

武汉理工大学

硕士学位论文

基于Web的物流仓储管理系统的设计

姓名：俞奕

申请学位级别：硕士

专业：计算机应用技术

指导教师：夏红霞

20070501

摘 要

随着计算机和网络技术的不断发展, Internet 已经成为人们生产和生活中的不可缺少的组成部分。社会的各个领域都在努力利用现有技术建立网络化的应用体系, 进而实现信息交互和资源共享。就现代企业而言, 建立一个基于 Internet/Intranet 的灵活的、易于扩展和维护的企业信息系统, 是企业适应瞬息万变的市场环境, 促进自身发展壮大的必然选择。基于 Web 的物流仓储管理系统是通过构建一个面向公众的操作环境, 克服了传统物流管理系统功能单一、难以与企业其他业务系统紧密集成的缺点。本课题的研究目标就是提出一个在 J2EE 平台下, 运用 MVC 设计模式来实现基于 Web 的物流仓储管理信息系统的解决方案。

本文首先分析了物流仓储管理系统应该具备的作用, 在此基础上对系统功能进行详细分析, 并得出具体的功能模型。然后根据该功能模型设计出基于 J2EE 平台下的系统结构。

J2EE 技术提供了开发分布式应用的底层服务支持, 同时也提供了一个基于 JavaBean、Servlet 和 JSP 等组件开发 Web 应用的平台。以 J2EE 的多层体系结构为依据, 本文给出了一个四层的设计模型, 它包括了客户层、Web 层、业务层和数据层, 其中的业务层又被划分成业务逻辑层和数据持久化层两个子层。在这个多层设计模型中, Web 层负责控制业务流程、动态生成客户端页面等功能; 业务逻辑层实现了核心业务逻辑, 持久化层负责对数据库的具体操作。

本文最后还探讨了基于 Web 的 GIS 以及在物流领域的应用, 并对求解最短路径问题的几种经典算法及算法的优化进行了研究。

关键字: 物流仓储、J2EE、GIS、算法

Abstract

With the development of computer and Internet technology, Internet has become an indispensable part in people's live and production. Each area of our society are trying to establish application system based on network, by using this way to implement the communication of information and the share of resource. As modern enterprises, if they want to adapt the variational environment, establishing a flexible and extensible enterprise information system based on Internet/Intranet is a necessary choice. By building an open environment for the Physical Distribution and Storage Management System, which is based on web technology, overcomes restraints of simplex function and difficultness to integrate with other business system in traditional Physical Distribution and Storage System, establishes a dynamic controllable uniform integrative platform, supplies the advanced solutions of Electronic business for enterprise.

The article first analysed the actions of Physical Distribution and Storage Management System, based on this, it analysed the functions in detail and get the material models. Then we designed the system framework based on J2EE platform as the models.

J2EE provides the infrastructure of distributed application, which also is an advance platform of developing Web application based on J2EE components, such as JavaBean, Servlet and JSP. According J2EE multi-layer architecture, the model is composed of four layers, which includes client layer, Web layer, logic layer and data persistent layer. Logic layer provides the distributed service and includes business sublayer and persistent sublayer. Business sublayer implements business logic and persistent sublayer is responsible of operating with data source.

Finally, this paper discuss the application of GIS based on Web, and reseache some classical algorithms to solve the problem of shortest path, then optimize it.

Key words: Physical Distribution and Storage、J2EE、GIS、Algorithm

独 创 性 声 明

本人声明，所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得武汉理工大学或其它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。

研究生签名： 俞亮 日期： 2007.5.20

关于论文使用授权的说明

本人完全了解武汉理工大学有关保留、使用学位论文的规定，即学校有权保留、送交论文的复印件，允许论文被查阅和借阅；经作者同意学校可以公布论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存论文。

(保密的论文在解密后应遵守此规定)

研究生签名： 俞亮 导师签名： 夏明 日期： 2007.5.20

第 1 章 绪论

1.1 引言

随着经济体制改革的不断深入和市场的进一步开放，以及信息技术在物流仓储行业的广泛应用，我国的物流仓储行业已经进入了一个由传统物流向现代化物流过渡的变革时期。为了更有效的实现物流的各个基本功能，需要现代的各种物流技术提供支持。现代物流技术是指在现代物流活动中把商品进行移送和储存，为社会提供无形服务的技术。它可分为硬技术和软技术：硬技术又包含基础设施、机械技术、材料技术、信息通信技术等；软技术包括规划技术、运用技术、评价技术等。在各个子技术中又还可以分为更具体的仓库技术、装卸技术、包装材料技术等等。目前发展较快的技术有：运输、仓储、搬运、包装、集装单元化、物流信息技术等等。

1.2 物流仓储的现状与发展趋势

物流（Physical Distribution）一词源于国外，最早出现于美国，1915 年阿奇·萧在《市场流通中的若干问题》一书中就提到物流一词，并指出“物流是与创造需求不同的一个问题”。因为在本世纪初，西方一些国家已出现生产大量过剩、需求严重不足的经济危机，企业因此提出了销售和物流的问题，此时的物流指的是销售过程中的物流。二次世界大战中，围绕战争供应，美国军队建立了“后勤”（Logistics）理论，并将其用于战争活动中。其中所提出的“后勤”是指将战时物资生产、采购、运输、配给等活动作为一个整体进行统一布置，以求战略物资补给的费用更低、速度更快、服务更好。后来“后勤”一词在企业中广泛应用，又有商业后勤、流通过程的提法，这时的后勤包含了生产过程和流通过程的物流，因而是一个包含范围更广泛的物流概念。

因此，物流概念从 1915 年提出起（Physical Distribution），经过 70 多年的时间才有定论（Logistics），现在欧美国家把物流称作 Logistics 的多于称作 Physical Distribution 的。Logistics 包含生产领域的原材料采购、生产过程中的物料搬运与厂内物流和流通过程中的物流或销售物流即 Physical Distribution，可见其外延更为广泛。日本的物流概念是 1956 年直接从英文的 Physical Distribution 翻译过去的，1956 年日本派团考察美国的流通技术，引进了物流的

概念。到了 70 年代，日本已成为世界上物流最发达的国家之一。

1.2.1 运输技术的现状及趋势

运输是人和物的载运及输送。本文中专指“物”的载运及输送。它是在不同地域范围间(如两个城市、两个工厂之间，或一个大企业内相距较远的两车间之间)，以改变“物”的空间位置为目的的活动，对“物”进行空间位移。和搬运的区别在于，运输是较大范围的活动，而搬运是在同一地域之内的活动。目前国内外主要运用的运输技术有：智能运输系统（ITS）、重型载货汽车技术、全球卫星定位系统（GPS）与地理信息系统（GIS）等等。

1.智能运输系统

智能运输系统是将先进的信息技术、计算机技术、数据通信技术、传感器技术、电子控制技术、自动控制理论、运筹学、人工智能等有效地综合运用于交通运输、服务控制和车辆制造，加强了车辆、道路、使用者三者之间的联系，从而形成一种定时、准确、高效的综合运输系统。我国是在 20 世纪 90 年代开始大规模地对智能运输系统技术进行研究的，现已在全国许多地方相继开始推广应用，有的（如北京）已初步取得了成效。

2.重型载货汽车产品技术

近几年，世界发达国家各商用车制造商纷纷投巨资研制新一代产品，通过采用高新技术，最大限度地提升重型汽车产品技术水平。我国各重型汽车生产厂家为了迎接挑战，近几年亦集中力量发展新产品，提高主导产品技术水平，增强竞争力。目前我国重型载货汽车产品技术总的发展趋势是高档化、电子化、大吨位、环保化，重点在经济性、动力性、安全性、舒适性和可靠性等方面的技术水平日益得到提高。

3.地理信息系统（GIS）与全球卫星定位系统(GPS)技术

GIS（Geographical Information System，地理信息系统）以地理空间数据为基础，采用地理模型分析方法，适时地提供多种空间的和动态的地理信息，是一种为地理研究和地理决策服务的计算机技术系统。GPS（Geographical Position System，全球卫星定位系统）是一种先进的导航技术。地面监控中心利用 GPS 技术可以实时监控车辆等移动目标的位置，根据道路交通状况向移动目标发出实时调度指令。如国外的一些大型快递公司（如 UPS），已经应用 GPS 技术进行货物跟踪等应用。

1.2.2 仓储技术的现状及趋势

仓储作为连接生产者和消费者的纽带，是物流系统的一个中心环节。随着全球化信息网和全球化市场的逐步形成，仓库已经成为企业 B2B 的信息接点和生产零库存（JIT）的保障。深入研究仓储技术与设备，合理配置仓库资源、优化仓库布局和提高仓库的作业水平，从而提高供应链的竞争力具有十分重要的意义。目前，仓储货物的集装单元化技术、通用仓储机械、分拣系统、自动仓储控制系统、高架仓库和仓库选址和规划技术已经开始逐步推广。

1. 仓储物流标准化技术 仓储物流标准化是指物品在仓库中的包装尺寸和集装单元的尺寸要符合一定的标准。标准的建立可以大大提高仓储作业的效率，也可以提高整个物流系统的运作效率。标准化的重点就是：货物的集装单元与物流过程中的固定设施、移动设备、专用工具的统一性和配合性，其中包括标准托盘、标准集装箱、标准货架的应用等。

2. 自动化仓库的类型将向多品种发展：目前，我国设计、制造的自动化仓库几乎全部是分离式自动化仓库和托盘单元式自动化仓库。但大型、高层的自动化仓库，国外为降低成本，往往采用整体或自动仓库，如：1998 年日本整体式自动仓库年建 72 座（占总量的 9%），但其产值达 319.35 亿日元（占总产值的 51%）。此外，适用于家电、医药、电子等行业的箱盒单元大自动仓库必将有广阔的应用前景，如：1990 年日本建箱盒单元式自动仓库 316 座（占总量的 18%），年增长率达 157%。我国有必要予以开发、推广。

3. 出入库系统：我国日前大多采用链式和辊道输送机组合而成，应开发高速轨道式输送台车及其系统、滑块式分拣输送机、自动搬运车系统（AGVs）等高效、柔性的出入库输送设备及其系统。

4. 随着仓库设备及其系统的自动化、高速化，条码技术和自动识别技术必将成为信息录入和系统的瓶颈，其发展和应用值得重视。

5. 现代仓储系统是基于网络的由计算机控制的复杂系统，因此仓库管理系统软件、运输管理软件、车辆管理 GPS/GIS 软件、销售和分销软件、物料采购软件等和综合性的企业制造资源计划（MRP-II），企业资源计划（ERP）软件等都将有较大的需求和发展。

1.2.3 物流信息技术的现状

我国的物流服务企业中，绝大多数物流服务企业尚不具备运用现代信息技

术处理物流信息的能力。目前,我国物流领域中现代信息技术应用和普及的程度还不高,发展也不平衡。据调查,我国的物流服务企业,仅有 39%的企业拥有物流信息系统,绝大多数物流服务企业尚不具备运用现代信息技术处理物流信息的能力。

1、条形码技术

条形码作为商品标识方面的应用,目前国内约有 38.8%的零售企业并没有完全实现条码化。这不仅对零售企业利用 POS 系统实现销售效率提高有较大影响,而且影响了整个物流流程的信息采集与反馈、物流企业与工商企业之间的信息共享和相互合作、以及物流作业自动化的开展。

2、电子数据交换(EDI)

电子数据交换(EDI)作为一种新型有效商业信息管理手段,可以提高整个物流流程的信息管理水平和各个物流环节的协调发展。然而目前国内电子数据交换(EDI)的应用范围还非常有限,相对集中在进出口企业与海关、商检等管理部门之间的使用。

