用户特定的权限等。

此外系统还提供了所有角色的用户均具有权限的一些功能，包括：个人密码修改，个人信息查询等。

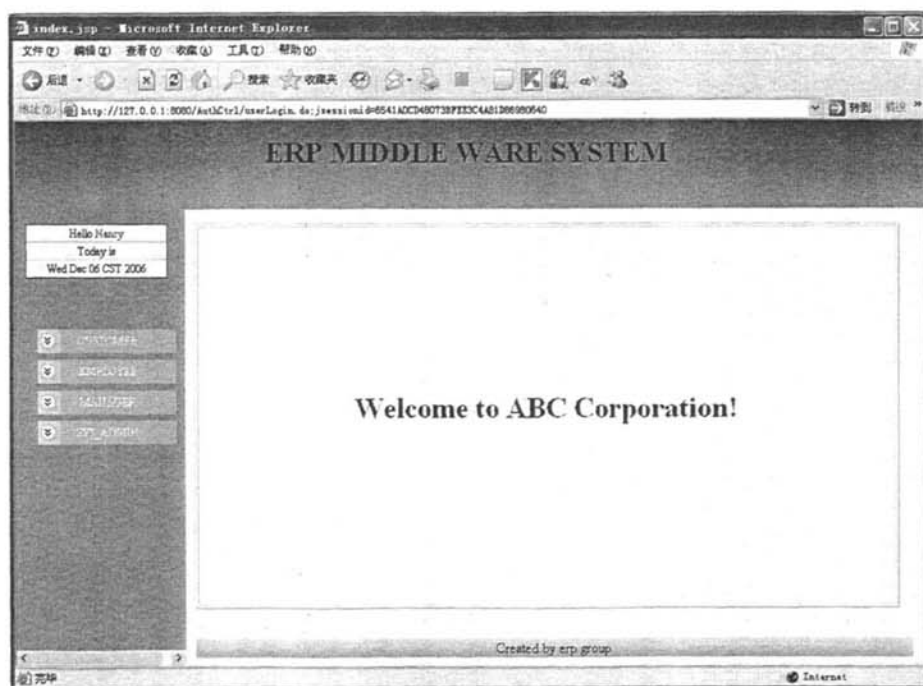


图 6-3 系统主界面

通过对该模块的运行效果的观察，发现该模块提供了 ERP 销售管理所需的大部分功能，用户可以此为基础开发自己的 ERP 应用相应的功能模块。同时该模

块具有较高的稳定性和可移植性。

6.2 中间层运行效果

本届对 BeaconERP 中间层部分（核心组件层，基本组件层，扩展组件层）主要模块的运行效果进行了展示。

6.2.1 搜索系统

运行测试是通过对本地文件系统中的文档进行索引，之后对其进行搜索。文件系统包含了本人所在项目组中所有的论文，项目文档和技术文档共 2509 个，涵盖了大多数常用类型的文档，其中各种类型文档的数量如表 6-1 所示。

表 6-1 各类文档个数

类型	个数
普通文本	109
Word	189
PDF	466
HTML	1399
Excel	28
XML	318

本人对索引模块的索引操作进行了计时，索引 2509 个文档的共用时 431.187 秒，平均每个文档索引用时 0.172 秒。这个结果说明，系统刚刚启动时，需要对所有文档创建索引，这通常需要较长时间，但这个操作只进行一次，之后对索引进行更新操作时，由于对每个文档索引所用时间较少，并且需要更新的文档个数有限，因此对索引的更新操作占用的时间较短，几乎不会对查询造成影响。这有利于系统的平稳运行。

由于两种不同方式提交的查询经过的处理流程不同，因此有必要对这两种方式进行独立的性能测试，采取的测试方法也有所不同。对于字符串方式提交的查询，需要将查询字符串提交给语言分析器进行分析，分析的时间复杂度与查询字符串长度没有直接的联系，并且查询字符串一般长度较短，差别也不大，因此对字符串方式的查询不讨论字符串对查询性能的影响；而 SEL 方式提交的查询涉及到对 SEL 的解析，这包含一些对 XML 的操作，因此提交的 SEL 的复杂度将