3、物流软件

信息技术在物流领域广泛应用的另一个主要标志,是针对物流活动的需要开发的、使用大量信息技术支持的管理软件。目前应用十分广泛的物流系统集成软件,有制造资源计划(MRP II)、企业资源计划(ERP)、供应商管理库存系统(VMI)、供应链管理(SCM)等。据调查显示,ERP、SCM 及 VMI 等集成系统软件在我国企业中实施的尚不足十分之一,其中制造企业应用情况略好于流通企业,如流通企业中实施 ERP 的比例目前则仅为 3%左右。

1.3 课题研究目标

武汉舵落口大市场是经国家经贸委批准,由武汉市人民政府组织实施的综合性、多功能的大型批发市场,已列入国务院批复的《武汉市城市总体规划》,是武汉市“十五”规划中物流基地的重点项目。然而与华中地区重点物流基地之地位不相称的是,该市场的经营、管理还停留在以单据、电话构件起来的传统平台上。在信息化、自动化、智能化上的落后已经成为制约市场发展的主要因素之一。在各级领导的关心下,该系统于 2005 年 10 月立项,并于次年 3 月组织实施。期望通过对舵落口大市场管理系统的研究与实现,进一步提升市场的经营、管理水平,强化市场功能,完成传统物流向现代物流的飞跃。

传统物业管理的手工操作已经不能满足当前大量信息管理的要求，而普通的 C/S 模式尽管能够提供数据综合管理，但是庞大的客户端使得系统修改和维护变得不方便。针对上述情况，迫切需要一个既能处理大量信息，又能根据不同的管理需求方便对其修改和维护的新型的管理系统。

舵落口大市场管理系统在分类上属于物流企业的管理信息系统，《中国企业管理百科全书》给管理信息系统（MIS）的定义是：“一个由人、计算机等组成的能进行信息的收集、传送、储存、加工、维护和使用的系统”，它不仅是一个技术系统，也是一个包括人在内的人机系统、社会系统。它是现代企业决策与运作管理的基础，也是这几年发展最快的企业信息系统之一。虽然现代物流正在向综合物流方向发展，但由于物流企业不同而主营业务不同，因此物流信息系统也有不同的类型，比如运输企业的信息系统、仓储企业的信息系统、物流中心信息系统、供应链物流信息系统等。不论哪一种类型的物流信息系统，其基本结构是相同的，只不过是物流信息技术、分信息系统在整个系统的地位有差异。比如，物流专家系统在以外包物流业务为主的第三方物流企业中就非常重要，而在其它物流企业的信息系统中地位就显得次要一些；又如，以 GPS、GIS 为基础的车辆运行管理系统在运输企业的信息系统中非常重要，而在其它物流企业中的地位则次之。

1.4 论文结构

本论文分为五章。

第一章为绪论，介绍了物流仓储管理系统所包含的内容及其特点，并对其发展状况做了介绍，最后给出了本文的研究目标和采用的技术路线。

第二章对本课题采用的 J2EE 分布式应用模型进行了深入的研究。

第三章介绍了地理信息系统技术和全球卫星定位系统技术，并就这些技术在物流仓储上的运用进行了分析。

第四章概要设计。论述了物流仓储管理系统的系统架构设计和功能模块的划分，并对业务逻辑进行了初步分析。

第五章物流管理辅助决策。提出了基于 GIS 的物流平台数据组织结构，并对求解最短路径问题的几种经典算法及算法的优化进行了研究。

第 2 章 J2EE 平台

2.1 J2EE 技术体系

2.1.1 J2EE 平台概述

随着 Internet 计算的迅速发展, 作为一门新兴的语言, Java 以其完全面向对象的设计风格和良好的平台无关性, 已经超越了语言本身的功能, 成为了开发 Web 应用的理想平台。Java 平台包括了一种用于实现 Java 对象的语言和这些 Java 对象的运行环境——Java 虚拟机 JVM (Java Virtual Machine)。Java 语言的主要特点是平台无关性: 在一个操作系统下开发的 Java 对象可以不做修改的运行在另外一个操作系统的 JVM 上。JVM 封装了各种底层操作系统在体系结构和语义上的差异, 在此基础上为 Java 对象提供了一个统一的运行环境。Java 对象的源代码被编译为平台无关的中性代码——字节码。在运行时刻, 字节码由 JVM 进行解释, 被映射为真正的相关的操作系统调用而得以执行功能。因此, 只要是 100% 纯 Java 语言实现的对象可以被各种具有相应 JVM 的操作系统所支持。

然而, 要开发基于 Internet 的分布应用, 仅依靠 Java 语言本身提供的功能是远远不够的, 大型分布系统特别是电子商务应用对企业的信息系统增加了复杂性。企业应用系统的开发一直面临着重大挑战: 一方面, 企业应用系统面对的是一个异构的分布式环境, 它必须支持与已有系统的集成性和与其他系统的互操作性; 另一方面, 作为给客户、合作伙伴和企业内部提供信息服务的平台, 企业系统还必须具有高可用性、安全性、可靠性和可伸缩性。这些要求再加上复杂多变的用户需求和不断伸缩的交付时间, 使得企业系统的开发越来越困难。开发商和广大程序员一直在努力推动和殷切期待一个成熟、标准的企业平台来简化和规范企业系统的开发和部署。Java 技术的出现, 尤其是 J2EE (Java 2 Platform Enterprise Edition) 平台的推出正是这种努力的结果, 也使得企业系统的开发由此变得更加快速和方便。

J2EE 是 Sun 公司在 1999 年 6 月推出的技术规范, 是目前能够满足当今企业开发要求的最佳解决方案。J2EE 是一个标准的体系结构, 它提供了基于组件的、以应用服务器为核心的多层应用体系结构, 以及系统必须的可靠性、可扩展

展性、可管理性和安全性。这种多层结构简化了开发、配置和维护企业应用的过程，它最大的优点就是将企业的业务逻辑同显示逻辑分开，应用服务器为组件提供了一系列的底层服务，如事务管理、多线程、分布式程序设计、缓冲池等，这就使得开发者能够将精力集中于特定领域的业务规则的组织上，而不用过多的关心那些底层的复杂而又烦琐的系统级事项。它是一组规范，旨在为支持 Java 语言服务器端部署而提供平台无关的、可移植的、多用户的、安全的和标准的企业级平台。

J2EE 平台由一整套服务 (services)、应用程序接口 (APIs) 和协议构成，它对开发基于 Web 的多层应用提供了功能支持，主要包括 JDBC、JNDI、EJB、RMI、Java IDL/ CORBA、JSP、Java Servlet、XML、JMS、JTA、JTS、JavaMail、JAF 等多种技术，其中 EJB 组件体系结构是 J2EE 平台的主干。一个 J2EE 应用的核心由一个或几个 EJB 组件组成，这些 EJB 组件执行这个应用的业务作业并封装业务逻辑。J2EE 平台的其他部分，例如 JSP，是对 EJB 体系结构的补充以提供呈现逻辑以及客户交互控制逻辑等功能。

总之，J2EE 平台包括建立企业应用系统的各个方面，使用 J2EE 技术能够快速建立可伸缩性企业应用系统，J2EE 的目标是提供一个基于 Java 语言的服务器端应用结构，支持平台独立、可移植、多用户、安全和标准的企业级应用。

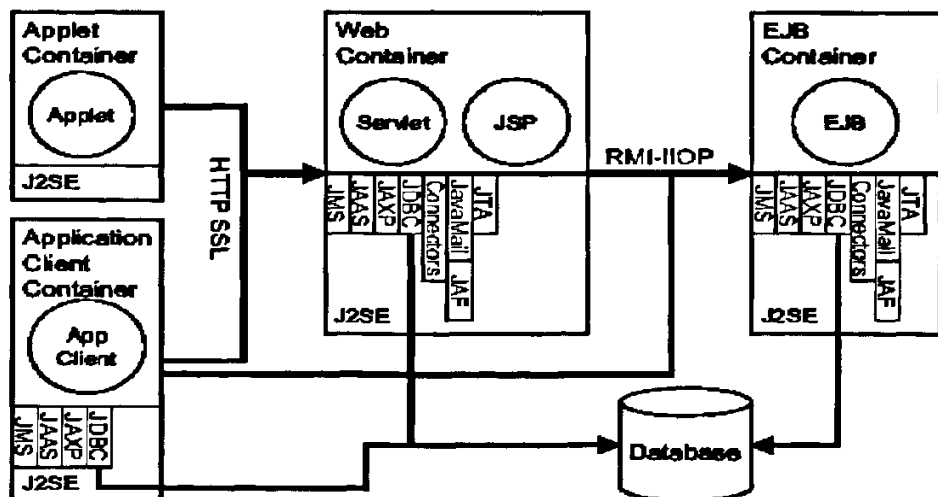


图 2-1 J2EE 平台结构图

2.1.2 J2EE 的各种组件

1. Servlet

Servlet 是 Java 平台上的 CGI 技术。Servlet 在服务器端运行，动态地生成 Web 页面。与传统的 CGI 和许多其它类似 CGI 的技术相比，Java Servlet 具有更高的效率并更容易使用。对于 Servlet，重复的请求不会导致同一程序的多次转载，它是依靠线程的方式来支持并发访问的。

2. JSP

JSP(Java Server Page)是一种实现普通静态 HTML 和动态页面输出混合编码的技术。从这一点来看，非常类似 Microsoft ASP、PHP 等技术。借助形式上的内容和外观表现的分离，Web 页面制作的任务可以比较方便地划分给页面设计人员和程序员，并方便地通过 JSP 来合成。在运行时态，JSP 将会被首先转换成 Servlet，并以 Servlet 的形态编译运行，因此它的效率和功能与 Servlet 相比没有差别，一样具有很高的效率。

3. EJB

EJB 定义了一组可重用的组件：Enterprise Beans。开发人员可以利用这些组件，像搭积木一样建立分布式应用。在装配组件时，所有的 Enterprise Beans 都需要配置到 EJB 服务器(一般的 Weblogic、WebSphere 等 J2EE 应用服务器都是 EJB 服务器)中。EJB 服务器作为容器和低层平台的桥梁管理着 EJB 容器，并向该容器提供访问系统服务的能力。所有的 EJB 实例都运行在 EJB 容器中。EJB 容器提供了系统级的服务，控制了 EJB 的生命周期。EJB 容器为它的开发人员代管了诸如安全性、远程连接、生命周期管理及事务管理等技术环节，简化了商业逻辑的开发。EJB 中定义了三种 Enterprise Beans：(1) Session Beans；(2) Entity Beans；(3) Message-driven Beans。

4. JDBC

JDBC(Java Database Connectivity, Java 数据库连接)API 是一个标准 SQL(Structured Query Language, 结构化查询语言)数据库访问接口，它使数据库开发人员能够用标准 Java API 编写数据库应用程序。JDBC API 主要用来连接数据库和直接调用 SQL 命令执行各种 SQL 语句。利用 JDBC API 可以执行一般的 SQL 语句、动态 SQL 语句及带 IN 和 OUT 参数的存储过程。Java 中的 JDBC 相当与 Microsoft 平台中的 ODBC(Open Database Connectivity)。

5. JMS

JMS(Java Message Service, Java 消息服务)是一组 Java 应用接口,它提供创建、发送、接收、读取消息的服务。JMS API 定义了一组公共的应用程序接口和相应语法,使得 Java 应用能够和各种消息中间件进行通信,这些消息中间件包括 IBM MQ-Series、Microsoft MSMQ 及纯 Java 的 SonicMQ。通过使用 JMS API,开发人员无需掌握不同消息产品的使用方法,也可以使用统一的 JMS API 来操纵各种消息中间件。通过使用 JMS,能够最大限度地提升消息应用的可移植性。JMS 既支持点对点的消息通信,也支持发布/订阅式的消息通信。

6. JNDI

由于 J2EE 应用程序组件一般分布在不同的机器上,所以需要一种机制以便于组件客户使用者查找和引用组件及资源。在 J2EE 体系中,使用 JNDI(Java Naming and Directory Interface)定位各种对象,这些对象包括 EJB、数据库驱动、JDBC 数据源及消息连接等。JNDI API 为应用程序提供了一个统一的接口来完成标准的目录操作,如通过对象属性来查找和定位该对象。由于 JNDI 是独立于目录协议的,应用还可以使用 JNDI 访问各种特定的目录服务,如 LDAP、NDS 和 DNS 等。

7. JTA

JTA(Java Transaction API)提供了 J2EE 中处理事务的标准接口,它支持事务的开始、回滚和提交。同时在一般的 J2EE 平台上,总提供一个 JTS(Java Transaction Service)作为标准的事务处理服务,开发人员可以使用 JTA 来使用 JTS。

8. JCA

JCA(J2EE Connector Architecture)是 J2EE 体系架构的一部分,为开发人员提供了一套连接各种企业信息系统(EIS,包括 ERP、SCM、CRM 等)的体系架构,对于 EIS 开发商而言,它们只需要开发一套基于 JCA 的 EIS 连接适配器,开发人员就能够在任何的 J2EE 应用服务器中连接并使用它。基于 JCA 的连接适配器的实现,需要涉及 J2EE 中的事务管理、安全管理及连接管理等服务组件。

9. JMX

JMX(Java Management Extensions)的前身是 JMAPI。JMX 致力于解决分布式系统管理的问题。JMX 是一种应用编程接口、可扩展对象和方法的集合体,可以跨越各种异构操作系统平台、系统体系结构和网络传输协议,开发无缝集

成的面向系统、网络和服务的管理应用。JMX 是一个完整的网络管理应用程序开发环境，它同时提供了厂商需要收集的完整的特性清单、可生成资源清单表格、图形化的用户接口；访问 SNMP 的网络 API；主机间远程过程调用；数据库访问方法等。

10. JAAS

JAAS (Java Authentication and Authorization Service) 实现了一个 Java 版本的标准 Pluggable Authentication Module (PAM) 的框架。JAAS 可用来进行用户身份的鉴定，从而能够可靠并安全地确定谁在执行 Java 代码。同时 JAAS 还能通过对用户进行授权，实现基于用户的访问控制。

11. JACC

JACC (Java Authorization Service Provider Contract for Containers) 在 J2EE 应用服务器和特定的授权认证服务器之间定义了一个连接的协议，以便将各种授权认证服务器插入到 J2EE 产品中去。

12. JAX-RPC

通过使用 JAX-RPC (Java API for XML-based RPC)，已有的 Java 类或 Java 应用都能够被重新包装，并以 Web Services 的形式发布。JAX-RPC 提供了将 RPC 参数 (in/out) 编码和解码的 API，使开发人员可以方便地使用 SOAP 消息来完成 RPC 调用。同样，对于那些使用 EJB (Enterprise JavaBeans) 的商业应用而言，同样可以使用 JAX-RPC 来包装成 Web 服务，而这个 Web Service 的 WSDL 界面是与原先的 EJB 的方法是对应一致的。JAX-RPC 为用户包装了 Web 服务的部署和实现，对 Web 服务的开发人员而言，SOAP/WSDL 变得透明，这有利于加速 Web 服务的开发周期。

13. JAXR

JAXR (Java API for XML Registries) 提供了与多种类型注册服务进行交互的 API。JAXR 运行客户端访问与 JAXR 规范相兼容的 Web Services，这里的 Web Services 即为注册服务。一般来说，注册服务总是以 Web Services 的形式运行的。JAXR 支持三种注册服务类型：JAXR Pluggable Provider、Registry-specific JAXR Provider、JAXR Bridge Provider (支持 UDDI Registry 和 ebXML Registry/Repository 等)。

14. SAAJ

SAAJ (SOAP with Attachments API for Java) 是 JAX-RPC 的一个增强，为

进行低层次的 SOAP 消息操纵提供了支持。

15. Javamail/JAF

Javamail API 是为 J2EE 组件提供 e-mail 服务的。它在具体的邮件服务实现（例如 SMTP 和 POP3）之上，统一对各种可能用到的公共服务的访问形式，例如邮件格式、发送/接收邮件、附件处理等。JavaMail 要求 Java 激活框架（JAF, Javabeen Activation Framework）来处理复杂的数据类型，例如 MIME。

2.1.3 J2EE 分布式体系结构

J2EE 使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据它们所在的层分布在不同的机器上。也就是说 J2EE 平台提供了基于组件的、以应用服务器为核心的多层应用体系结构，以及系统必需的可靠性、可扩展性、可管理性和安全性。这种多层结构简化了开发、配置和维护企业应用的过程，它最大的优点就在于将企业的业务逻辑同呈现逻辑分开，应用服务器为组件提供了一系列的底层服务，如事务管理、多线程、分布式程序设计、缓冲池等，这就使得开发者能够将精力集中于特定领域的业务规则的组织上，而不用过多地关心那些底层的复杂而又烦琐的系统级事项。

整个应用系统由客户层 (Client)、中间层，也叫业务逻辑层 (Business Logic) 和企业信息系统层 (Enterprise Information Systems) 构成：

一个 J2EE 典型的结构如图 2-2 所示。

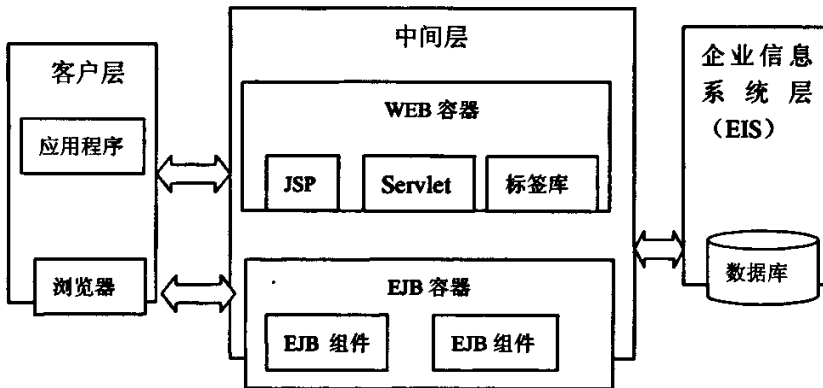


图 2-2 J2EE 层次结构图

1. 客户层：应用客户端部分，包含浏览器和普通的 GUI 应用程序。浏览器使用 HTTP 与 WEB 容器进行通信。一个独立的应用程序也可以使用 HTTP，或者

使用远程方法调用(RMI)直接与 EJB 容器进行通信。它可以是基于 Web 应用的,也可以不是基于 Web 的。在一个基于 Web 的 J2EE 中,浏览器负责显示由运行在 Web 层的 Web 组件生成的包含各种标记语言(HTML、XML、XSL 等)的 Web 页面,它可以通过运行在 Web 层中的 JSP 页面和 servlet 与 J2EE 服务器中的业务层进行通信。一个不基于 Web 的 J2EE 应用系统中,一个独立的客户端程序是运行在其它一些基于网络系统中的 Java 程序,可以通过 RMI-IIOP 直接访问运行在 EJB 容器中的 EJB 组件。不基于 Web 的客户端程序可能也包括一个 JavaBeans 类来管理用户输入,并将输入发送到业务逻辑层中运行的 EJB 对象来处理。

2. 中间层:包括 WEB 容器和 EJB 容器,并通过它们为客户端提供服务。J2EE 的 Web 容器组件既可以是 servlet 也可以是 JSP 页面。servlet 是一种服务器方程序,允许应用程序逻辑嵌入到 HTTP 请求响应过程中。JSP 提供了一种在网页中嵌入组件的方式,它是一个基于文本的文档,由 Web 容器编译成相应的 servlet,这样在应用程序逻辑和表示逻辑之间有了清楚的分界。Web 层也可以包括一个 JavaBeans 类来管理用户输入,并将输入发送到业务逻辑层中运行的 EJB 对象来处理。EJB 容器构成了应用的业务逻辑规则,是整个应用的核心部分。它按业务划分成一个个独立的逻辑单元 EJB。EJB 从客户程序处接收数据,对数据进行处理,再将数据发送到企业信息系统层存储,还可以从存储中检索数据,并将数据送回客户程序。业务逻辑层位于应用服务器中,应用服务器中的 EJB 容器提供了分布式计算中组件所需要的所有服务,例如:组件生命周期的管理、数据库连接的管理、分布式事务管理、组件的命名服务、自动容错以及负载的自动均衡能力等。实现业务逻辑的 EJB 组件可以更加高效地运行在应用程序服务器中。

3. 企业信息系统层(EIS):它运行企业信息系统软件,掌管着应用程序的包括企业基础设施系统。通常由一个或多个关系数据库管理服务器组成。同时像企业资源计划(Enterprise Resource Planning, ERP)这样的应用程序以及一些大型机事务处理系统都可以被包括到这一层中。中间层使用 J2EE API(例如 JDBC 和 JTA/JTS)与 EIS 层发生相互作用。

2.2 J2EE 架构企业级应用

企业级应用是指那些为商业组织、大型企业而创建并部署的解决方案及应用。这些大型企业级应用的结构复杂,涉及的外部资源众多、事务密集、数据

量大、用户数多，有较强的安全性考虑。当代的企业级应用决不可能是一个个的独立系统。在企业中，一般都会部署多个彼此连接的、相互通过不同集成层次进行交互的企业级应用，同时这些应用又都有可能与其它企业的相关应用连接，从而构成一个结构复杂的、跨越 Intranet 和 Internet 的分布式企业应用群集。其中，连接企业内部各种应用的技术称为 EAI (Enterprise Application Integration, 企业应用集成)，而连接企业间各种应用的技术称为 B2BI (Business-To-Business Integration, 企业间集成)，采用 EAI、B2BI 技术及早先的 N 层体系架构就是当今企业级应用的最大特征。

此外，作为企业级应用，其不但要有强大的功能，还要能够满足未来业务需求的变化，易于升级和维护。下面我们使用一个图示来简单地解析一下企业级应用的架构，见图 2-3。

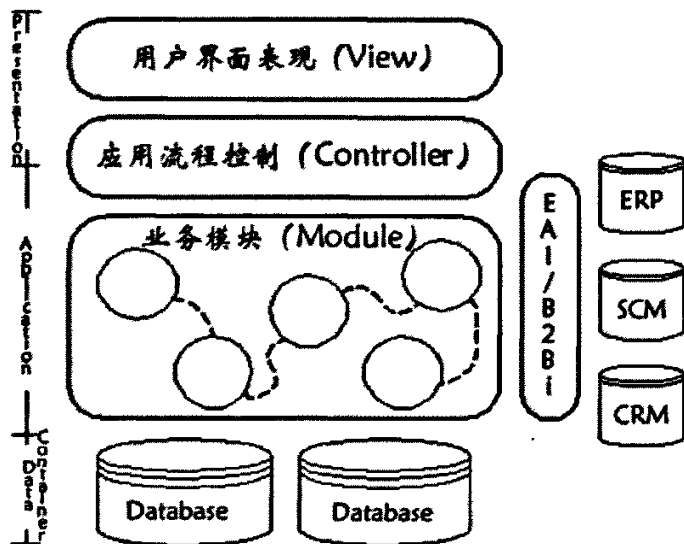


图 2-3 企业级应用的体系架构

这个体系架构的主体是 MVC 架构。MVC 是 Model/View/Control 的缩写。Model/View/Control 是软件设计的典型结构。在这种设计结构下，一个应用被分为三个部分：Model、View 和 Controller，每个部分负责不同的功能。Model 是指对业务数据/信息的处理模块，包括对业务数据的存取、加工、综合等；View 是指用户界面，也就是面向用户的数据表示；Controller 则负责 View 和 Model 之间的流程控制，也就是完成两个方向的动作：1. 将用户界面(View)的操作映射到具体的 Model，以完成具体的业务逻辑；2. 将通过 Model 处理完的业务数