会对查询的速度产生影响，对这类查询的分析要考虑查询复杂度对性能的影响。

字符串方式的查询测试方法如下：随机选取一篇文章中的 1000 个字符串，分别对这 1000 个字符串进行查询，记录查询总时间，再除以 1000 得到平均查询时间，测试得到的平均查询时间是 0.0016 秒。作者采用类似的方法，对查询项个数不同的情况下，SEL 方式的查询进行了性能测试，结果如表 6-2 所示。

表 6-2 SEL 方式查询测试结果

查询项个数	查询时间（秒）
1	0.00458
2	0.00555
3	0.00582
4	0.00614
5	0.00660
6	0.00718
7	0.00756
8	0.00834
9	0.00883
10	0.00940

由以上测试结果可知，由于对 SEL 的解析所产生的开销大于语言分析器分析查询字符串产生的开销，因此 SEL 方式的查询比字符串查询需要更多的时间。查询系统的可扩展特性在查询过程加入了一些额外的环节，如语言检测，语言分析器，过滤器等，以满足企业个性化的需求，但这些额外操作所带来的开销并没有对性能产生严重的影响，即使是对速度较慢的 SEL 方式的查询而言，当查询项达到 10，平均查询时间也仅为 0.00940 秒。以上结果说明，搜索系统具有较高的性能，稳定性和可扩展性。

6.2.2 消息系统

为了验证消息系统的功能，本人开发了一个客户端，利用消息系统的功能进行一些不同用户之间的协作。图 6-4 显示了共享白板应用的界面。用户可在左侧面板中选择相应的属性，在右侧的白板上绘制内容，一个用户绘制的内容可以被协作群组中所有其它用户看到。用户可在左侧面板中选择的属性包括：绘制的内容，绘制颜色，字体（样式，大小），线形，线宽，是否填充等。绘制的内容由左上的两排按钮确定，具体按钮相应的功能如表 6-3 所示。用户可以保存白板上