据及时反应到用户界面(View)上。

MVC 架构使得应用程序的结构更加清晰,通过将代码按照层次划分为业务逻辑/数据、用户界面和应用流程控制这三个层次,增强代码稳定性。我们知道,对于 Model、View、Controller 这三部分功能来讲,View 的实现一般是由界面设计人员和界面程序员来完成,Model 则是由业务逻辑程序员来完成,Controller 则一般由负责整体控制的程序员来完成。Controller 部分的代码比较稳定,一般会实现一个通用的架构;而 Model 则跟随商务流程的变化而变化;View 的更改则是随着用户需求的更改而更改。这种模块功能的划分有利于在代码修改过程中进行模块的隔离,而不需要把具有不同功能的代码混杂在一起造成混乱。对于项目开发而言,有利于在项目小组内按照小组成员各自的擅长进行分工,有利于三个部分并行开发、加快项目进度。

对于 Model 部分,也就是业务逻辑的处理部分,一般总是对商务数据进行处理、加工、综合等。对于企业级应用而言,商务数据有两种来源,一种是数据库,另一种则是其它的企业级应用系统。

对于数据库而言,这应当是大家熟悉的领域。我们通过使用数据库驱动程序,利用 SQL 来查询、操纵数据库。而对于其它的企业级应用而言,一般这些应用都会提供 API,通过这些 API,其它的应用就能够存取访问其中的数据,甚至是触发这些企业级应用中的一些业务流程。

企业级应用需要使用优秀的企业级应用体系结构,而优秀的企业级应用体系结构通常来自于优秀的解决方案。应用程序设计开始就要考虑其体系结构的合理性、灵活性、健壮性,从而既可满足企业级应用的复杂需求,也能为今后系统的调整和升级留有余地。体系结构影响了整个应用的生命周期,实际上能够延长整个应用的生命周期,同时增强了用户在多变的商业社会中的适应性,减少了系统维护的开销和难度,从而给用户带来最大的利益。

第3章 GIS 与 GPS

目前的中国物流业正处于蓬勃发展的时期,物流业已经成为我国第三产业新的“经济增长点”,越来越受到人们的关注。信息化物流被业内人士称为“企业管理的又一次革命”,因为其在集中采购、集中库存、运输优化等方面的作用,较其他形式的物流业占据有绝对优势,物流信息化就是降低成本和节约时间。但目前我国物流企业的信息化建设还处于初级阶段,业务流程和操作的优化尚处于起步和摸索阶段。因为很多物流企业是从原来的物资企业改制而来,他们面临的问题是尽快建立起企业内部的信息化管理体系。另外,由于我国制造业和商业分销领域的集中化程度不高,与其相适应的物流企业也以中小型企业为主体,所以,无论企业物流还是物流企业的信息化总体水平都不高。而在国外,特别是一些发达国家的物流公司,他们早就完成了物流信息化,洋物流杀进来后,众多中国物流企业将面临生死存亡严峻问题。

随着企业电子商务的崛起,分销渠道的进一步整合和供应链管理的出现,要求物流企业能够向客户提供全面的配送解决方案。但信息技术应用的落后,使得上下游企业之间物流活动难以协调,让物流活动变成模糊的黑洞,成本高且可控制性差,严重制约了我国物流企业的发展。据中国仓储协会的调查报告显示,我国车辆运营的空载率约 45%左右。造成这一情况的重要原因之一就是物流企业无法准确知道运行车辆的具体位置,而且无法与司机随时随地的保持联系,不能为其组织货源和灵活配货。同时,司机只能凭个人经验确定路线,有时不能找到最佳路径,不仅延误时机而且增加运行成本。另外,实际客户也不能及时了解货物配送过程的情况,不能和物流企业协调配合。随着互联网的发展和通讯技术进步,跨平台、组件化的 GIS(地理信息系统)和 GPS(全球定位系统)技术的逐步成熟,基于 GIS/GPS 的应用将构造具有竞争力的透明物流企业。

3.1 地理信息系统(GIS)

GIS (Geographical Information System, 地理信息系统)是多种学科交叉的产物,它以地理空间数据为基础,采用地理模型分析方法,适时地提供多种空间的和动态的地理信息,是一种为地理研究和地理决策服务的计算机技术

系统。其基本功能是将表格型数据（无论它来自数据库、电子表格文件或直接在程序中输入）转换为地理图形显示，然后对显示结果浏览、操作和分析。其显示范围可以从洲际地图到非常详细的街区地图，显示对象包括人口、销售情况、运输线路以及其他内容。

目前 GIS 重要的研究成果主要表现在：OpenGIS 的研究保证了用户可以存取在网络上的异构 GIS 数据和处理单元；关系数据库 (RDBMS) 和 GIS 的有效结合，使得许多 RDBMS 也将支持新的对象关系模型，从而可以更好地支持空间数据类型；GIS 构件 (Component) 的开发使得原来的大型 GIS 系统正迅速走向构件化，分解为基本的 GIS 构件；WEB (Internet / Intranet) 已经成为 GIS 的新的操作平台；数据挖掘 (Data Mining) 技术的发展，为知识发现提供了新的工具。GIS 的最新研究成果为 GIS 技术引入物流管理提供了基础的技术条件。

3.1.1 GIS 概述

地理信息系统主要由四部分组成：即计算机硬件系统，计算机软件系统，空间数据及系统的组织和使用维护人员即用户。其核心内容是计算机硬件和软件，空间数据反应了应用地理信息系统的信息内容，用户决定了系统的工作方式。如图 3-1 所示。

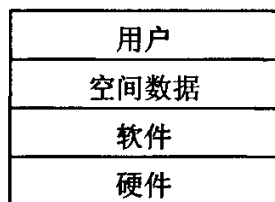


图 3-1 地理信息系统的组成

一、计算机硬件系统

计算机硬件系统是计算机系统中实际物理设备的总称，主要包括计算机主机，输入设备，存储设备和输出设备。

二、计算机软件系统

计算机软件系统是地理信息系统运行时所必需的各种程序。包括：

1. 计算机系统软件。这些软件通常由计算机生产厂家提供。

2. 地理信息系统软件及其支撑软件。包括地理信息系统工具或地理信息系统实用软件程序，以完成空间数据的输入、存储、转换、输出及其用户接口功能等。

3. 应用程序。这是根据专题分析模型编制的特定应用任务的程序，是地理信息系统功能的扩充和延伸。一个优秀的地理信息系统工具，对应用程序的开发应是透明的。应用程序作用于专题数据上，构成专题地理信息系统的基本内容。

三、空间数据

空间数据是地理信息系统的重要组成部分，是系统分析加工的对象，是地理信息系统表达现实世界的经过抽象的实质性内容。它一般包括三个方面的内容：即空间位置坐标数据，地理实体之间空间拓扑关系以及相应于空间位置的属性数据。通常，它们以一定的逻辑结构存放在空间数据库中，空间数据来源比较复杂，随着研究对象不同，范围不同，类型不同，可采用不同的空间数据结构和编码方法，其目的就是为了更好地管理和分析空间数据。

四、系统使用管理和维护人员

地理信息系统是一个复杂的系统，仅有计算机硬件、软件及数据还不能构成一个完整的系统，必须要有系统的使用管理人员。其中包括具有地理信息系统知识和专业知识的高级应用人才；具有计算机知识和专业知识的软件应用人才以及具有较强实际操作能力的硬软件维护人才。

地理信息系统软件一般由五部分组成，即空间数据输入管理、空间数据库管理、空间数据处理分析、空间数据输出管理及应用模型组成。它们之间关系见图 3-2：

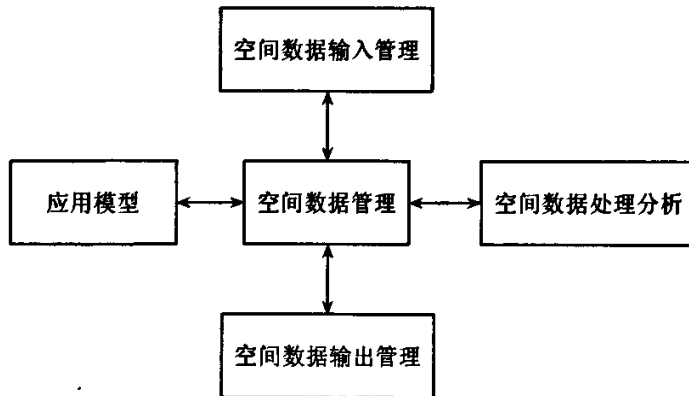


图 3-2 地理信息系统的主要模块

一、空间数据输入管理

空间数据输入管理模块是相对独立的功能模块。它的目的是将地理信息系

统中各种数据源输入，并转换成计算机所要求的数字格式进行存储。随着数据源种类的不同(如文本数据，数字数据和模拟数据等)，输入设备的不同及系统选用数据结构及数据编码的不同，在数据输入部分配有不同的软件，以确保原始数据按要求存入空间数据库中。通常，空间数据输入的同时，伴随着对输入数据处理，以实现数据的校验和编辑。

二、空间数据库管理

同一般数据库相比，地理信息系统数据库不仅要管理属性数据，还要管理大量图形数据，以描述空间位置分布，以及拓扑关系。而且，属性数据和图形数据之间具有不可分割的联系。此外，地理信息系统中数据库的数据量大，涉及内容多，这些特点决定了它既要遵循常用关系型数据库管理系统来管理数据，又要采用一些特殊的技术和方法，来解决通常数据库没法管理的空间数据问题。由于地理信息系统数据库具有明显的空间性，所以亦称为空间数据库。如图 3-3 所示。

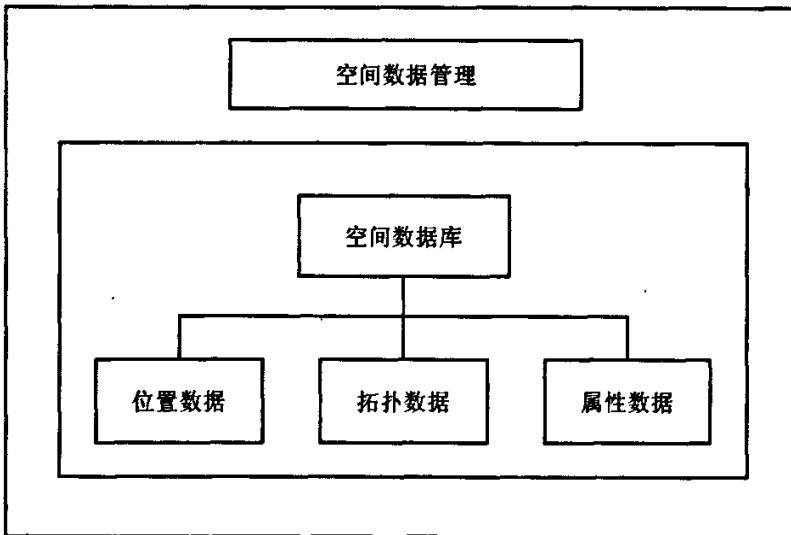


图 3-3 空间数据管理

三、空间数据处理和分析

空间数据处理和分析模块通常为地理信息系统提供一些基本和常用的处理和分析功能，其功能的强弱直接影响到地理信息系统应用范围。因此，这部分是体现地理信息系统功能强弱的关键部分。

四、应用模型

由于地理信息系统应用范围越来越广, 常规系统提供的处理和分析功能很难满足所有用户的要求。因此一个优秀的地理信息系统应当为用户提供二次开发手段, 以使用户开发新的空间分析模块, 即开发各种应用模型, 扩充地理信息系统功能。

五、空间输出数据管理

地理信息系统中输出数据种类很多, 可能是输出地图、表格、文字、图象等; 输出介质可以是纸、光盘、磁盘、显示终端等。随着输出数据类型不同和输出介质的不同需配备不同软件, 最终向用户报告分析结果。

3.1.2 GIS 发展简史

GIS 是为解决资源与环境等全球性问题而发展起来的技术与产业。60 年代中期, 加拿大开始研究建立世界上第一个地理信息系统(CGIS), 随后又出现了美国哈佛大学的 SYMAP 和 GRID 等系统。自那时起, GIS 就开始服务于经济建设和社会生活。在北美、西欧和日本等发达国家, 现在已建立了国家级、洲际之间以及各种专题性的地理信息系统。我国 GIS 的研究与应用始于 80 年代, 近二十年来发展也十分迅速, 在计算机辅助绘制地图等方面开展了大量基础性的试验与研究, 在理论、技术方法和实践经验等方面都有了长足的进步。当今, GIS 系统已成为年增长率为 35% 的新兴技术产业, 出现了诸如 ARC/INFO、MAPINFO 和 GENAMAP 等著名软件, 它们在城市建设、环境保护和社会发展等方面发挥了巨大的作用, 随着科技的发展, GIS 发展的势头越来越迅猛。

地理信息系统的发展大致可以分为三个阶段:

1. 模拟地理信息系统阶段

自 19 世纪以来就得到广泛应用的地图(包括地形图和专题图)——模拟的图形数据库和描述地理的文献著作——模拟的属性数据库相结合, 构成了地理信息系统的基本概念模型。但是, 这种模拟式的、基于纸张的信息系统和信息过程, 使得空间相关数据的存贮、管理、量算与分析、应用极为不规范、不方便和效率低下。随着计算机科学的兴起, 数字地理信息的管理与使用成为必然。

2. 学术探索阶段

50 年代, 由于电子技术的发展及其在测量与制图学中的应用, 人们开始有可能用电子计算机来收集、存贮和处理各种与空间和地理分布有关的图形和属性数据。1956 年, 奥地利测绘部门首先利用电子计算机建立了地籍数据库, 随

后这一技术被各国广泛应用于土地测绘与地籍管理。1963年,加拿大测量学家首先提出地理信息系统这一术语,并建立了世界上第一个地理信息系统——加拿大地理信息系统(CGIS),用于资源与环境的管理和规划。稍后,北美和西欧成立了许多与GIS有关的组织与机构,如美国城市与区域信息系统协会(URISA),国际地理联合会(IGU)地理数据收集和处理委员会(CGDPS),等等,极大地促进了地理信息系统知识与技术的传播和推广应用。

3. 飞速发展和推广应用阶段

70年代以后,由于计算机技术的工业化、标准化与实用化,以及大型商用数据库系统的建立与使用,地理信息系统对地理空间数据的处理速度与能力取得突破性进展。其结果是:a)一些发达国家先后建立了许多专业性的土地信息系统(LIS)和资源与环境信息系统(GIS),如1970年至1976年,美国地质调查局先后建50多个信息系统,其它国家也相继发展自己的GIS;b)关于GIS软件、硬件和项目开发的商业公司蓬勃发展。到1989年,国际市场上有报价的GIS软件达70多个,并出现一些有代表性的公司和产品,如美国环境系统研究所的ARC/INFO,Intergraph公司的MGE,及Genasys公司的Genamap等等。c)数字地理信息的生产标准化、工业化和商品化。如美国地质调查局(USGS)已经建立起数字地图生产技术体系,可定期向社会发布DLG、DEM等标准地理数据。d)各种通用和专用的地理空间分析模型得到深入研究和广泛使用,GIS的空间分析能力显著增强。e)有关GIS的具有技术权威和行政权威的行业机构和研究部门在GIS的应用发展中发挥引导和驱动作用。如美国国家科学基金会支持的国家地理信息和分析中心(NCGIA)每年都要制定GIS技术优先发展计划,以促进全国GIS事业发展。

3.1.3 基于Web的GIS

随着信息技术和网络技术的发展,WebGIS已经成为了GIS技术的一个重要的发展方向。与传统的GIS技术相比,它具有访问范围广、平台独立、能大规模降低系统成本以及维护、升级方便等特点。这使现代物流管理系统中充分利用GIS技术成为可能。但是,由于WebGIS技术尚处在发展之中,在建设基于它的物流信息系统时存在许多亟待研究解决的问题。例如,如何提高网络传输效率,解决海量数据在Internet网上传输的瓶颈问题;WebGIS服务对象的信息反馈问题;空间信息的加密传输与安全保护问题;与传统GIS技术的集成等等。本文首先

介绍了 WebGIS 技术的发展现状、国内外主流 WebGIS 产品的特点以及 WebGIS 技术在物流领域内的应用情况,通过对商用 WebGIS 平台 ArcIMS 的体系结构、工作原理及二次开发技术的研究,总结出结合商用 WebGIS 平台开发应用型 WebGIS 产品的模式与方法,即以商用 WebGIS 平台快速构筑应用系统的核心 GIS 功能;针对商用 WebGIS 不具备而用户又必需的功能,设计实现满足用户需求的与商用 WebGIS 具有平行结构的扩展功能服务器;根据商用 WebGIS 平台的工作原理及其提供的二次开发接口,将扩展功能和 GIS 功能有机地结合在一起,高效、可靠地满足项目的需求。课题设计了一个物流 WebGIS 模型,并根据物流业务的特点,设计实现了客户定位模型、运输工具定位模型以及最短路径规划模型等三个常用的业务应用模型,分别解决了客户空间位置的确定、运输工具空间位置和运行轨迹的确定、城市间和城市内交通运输的最短路径规划功能。利用手机短消息技术实现了便捷的低成本的双向通讯模型,与常用的 GPS+GSM 模块的单项通讯模式相比较,所能提供的服务更加多样。从实际应用出发,设计了一个“转换”工作模式以及有关数据协议,解决了因电信运营商不同而造成的数据格式不兼容的问题。最后,为了解决制约 WebGIS 技术应用的瓶颈—GIS 海量数据的网络传输瓶颈问题,课题基于代理服务器的思想,设计了一对网络数据传输组件,对以 XML 格式表达的 GIS 海量空间分析数据进行压缩传输。试验表明,在大多数条件下取得了令人满意的效果,尤其在窄网络带宽的环境下,效果尤其明显。

3.2 全球卫星定位系统(GPS)

GPS 是美国从本世纪 70 年代开始研制,历时 20 年,耗资 200 亿美元,于 1994 年全面建成,具有在海、陆、空进行全方位实时三维导航与定位能力的新一代卫星导航与定位系统。经近 10 年我国测绘等部门的使用表明, GPS 以全天候、高精度、自动化、高效益等显著特点,赢得广大测绘工作者的信赖,并成功地应用于大地测量、工程测量、航空摄影测量、运载工具导航和管制、地壳运动监测、工程变形监测、资源勘察、动力学等多种学科,从而给测绘领域带来一场深刻的技术革命。

随着全球定位系统的不断改进,硬、软件的不断完善,应用领域正在不断地开拓,目前已遍及国民经济各种部门,并开始逐步深入人们的日常生活。

3.2.1 GPS 概述

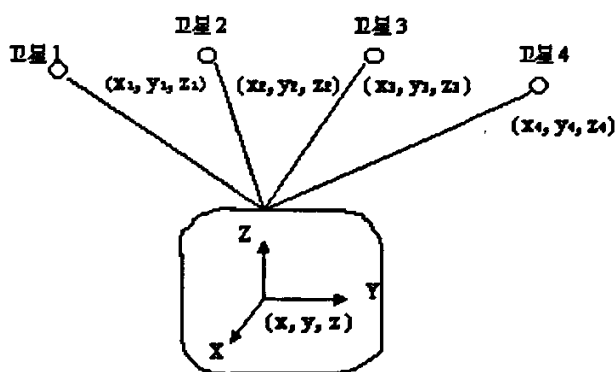
全球定位系统 (NAVSTAR GPS, NAVigation Satellite Timing And Ranging Global Positioning System, 以下简称 GPS) 是一个中距离圆型轨道卫星定位系统。它可以为地球表面绝大部分地区 (98%) 提供准确的定位、测速和高精度的时间标准。该系统的组成包括太空中的 24 颗 GPS 卫星; 地面上的 1 个主控站、3 个数据注入站和 5 个监测站及作为用户端的 GPS 接收机。最少只需其中 4 颗卫星, 就能迅速确定用户端在地球上所处的位置及海拔高度; 所能收联接到的卫星数越多, 解码出来的位置就越精确。

该系统是由美国政府于 20 世纪 70 年代开始进行研制于 1994 年全面建成, 原是美国国防部为了军事定时、定位与导航的目的所发展, 希望以卫星导航为基础的技术可构成主要的无线电导航系统, 未来并能满足下一个世纪的应用。第一颗 GPS 卫星在 1978 年发射, 首十颗卫星称为 BLOCK I 试验型卫星, 从 1989 年到 1993 年所发射的卫星称为 BLOCK II/IIA 量产型卫星, 第二十四颗 BLOCK II/IIA 卫星在 1994 年发射后, GPS 已达到初步操作能力 (Initial Operational Capability, IOC), 24 颗 GPS 卫星提供全世界 24 小时全天候的定位与导航信息。美国空军太空司令部于 1995 年 4 月 27 号宣布 GPS 已达到完整操作能力 (Full Operational Capability), 将 BLOCK I 卫星加以汰换而 24 颗卫星全部为 BLOCK II/IIA 卫星, 之后又发射四颗 BLOCK IIA 及一颗 BLOCK IIR 卫星, 成功地满足军事实务的操作。

由于此技术的迅速发展, 使得民间应用的需求与日遽增, 对于传统导航方式更有革命性的影响。全球卫星定位系统实际上是由 24 颗卫星所组成, 其中有 3 颗为备用卫星, 这些卫星分布于距地表 20,200 公里的上空, 而且分属于 6 个轨道面; 卫星轨道面倾斜角为 55 度, 提供全球全天候, 每秒一次, 持续不断的定位讯号。这些卫星每 11 小时 58 分环绕地球一次, 即每天绕过您的头顶二次, 就像是月球一样不停地绕着地球旋转, 其速度约每秒 1.8 哩。这些卫星需要地面管制站随时加以监控是否 GPS 卫星在其正确的轨道上及正常运作, 另外监控中心可上传资料给卫星, 卫星再将这些信息下传给 GPS 使用者使用。地面共有五个监控中心, 四个上传资料站及一个控制中心, 这些控制站以纬度来划分其所控制的卫星。在概念上, GPS 是代表着整个系统, 包括天空上的卫星、地面控制站及 GPS 接收机。不过一般而言, GPS 即意指一个 GPS 接收机, 这是因为

我们在使用上多半只会接触 GPS 接收机的缘故。一般所使用之 GPS 接收机在任何时刻接收到 4 颗或更多的卫星。使用者只需拥有 GPS 接收机，无需另外付费。GPS 信号分为民用的标准定位服务 (sps, standard positioning service) 和军规的精密定位服务 (pss, precise positioning service) 两类。民用讯号中加有误差，其最终定位精确度大概在 100 米左右；军规的精度在十米以下。2000 年以后，克林顿政府决定取消对民用信号所加的误差。因此，现在民用 GPS 也可以达到十米左右的定位精度。

GPS 定位的基本原理是根据高速运动的卫星瞬间位置作为已知的起算数据，采用空间距离后方交会的方法，确定待测点的位置。如图所示，假设 t 时刻在地面待测点上安置 GPS 接收机，可以测定 GPS 信号到达接收机的时间 Δt ，再加上接收机所接收到的卫星星历等其它数据可以确定以下四个方程式：



$$[(x_1-x)^2+(y_1-y)^2+(z_1-z)^2]^{1/2}+c(\tau_{t_r}-\tau_{t_o})=d_1$$

$$[(x_2-x)^2+(y_2-y)^2+(z_2-z)^2]^{1/2}+c(\tau_{t_r}-\tau_{t_o})=d_2$$

$$[(x_3-x)^2+(y_3-y)^2+(z_3-z)^2]^{1/2}+c(\tau_{t_r}-\tau_{t_o})=d_3$$

$$[(x_4-x)^2+(y_4-y)^2+(z_4-z)^2]^{1/2}+c(\tau_{t_r}-\tau_{t_o})=d_4$$

上述四个方程式中待测点坐标 x 、 y 、 z 和 τ_{t_o} 为未知参数，其中：

$$d_i=c\Delta t_i \quad (i=1, 2, 3, 4).$$

d_i ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 到接收机之间的距离。 Δt_i ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 的信号到达接收机所经历的时间。 c 为 GPS 信号的传播速度（即光速）。