绘制的内容，绘制内容以 XML 的格式存放在文件中，用户还可以选择文件对以前绘制的内容进行载入。

表 6-3 按钮功能对应关系

按钮	功能
T	文本
C	圆
E	椭圆
R	矩形
S	正方形
L	直线
A	弧
P	点

该应用的运行结果表明，消息系统具有较好的可用性与可扩展性，能为协同办公模块提供有力的支持。用户可对绘制的内容能够进行保存与载入，从而表明系统具有一定的容错性。

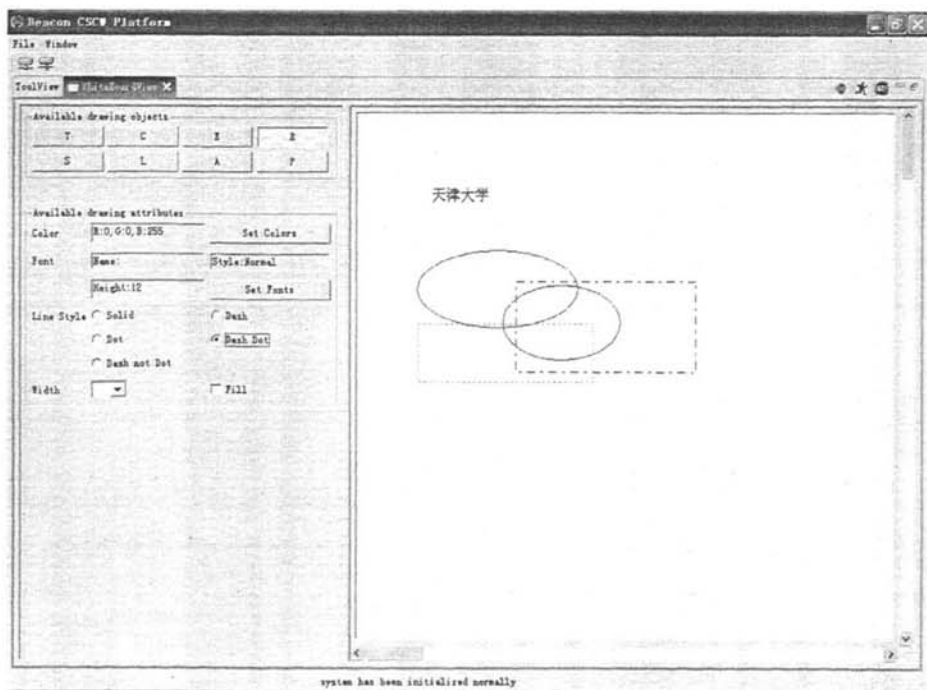


图 6-4 共享白板

6.3 辅助开发插件运行效果

BeaconERP 系统的辅助开发插件模块提供了一些功能支持配置文件及其它辅助文件的编辑。下面以搜索模块开发搜索配置文件以及高级搜索文件为例，展示辅助开发插件的运行效果。