四个方程式中各个参数意义如下：

x 、 y 、 z 为待测点坐标的空间直角坐标。

x_i 、 y_i 、 z_i ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 在 t 时刻的空间直角坐标, 可由卫星导航电文求得。

$V_{t i}$ ($i=1, 2, 3, 4$) 分别为卫星 1、卫星 2、卫星 3、卫星 4 的卫星钟的钟差, 由卫星星历提供。 $V_{t o}$ 为接收机的钟差。由以上四个方程即可解算出待测点的坐标 x 、 y 、 z 和接收机的钟差 $V_{t o}$ 。

目前 GPS 系统提供的定位精度是优于 10 米, 而为得到更高的定位精度, 我们通常采用差分 GPS 技术: 将一台 GPS 接收机安置在基准站上进行观测。根据基准站已知精密坐标, 计算出基准站到卫星的距离改正数, 并由基准站实时将这一数据发送出去。用户接收机在进行 GPS 观测的同时, 也接收到基准站发出的改正数, 并对其定位结果进行改正, 从而提高定位精度。差分 GPS 分为两大类: 伪距差分 and 载波相位差分。

1. 伪距差分原理

这是应用最广的一种差分。在基准站上, 观测所有卫星, 根据基准站已知坐标和各卫星的坐标, 求出每颗卫星每一时刻到基准站的真实距离。再与测得的伪距比较, 得出伪距改正数, 将其传输至用户接收机, 提高定位精度。这种差分, 能得到米级定位精度, 如沿海广泛使用的“信标差分”。

2. 载波相位差分原理

载波相位差分技术又称 RTK (Real Time Kinematic) 技术, 是实时处理两个测站载波相位观测量的差分方法。即是将基准站采集的载波相位发给用户接收机, 进行求差解算坐标。载波相位差分可使定位精度达到厘米级。大量应用于动态需要高精度位置的领域。

3.2.2 其它卫星导航定位系统

一、GLONASS

全球轨道导航卫星系统 (Global Orbiting Navigation Satellite System, Global'naya Navigatsionnaya Sputnikovaya Sistema, GLONASS) 是前苏联研制建立的, 1978 年开始研制, 1982 年 10 月开始发射导航卫星。自 1982 年至 1987 年, 共发射了 27 颗 GLONASS 试验卫星。它由 24 颗卫星组成卫星星座 (21 颗工作卫星和 3 颗在轨备用卫星), 均匀地分布在 3 个轨道平面内。卫星高度为 19100km, 轨道倾角为 64.8° , 卫星的运行周期为 11 时 15 分。GLONASS 卫星的这种空间配置, 保证地球上任何地点、任何时刻均至少可以同时观测 5 颗卫星。

二、北斗导航系统

北斗卫星导航定位系统，是中国自行研制开发的区域性有源三维卫星定位与通信系统（CNSS），是除美国的 GPS、俄罗斯的 GLONASS 之后第三个成熟的卫星导航系统。该系统由三颗（两颗工作卫星、一颗备用卫星）北斗定位卫星（北斗一号）、地面控制中心为主的地面部份、北斗用户终端三部分组成。可向用户提供全天候、二十四小时的即时定位服务，定位精度可达数十奈秒(ns)的同步精度，其精度与 GPS 相当。

“北斗导航系统”是全天候、全天时提供卫星导航信息的区域导航系统，此系统由二颗卫星、控制站和接收机组成，卫星编号分别为“北斗一号”和“北斗二号”，分别于 2000 年 10 月 31 日凌晨 0 时 02 分和 2000 年 12 月 21 日 0 时 20 分在西昌卫星发射中心发射升空，并准确进入预定轨道。2003 年 5 月 25 日，我国第三颗“北斗一号”卫星发射成功，6 月 1 日，我国自主研发的“北斗运营服务平台”正式开通，这标志着我国已经拥有了完全自主的卫星导航系统，北斗导航定位系统的大规模应用进入了实质性阶段。导航通信卫星是 2 颗地球同步卫星，距离地面 36000km，位于赤经 80E 和 140E，还有 1 颗备用卫星，将位于赤经 110.5E，2 颗卫星的升交点赤经相差 60。2 颗卫星于 2000 年 10 月 31 日和 12 月 21 日发射成功。覆盖范围东经约 70° — 140° ，北纬 5° — 55° 。

双星定位原理由陈芳允院士提出。双星定位系统实际上综合了卫星导航和卫星通信两种技术，因而兼容了两者的功能。该系统主要由三部分组成：两颗相隔一定距离的静止轨道卫星、用户终端和地面控制中心有两副天线，分别对准两颗静止卫星。北斗系统由 2 颗地球静止卫星（GEO）对用户双向测距，由 1 个配有电子高程图库的地面中心站进行位置解算。定位由用户终端向中心站发出请求，中心站对其进行位置解算后将定位信息发送给该用户。而它的定位则是基于三球交会原理，即以 2 颗卫星的已知坐标为圆心，各以测定的本星至用户机距离为半径，形成 2 个球面，用户机必然位于这 2 个球面交线的圆弧上。中心站电子高程地图库提供的是一个以地心为球心、以球心至地球表面高度为半径的非均匀球面。求解圆弧线与地球表面交点，并已知目标在赤道平面北侧，即可获得用户的二维位置。

3.2.3 GPS 的物流功能

随着 GPS 技术的民用化和信息技术的广泛普及，GPS 在物流领域的应用越

来越广泛。比较突出的有如下一些功能：

1. 实时监控功能

在任意时刻通过发出指令查询运输工具所在的地理位置（经度、纬度、速度等信息）并在电子地图上直观地显示出来。

2. 双向通讯功能

GPS 的用户可使用 GSM 的话音功能与司机进行通话或使用本系统安装在运输工具上的移动设备的汉字液晶显示终端进行汉字消息收发对话。驾驶员通过按下相应的服务、动作键，将该信息反馈到网络 GPS，质量监督员可在网络 GPS 工作站的显示屏上确认其工作的正确性，了解并控制整个运输作业的准确性（发车时间、到货时间、卸货时间、返回时间等等）。

3. 动态调度功能

调度人员能在任意时刻通过调度中心发出文字调度指令，并得到确认信息。可进行运输工具待命计划管理，操作人员通过在途信息的反馈，运输工具未返回车队前即做好待命计划，可提前下达运输任务，减少等待时间，加快运输工具周转速度。可进行运能管理，将运输工具的运能信息、维修记录信息、车辆运行状况登记处、司机人员信息、运输工具的在途信息等到多种信息提供调度部门决策，以提高重车率，尽量减少空车时间和空车距离，充分利用运输工具的运能。

4. 数据存储、分析功能

实现路线规划及路线优化，事先规划车辆的运行路线、运行区域，何时应该到达什么地方等，并将该信息记录在数据库中，以备以后查询、分析使用。可进行可靠性分析，通过汇报运输工具的运行状态，了解运输工具是否需要较大的修理，预先做好修理计划，计算运输工具平均天差错时间，动态衡量该型号车辆的性能价格比。可进行服务质量跟踪，在中心设立服务器，并将车辆的有关信息（运行状况，在途信息，运能信息，位置信息等用户关心的信息）让有该权限的用户能异地方便地获取自己需要的信息。同时还可对客户索取的信息中的位置信息用相对应的地图传送过去，并将运输工具的历史轨迹印在上面，使该信息更加形象化。依据资料库储存的信息，可随时调阅每台运输工具的以前工作资料，并可根据各管理部门的不同要求制作各种不同形式的报表，使各管理部门能更快速、更准确地作出判断及提出新的指示。

3.2.4 GPS 在物流领域的应用

1. 用于汽车自定位、跟踪调度

据丰田汽车公司的统计和预测,日本车载导航系统的市场在 1995 年至 2000 年间将平均每年增长 35% 以上,全世界在车辆导航上的投资将平均每年增长 60.8%, 因此, 车辆导航将成为未来全球卫星定位系统应用的主要领域之一。我国已有数十家公司在开发和销售车载导航系统。

2. 用于铁路运输管理

我国铁路开发的基于 GPS 的计算机管理信息系统, 可以通过 GPS 和计算机网络实时收集全路列车、机车、车辆、集装箱及所运货物的动态信息, 可实现列车、货物追踪管理。只要知道货车的车种、车型、车号, 就可以立即从近 10 万公里的铁路网上流动着的几十万辆货车中找到该货车, 还能得知这辆货车现在何处运行或停在何处, 以及所有的车载货物发货信息。铁路部门运用这项技术可大大提高其路网及其运营的透明度, 为货主提供更高质量的服务。

3. 用于军事物流

全球卫星定位系统首先是因为军事目的而建立的, 在军事物流中, 如后勤装备的保障等方面, 应用相当普遍, 尤其是在美国, 其在世界各地驻扎的大量军队无论是在战时还是在平时都对后勤补给提出很高的需求, 在战争中, 如果不依赖 GPS, 美军的后勤补给就会变得一团糟。美军在 20 世纪末的地区冲突中依靠 GPS 和其他顶尖技术, 以强有力的、可见的后勤保障, 为保卫美国的利益做出了贡献。目前, 我国军事部门也在运用 GPS。

第4章 概要设计

管理信息系统(MIS)产生于20世纪60年代初到70年代初期,但是,它至今仍是物流领域最重要的信息系统,其应用范围广泛,实用价值很高。国内外在物流领域中应用管理信息系统的结果证实,物流的各个领域都可以通过计算机为基础的管理信息系统得到改善。统计证实,应用管理信息系统,常常可使生产率提高10%—15%。物流管理信息系统以物流过程为特定的对象,把物流和物流信息结合成一个有机的系统。这个系统用各种方式选择收集、输入物流计划的、业务的、统计的各种有关数据,经过有针对性的、有目的的计算机处理,即根据管理工作的要求,采用特定的软件技术,对原始数据处理后输出对管理工作有用的信息。物流决策支持系统(DSS)。决策支持系统兴起于20世纪70年代中期到80年代初,它是以现代信息技术为手段,以支持半结构化和非结构化决策过程为特征的计算机决策辅助系统。决策支持系统是管理信息系统的高级形式和向纵深的延伸,其任务是利用信息系统所提供的信息和辅助决策的计算机软件辅助管理者和领导者进行决策,甚至模拟思维过程进行智能化的模拟决策,向更高级的人工智能自动化系统发展。在物流领域中,需要依靠决策支持系统。在物流活动开始之前科学地做好决策,才能保证最高的成功率。电子数据交换系统(EDI)。电子数据交换系统是对信息进行交换和处理的网络自动化系统,是将远程通信、计算机及数据库三者有机结合在一个系统中,实现数据交换、数据资源共享的一种信息系统。这个系统也可以作为管理信息系统和决策支持系统的重要组成部分。采用电子数据交换系统之后,信息交换可由两端直接进行,而越过很多中间环节,这就使物流过程中每个衔接点的手续大大简化,由于减少甚至消除了物流各个过程中的单据凭证,不但减少了差错,而且大大提高了工作效率。

4.1 系统总体架构

4.1.1 持久层

持久层(Persistence layer)也叫设备层是整个系统的最底层,它包括了整个系统的所有设备如和计算机相关的部分设备:服务器、终端机、局域网设备、广域网设备以及其它仓储自动化设备如:自动堆垛机、电子显示屏、其他射频

设备等等。该层为整个网络系统提供了一个数据显示和交换的平台，它的主要设备的选取和网络拓扑结构的设计将直接决定系统的工作效率和维护成本。

1. 网络布局

在物流仓库系统中，计算机网络采用的是以 INTRANET（内部网）为主、附之以连入 INTERNET（互联网）的体系结构。如图 4-1 所示。系统采用的是典型的集中式方案，即数据中心包括数据库服务器、应用服务器、DDN 专线；各仓库只需安装 PC 终端，通过本地局域网和电信 DDN 专线与数据中心相连，通过中心应用服务器访问中心数据库。由于各仓库与数据中心是以电信专线方式连接，这个网络实质上是一个内部的专用网络，所以称作内部网（INTRANET）。在安全性和速度上基本不受外界的影响，比互联网（INTERNET）环境有更强的技术保障。同时，为了加强与外界的联络，数据中心有一个出口连入互联网（INTERNET），这样，部分工作如内部与外部的信息交流可以通过互联网（INTERNET）方式来实现，客户的存货信息查询也可通过互联网来进行。

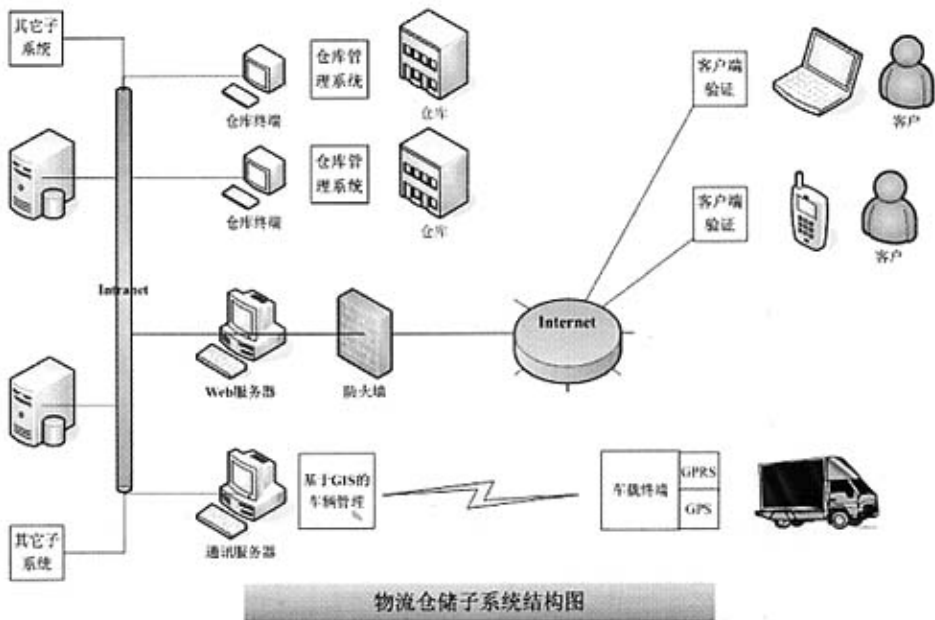


图 4-1 物流仓储系统结构图

先进的系统网络布局使仓库应用系统具备多种优势：

- (1) 各仓库业务流程标准，生产数据统一，服务规范；
- (2) 各仓库之间的数据信息可以实时处理；

- (3) 减少网络故障点, 增加了系统的运行可靠性;
- (4) 各仓库系统建设前期投资少, 见效快;
- (5) 系统维护专业化, 各仓库系统维护成本低廉;
- (6) 系统升级快捷方便。

2. 数据库应用

目前, 面向企业级应用的数据库系统以关系型数据库为主导产品。主流关系型数据库管理系统有 Oracle、Sybase、MS SQL Server、Informix、Ingres 和 DB2。由于整个系统在 Windows 平台上开发, 再针对物业管理系统中数据量大小的估计, 本系统选择 MS SQL Server2000 Enterprise Edition 作为后台数据库服务器。MS SQL Server2000 是目前广泛使用的一种数据库服务器, 它具有以下特点:

(1) SQL Server 是一种高效的关系数据库系统, 它与 Windows NT/2000 及 Windows 9x 等操作系统紧密集成。这种安排使 SQL Server 能充分利用操作系统所提供的特性。对于今天复杂的客户/服务器系统来说, SQL Server 是一个很好的选择;

(2) 无范式要求, 可根据实际系统需求构造数据库;

(3) 采用标准的 SQL 结构化查询语言, 具有分布优化查询功能;

(4) 具有丰富的开发工具, 覆盖开发周期的各阶段;

(5) 通过 SQL*DBA 控制用户权限, 提供数据保护功能, 监控数据库的运行状态, 调整数据缓冲区的大小;

(6) 具有数据透明、网络透明, 支持异种网络、异构数据库系统。

4.1.2 业务逻辑层

业务逻辑层(Business Logic layer)是表示层和持久层之间的桥梁, 它响应表示层的用户请求, 根据用户的请求生成 SQL 语句检索或更新数据库, 并把结果返回给客户端。从编码的视角来看, 这一层是最容易被忽视的一层。我们往往在表示层或持久层周围看到这些业务处理的代码, 这其实是不正确的, 因为它导致了程序代码的高耦合, 这样一来, 随着时间的推移这些代码将很难维护。针对这一问题现在有不少可以选择的框架, 最受欢迎的两个框架是 Spring 和 PicoContainer。这些也被称为轻量级容器(micro containers), 它们能够很好的把对象搭配起来。这两个框架都着手于‘依赖注入’(dependency

injection)。关于 Spring 的依赖注入和面向方法的编程，其实是通过一个给定参数的 Setter 方法来构造 Bean 的。此外 Spring 的 Setter 方法还提供了 Setter Injection(type2)，Constructor Injection(type3) 等方式供我们选择。Spring 把程序中所涉及到包含业务逻辑和数据存取对象 (DataAccess Object) 如：transaction management handler (事物管理控制)、Object Factories (对象工厂)、service objects (服务组件) 都通过 XML 配置联系起来。

业务逻辑层的作用如下：

- (1) 处理应用程序的业务逻辑和业务校验；
- (2) 管理事物；
- (3) 提供与其它层相互作用的接口；
- (4) 管理业务层级别的对象的依赖；
- (5) 在表示层和持久层之间增加了一个灵活的机制，使得他们不需要直接的联系在一起；
- (6) 通过揭示从表示层到业务逻辑层之间的上下文，来得到业务逻辑；
- (7) 管理程序的执行 (从业务逻辑层到持久层)。

4.1.3 表示层

表示层 (Presentation layer) 也叫用户界面层，其主要的作用是处理用户的输入和向用户提供输出，但并不负责解释其含义。界面层同时也提供一定的安全性，确保用户看到符合自身权限的信息。这一层通常由前端工具 Jsp, Applet 等实现。

表示层所负责的功能如下：

- (1) 管理用户的请求，作出相应的响应；
- (2) 各仓库之间的数据信息可以实时处理；
- (3) 减少网络故障点，增加了系统的运行可靠性；
- (4) 各仓库系统建设前期投资少，见效快；
- (5) 系统维护专业化，各仓库系统维护成本低廉；
- (6) 系统升级快捷方便。

4.2 部分主要功能

物流仓储管理系统作为舵落口大市场管理系统的一个子系统模块。其作用

是为舵落口大市场管理部门及租用大市场仓库的商户们提供了一个集中交互的管理平台，通过该系统，商户可以及时了解仓库的使用情况，随时知道自己货物的位置，了解租金及资费的相关情况，在管理部门不论是普通员工，出纳，会计还是管理者都能通过这个系统及时查询自己需要的信息。该系统不但提供了全面的信息服务，还做到了在部门内实现了垂直管理，在部门间实现了平行交互。此外系统在实现信息的录入，查询，管理的同时尽可能的保证了信息的安全及系统的安全。通过划分用户组，分配用户权限充分弥补了传统管理中容易出现失误及漏洞。

此外整个大市场管理系统还有网站，安防，短信平台等几个子系统，为了实现各子系统间的数据交换本系统还特别引入了一个数据交换功能。网站，安防，短信平台等几个子系统因由小组其他成员负责开发，且不是本论文主要研究内容，故这里不再赘述。

系统的主要功能分析如图 4-1 所示：

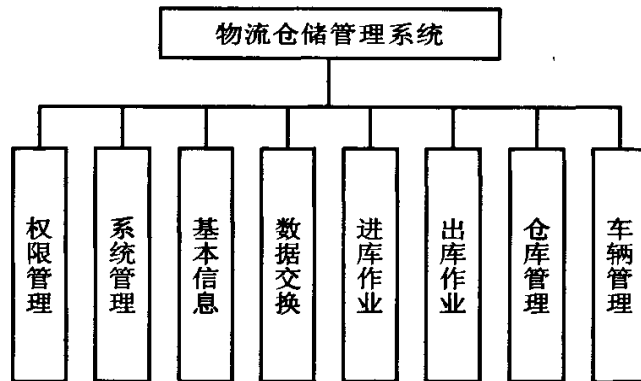


图 4-2 物流仓储管理系统结构图

4.2.1 权限管理

用户权限管理作为物流仓储管理系统中的一个模块，担负着增强系统的安全性及可行性的任务。由于大市场中人员众多而复杂，因此有必要针对不同的用户角色设计不同的系统功能界面。整个系统设计了用户权限管理安全机制，用户对系统的操作只能在事先规定的权限范围内进行；而系统管理员、授权管理员和用户密码的控制上具有相互制约关系。用户登录后，系统自动根据权限提供相应的界面，用户只能对相关的事务进行处理。系统管理员可以根据企

业的组织构架对系统进行划分, 并进行相应的用户组、用户和角色设定。

用户组设定: 可以对大市场内各企业部门的组织架构进行划分和设定, 包括设定部门名称、部门经理、部门成员、成员邮件列表以及部门的层次架构, 从而建立起各部门之间的关联关系。

用户设定: 用户是组织架构的基础信息, 根据不同用户的级别设定相应的权限。

角色设定: 角色的应用有助于系统将工作岗位与涉及的具体用户进行分离, 这样当企业调整组织机构或用户时, 系统管理员只需要对该角色涉及的用户作简单的调整, 即可保证系统的正常运行, 不必大规模地修改 workflow 或用户的权限控制, 从而将系统管理员地维护工作量减少到最低。

4.2.2 系统管理

该功能的第一个作用是针对大市场中不同的人员设置不同的使用该系统的权限。根据管理层次设置四个级别的用户权限: 读取权、修改权、新增权、删除权。根据不同的用户权限, 将用户分为普通用户、管理员和超级管理员三种类别。分类用户按照分配的帐号登录系统。其中普通用户只能浏览指定页面的信息并只能修改自己的登陆密码; 管理员用户除了具备普通用户功能外, 还可以对所管理范围内用户的权限进行修改, 例如: 增加用户, 修改权限, 删除用户, 修改密码; 超级管理员负责维护整个系统的正常运行, 一般由专业的技术人员担当。他们具有整个系统的调配权。

该功能的第二个作用是数据的备份与恢复, 防止丢失。备份可由管理员进行操作也可以是系统在指定时间自动进行备份。备份的内容包括系统通用代码以及数据库储存信息。

4.2.3 数据交换

目前在物流领域被广泛使用的是 EDI, 也就是电子数据交换 (Electronic Data Interchange), 国际数据交换协会 (IDEA) 定义为: 不同计算机系统之间电子手段传递约定的电文标准组成的信息数据的应用技术。EDI 是电子商务的一种, 真正意义上的电子商务不仅包括 EDI, 还包括网络购物、网上支付、网络服务提供等许多方式, 而 EDI 只是一种狭义的电子商务。它的服务对象只限于企业与企业之间, 并不向普通的消费者提供服务。EDI 实际上是一种企业与企业之间的封闭式的物流购销网络, 简单地说就是企业的内部应用系