首先新建工程，选择工程类型为高级搜索文件（Advanced Search File），之后点击下一步。如图 6-5 所示。

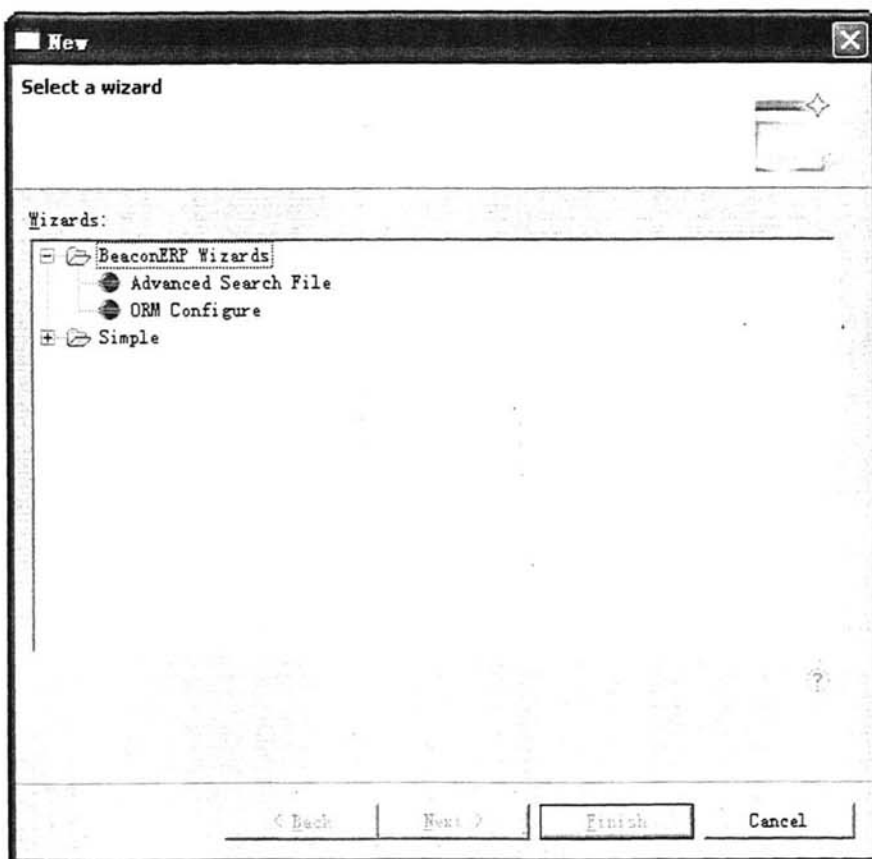


图 6-5 工程类型

在下一个对话框中输入工程名，文件名，文件目录等内容，单击完成按钮。如图 6-6 所示。

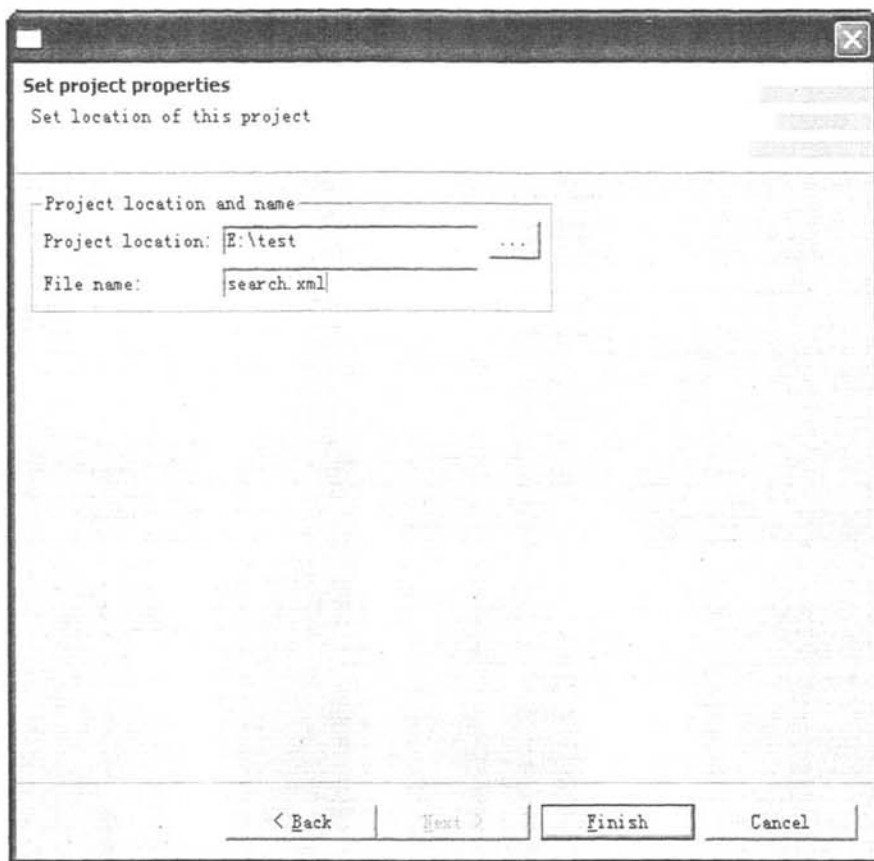


图 6-6 工程属性

这时辅助开发插件会自动生成相应的高级搜索文件以及搜索配置文件，并自动打开高级搜索文件。文件的框架已经生成，用户只需要填入相应的内容即可，为了辅助用户开发，编辑器对关键字，引号中的内容，正文等相应内容提供了不同的显示颜色。在左侧的文件浏览视图（File Navigation View）中，显示了生成的文件，可以通过视图中的树控件打开文件，删除文件或重命名文件。

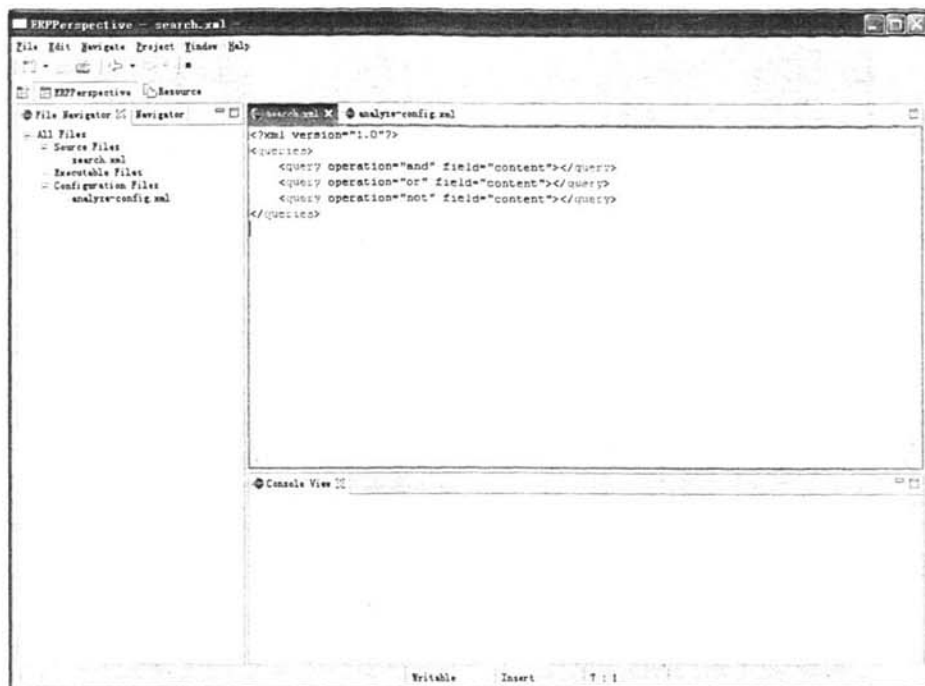


图 6-7 辅助开发插件界面

第七章 总结与展望

7.1 全文总结

本文对 ERP 的发展过程，以及国内外 ERP 的发展状况进行了分析，指出了当前 ERP 软件发展中的一些问题，例如产品功能定位和功能覆盖方面的缺陷，以及对二次开发支持的欠缺等等。

为了改进这些缺点与问题，我们开发了支持 ERP 开发的中间件与组件系统——BeaconERP 系统。该系统由一系列组件组成，组件之间采用分层体系结构，主要分为三层：

核心组件层组件提供了最底层，最基本的服务和功能。该层的组件包括：日志系统，权限控制，Web 框架和 O/R 映射。日志系统为用户提供了两种不同的日志，文件日志用来对系统进行调试，数据库日志用来监控系统运行信息；权限控制模块提供了一种基于角色的权限控制机制，对页面和 Action 进行分类，采取不同的控制策略；Web 框架提供了一个结构优良的框架和一组实用工具，用户在该框架上开发出的应用具有 MVC 的体系结构模式；O/R 映射提供了一组功能，用户只需要修改一些配置文件，就可以方便的实现 Java 对象与数据库记录之间的相互转换。

基本组件层针对 ERP 系统中比较典型的功能，提供一些基本的功能模块，这些模块的功能一般与具体应用无关，但功能相对核心组件层较为高级。该层的模块包括报表系统，搜索系统，消息系统。报表系统可根据配置文件的内容自动从数据库中读取相关信息并生成报表；搜索系统可对多种类型的文件进行索引，用户也可在此之上加入自己的解析器，该模块提供了简单搜索高级搜索两种功能，用户可编写自己的高级搜索文件；消息系统提供了多种方式的消息服务，如同步消息，异步消息，单播消息，组播消息等等。

扩展组件层提供较为高级的功能，这些功能依赖于核心组件层和基本组件层提供的服务与功能，它们与具体应用的联系也较为紧密。该层的功能可包括内容管理，协同办公等等。

7.2 未来展望

ERP 的开发是一个需求驱动的过程[36]，近些年来计算机技术又有了新的的发展，用户对 ERP 也有了新的需求，因此这些新的技术必将对 ERP 产生新的影响，推动 ERP 的进一步发展。

Internet 对 ERP 的发展产生了巨大的影响[37]，因此 ERP 未来的发展趋势首先表现在 ERP 与网络技术的集成，包括与 Web, Intranet 技术的集成[38]。随着近些年网络技术的飞速发展，这种趋势也愈加明显。在 BeaconERP 提供了对 Web 方式的支持，如 Web 架构模块，但这种支持还不够，例如近年兴起的无线技术在 BeaconERP 还没有得到体现，还无法与手机等移动终端进行交互。这将是未来工作的内容的一部分。

经过多年的发展，ERP 已经走出最初的制造业，而在能源、电信和其它更广阔的行业中寻找更加广阔的市场。这也对 ERP 软件的专业化提出了更高的要求。BeaconERP 中提供的功能多为通用功能，虽然扩展组件层主要面向专业化的模块，但在 BeaconERP 中，该层的模块数量相对较少，提供的功能也仍然较为通用，因此提供更加专业化的模块将是未来工作的内容。

随着知识经济的到来，知识和信息已成为企业获胜的最重要的资源。因此近些年有许多工作被投入到有效的开发、管理、利用知识资源上。因此也应当将其加入到 ERP 系统中，目前 BeaconERP 系统还没有提供对知识库的支持，这同样是未来研究的内容。

总之，ERP 总是处在不断发展的过程中，只有把握最新的技术动向，和最新的管理理念，才能更好的把握 ERP 的需求，提供更好的支持 ERP 开发的中间件与组件系统。

参考文献

- [1]王东迪, ERP 原理应用与实践, 北京: 人民邮电出版社, 1~5
- [2]朱春燕, 马光华, 管理新视野—ERP 与供需链管理, 北京: 中国电子音像出版社, 1999: 11~12
- [3]林健, 张玲玲, ERP 的未来发展趋势, 系统工程理论与趋势, 2002, Vol4: 69~74
- [4]郑人杰, 殷人昆, 陶永雷, 实用软件工程, 北京: 清华大学出版社, 1997, 89~96
- [5]严蔚敏, 吴伟民, 数据结构, 北京: 清华大学出版社, 1997, 13~17
- [5] Mary C. Burton, Joseph B. Walther, A survey of web log data and their application in use-based design, System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference, 2001, 1~10
- [6]Lindquist T. E, Gary K. A, Koehnemann, H. E, etc, Component framework for Web-based learning environments, Frontiers in Education Conference, 1999, Vol2, 1999, 12C3/23 ~ 12C3/28
- [7]张智远, 三层体系结构计算模式的技术特点分析, 中国传媒科技, Vol6, 2005, 25~27
- [8]Frank Buschmann, Regine Meunier, Hans Rohnert, etc, Pattern-Oriented Software Architecture Volume 1: A System of Patterns, 北京: 机械工业出版社, 2003, 18~30
- [9]Dirk Balfanz, Usable Access Control for the World Wide Web, Computer Security Applications Conference, 2003, 406~415
- [10]肖健, 可扩展的通用 CSCW 开发平台的研究与设计, 天津, 天津大学计算机系, 2004
- [11]Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman, Role-Based Access control Models, IEEE Computer, Vol29, Number 2, February 1996, 38~47
- [12]刘艳瑞, 办公自动化正在走向协同时代, 办公自动化, Vol8, 2005, 4~4
- [13]何团委, 支持 ERP 开发的中间件系统中日志模块的设计与实现, 天津, 天津大学计算机系, 2004
- [14]Log for Java Project: <http://logging.apache.org/log4j/docs/>
- [15]Struts Project: <http://struts.apache.org>

- [16] Arthur J., Azadegan S., Spring framework for rapid open source J2EE Web application development: a case study, Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005, 90~95
- [17] 王鑫, 孙小平, 谭畅, 论 MVC 模式改进 Model 2x 模型, 控制工程, Vol12, 2005, 480~481
- [18] Crystal Report: <http://www.businessobjects.com/products/reporting/crystalreports/>
- [19] Brio.Report: <http://www.mercator.gr>
- [20] Jasper Report: <http://jasperforge.org/sf/projects/jasperreports>
- [21] HAO CHEN, SHI YING, JIN LIU, et al. A Specific Search Engine for Software Components, Proceedings of the Fourth International Conference on Computer and Information Technology, 2004, 863~868
- [22] FANG WAN, YA-NAN HAO, GE YU, The Study of Key Techniques in Intelligent XML Search Engine, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, August 2004, 1194~1197
- [23] Mhom UEHARA, Change Aware Distributed File System for a Distributed Search Engine, Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference, Vol3, 2004, 1511~1514
- [24] YIP Chi Lap, Ben KAO, Indexing multilingual information on the web, Computer Software and Applications Conference, 1998. COMPSAC '98. Proceedings, 1998, 576~581
- [25] James F.Kurose, Keith W.Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 北京: 清华大学出版社, 2003, 265~267
- [26] JGroups - A Toolkit for Reliable Multicast Communication, www.jgroups.org, 2006
- [27] Erich Gamma, Richard Helm, Ralph Johnson, etc, Design Patterns: Elements of Reusable Object-Oriented Software, 北京: 机械工业出版社, 2000, 70~77
- [28] David Hausheer, Burkhard Stiller, Design of a Distributed P2P-based Content Management Middleware, Euromicro Conference, 2003. Proceedings. 29th, 2003, 173~180
- [29] Qian Zhang, Yu Sun, Zheng Liu, etc, Design of a Distributed P2P-based Grid Content Management Architecture, Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual, 2005, 339~344

- [30]Jean-Paul, Smets-Solanes, Rogério Atem de Carvalho, ERP5: a next-generation, open-source ERP architecture, IT Professional, Vol5, Issue 4, 2003, 38~44
- [31]李新, 一种协同办公系统的设计, 计算机应用与软件, Vol122, 2005, 128~130
- [32]IBM Corporation. Eclipse3.0 user guide, <http://www.eclipse.org/>, 20th June, 2002
- [33]Carolyn MacLeod, Steve Northover, SWT: The Standard Widget Toolkit <http://www.eclipse.org/> November 27, 2001
- [34]Object Technology International, Inc. Eclipse Platform Technical Overview, February 2003, <http://www.eclipse.org/articles/>
- [35]Erica Mealy, Paul Strooper, Evaluating software refactoring tool support, Software Engineering Conference, 2006, 10~19
- [36] Colette Rolland, Naveen Prakash, Matching ERP system functionality to customer requirements, Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium, 2001, 66~75
- [37] Yao Yurong, He Houcun, Data Warehousing and the Internet's Impact on ERP, IT Professional, Volume 2, Issue 2, 2000, 37~41
- [38]杨曼, ERP 发展趋势研究, 情报科学: 科学出版社, 2002, Vol19: 1002~1004

发表论文和科研情况说明

发表的论文:

[1] 樊莉亚, 孙济洲, 肖健, “可扩展高性能 CSCW 通用平台设计与研究”,
《计算机工程》, 2006 年 11 月

[2] 孙济洲, 樊莉亚, 孙敏, 于策, 张绍敏, “改进的并行高斯全主元
消去法”, 《天津大学学报》, 2006 年 9 月

参与的科研项目:

本人参与了天津市科技重点攻关项目《通用 CSCW 平台》和天津市科技发展计划项目《支持 ERP 开发的中间件与组件系统》的开发和研究。

致 谢

本论文的工作是在我的导师孙济洲教授的悉心指导下完成的，孙济洲教授严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢多年来孙济洲老师对我的关心和指导。

于策老师悉心指导我们完成了实验室的科研工作，在学习上和生活上都给予了我很大的关心和帮助，在此向于策老师表示衷心的感谢。

在实验室工作及撰写论文期间，肖健、魏继增、乌晓峰、何团伟、徐祯、孙敏、武华北、李文等同学对我论文的研究工作给予了热情帮助，在此向他们表达我的感激之情。