统之间，通过计算机和公共信息网络，以电子化的方式传递商业文件的过程。换言之，EDI 就是供应商、零售商、制造商和客户等在其各自的应用系统之间利用 EDI 技术，通过公共 EDI 网络，自动交换和处理商业单证的过程。

电子数据交换技术自问世以来，因其技术先进，可大大减少贸易文件及文件处理成本，因而受到世界各国普遍重视，发展迅速。现在，EDI 用户根据国际通用的标准格式编制电文，以机器可读的方式将结构化的信息（如发票、海关申报单、进出口许可证等“经济信息”）按照协议经过通信网络传送。报文接受方按国际统一规定的语法规则对报文进行处理，通过相应的管理信息系统，完成综合的自动交换和处理。EDI 遵循一定的国际标准或行业规则，自动地进行数据发送、传送及处理，而不需人工介入，从而实现事务处理或贸易自动化。联合国欧洲经济理事会（UN/ECE）经过多年来的大量工作，于 1987 年公布了一套 EDI 国际标准，命名为 UN/EDIFACT，而国际标准化组织 ISO 为该标准制定了一套语法规则（SYNTAX RULES, ISO9735），UN/EDIFACT 是联合国推荐的用于行政、商业和运输业的电子交换标准报文格式。EDI 技术发展的重点任务之一是统一报文格式。目前，UN/EDIFACT 标准已占据全球 EDI 标准的主导地位。

本论文所研究的数据交换因为没有在广域网条件下的大规模数据传输要求，所以在此不考虑基于 VPN 等虚拟专用网的复杂应用。但依然要实现多子系统平台下的数据交换，大市场管理系统间的数据交换如下图 4-2 所示：

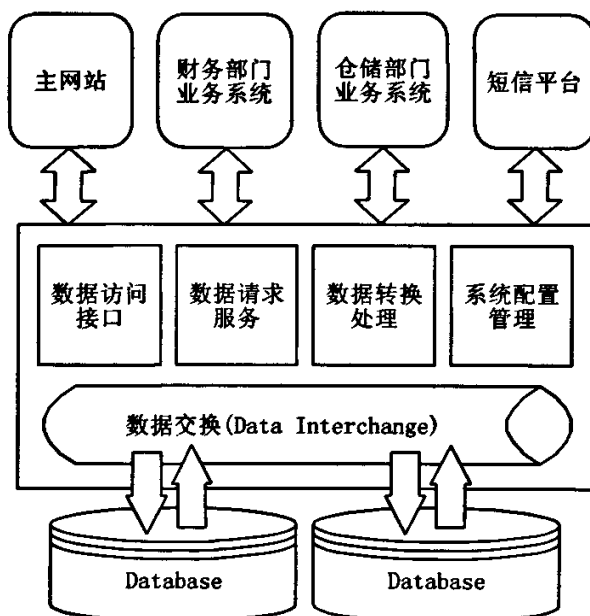


图 4-3 数据交换示意图

4.2.4 入库作业

商品入库是仓储业务的第一阶段，是指商品进入仓库储存时所进行的商品接收、卸货、搬运、清点数量、检查质量和办理入库手续等一系列活动的总称。它包括商品入库的准备、入库商品接运与交接、验收、入库等过程。

(1) 入库商品的接收；入库商品的接收主要有四种方式：车站码头接货；专用铁路线或码头接货；到供货方仓库提货；本库接货。

(2) 入库商品的验收；入库商品的验收工作，主要包括数量验收、质量验收和包装验收三个方面。在数量和质量验收方面应分别按商品的性质、到货情况，来确定验收的标准和方法。

(3) 验收发现问题的处理；验收中出现的问题，大体有如下几种情况：①数量不符；②质量问题；③包装问题；④单货不符或单证不全。

(4) 办理商品入库手续；商品经过质量和数量验收后，由商品检查人员或保管员在商品入库凭证上盖章签收。仓库留存商品入库保管联，并注明商品存放的库房、货位，以便统计、记账。同时，将商品入库凭证的有关联迅速送回存货单位，作为正式收货的凭证。

此外入库操作还应严格遵守储位分配原则，具体如下：

(1) 同客户货品相邻原则；

(2) 同种货品相邻原则；

(3) 货品不相互污染原则；

(4) 如果规格经验数据没有相应货品存储信息，根据货品与储位属性分配一个保守数量；

(5) 货品放满储位原则。

具体流程如下图：

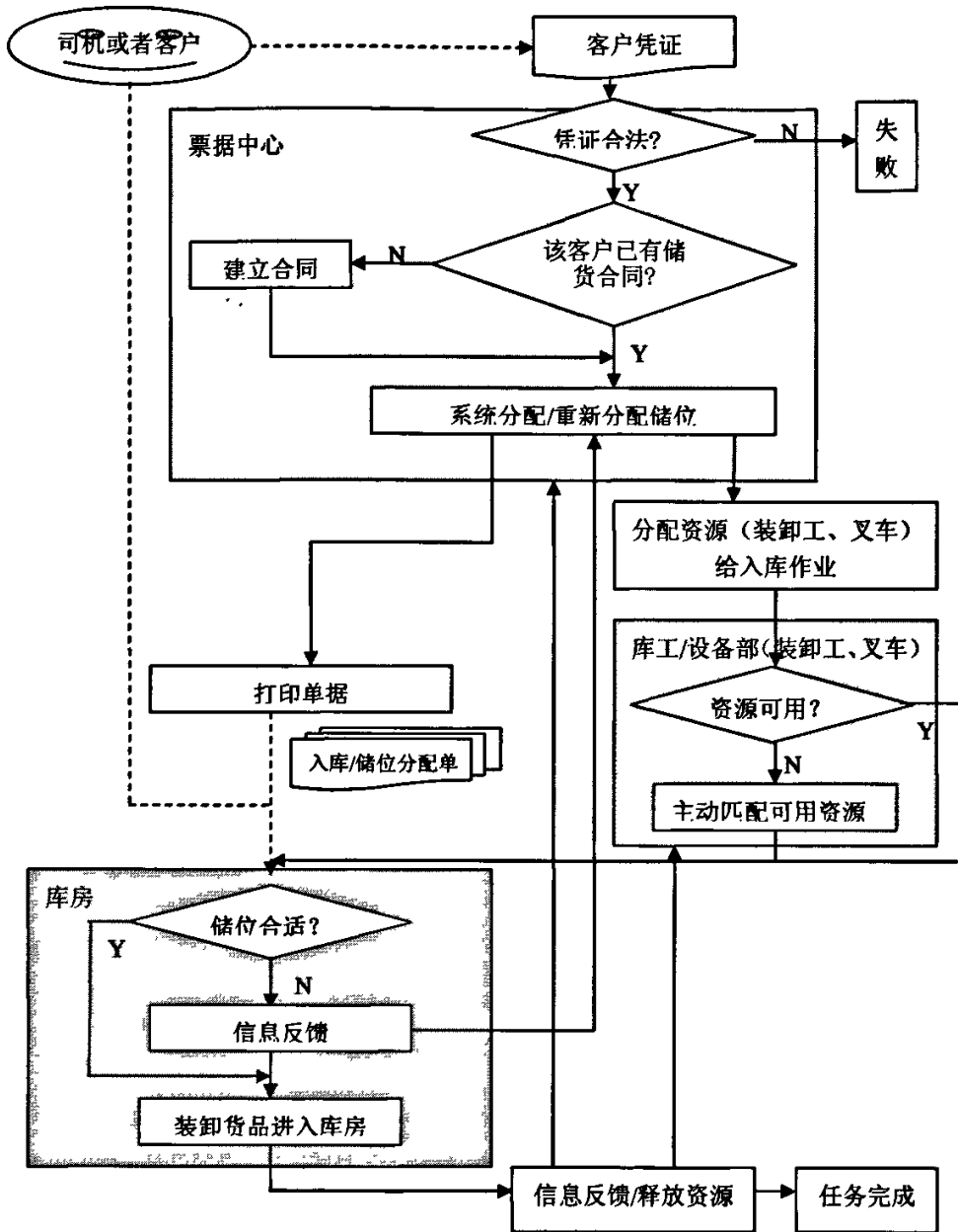


图 4-4 入库作业流程图

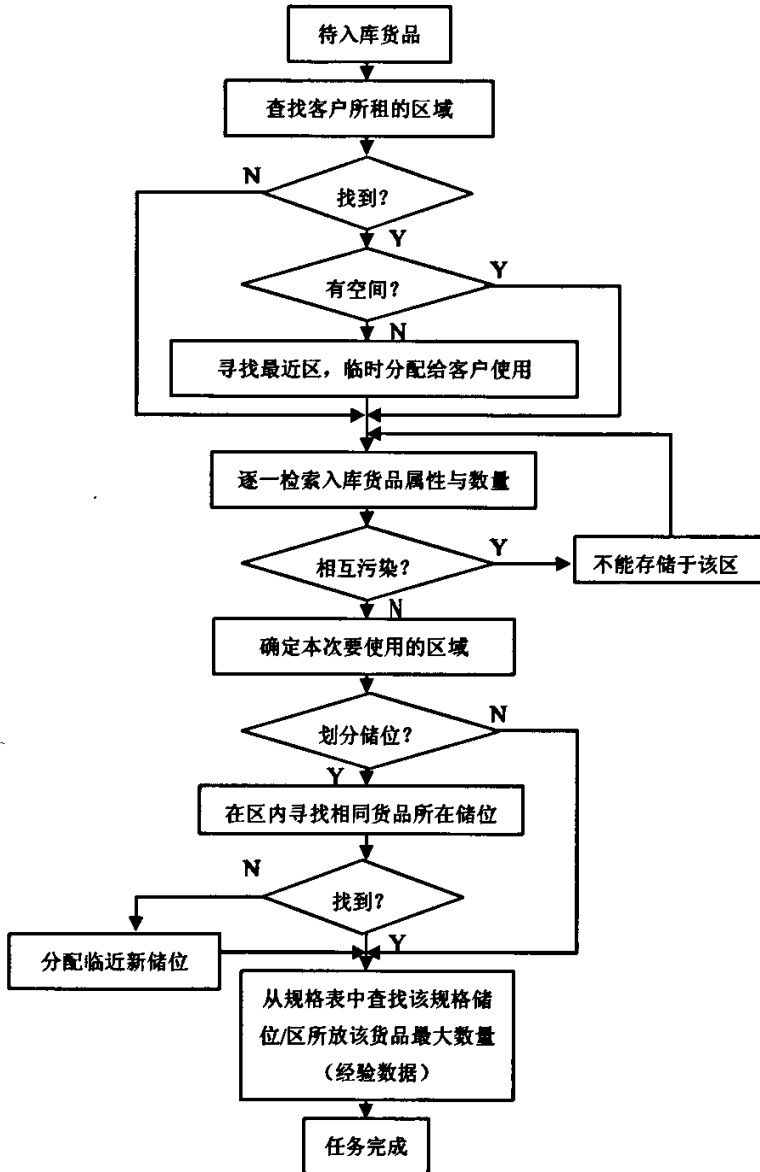


图 4—5 储位分配作业流程图

入库作业操作中对失败、异常的捕获以及反馈:

(1) 装卸作业失败; 失败原因: 调用人力资源和设备无法满足卸货及入库需要。解决方法: 装卸负责人返回库工部或与库工部联系重新安排资源, 得到新资源继续作业。注意: 重新安排资源后要修改作业人力资源表和作业设备表, 保证系统记录与实际相符。

(2) 装卸作业异常；异常情况：人力或者设备损坏货品。解决方法：由司机、库管员、作业负责人、实际损坏人四方共同协商，达成货品损失情况，由司机询问货品所属客户损失物品处理办法。损失金额可以过后再议。作业负责人在装卸单上，入库单上填写货品损失情况，如果损坏货品运回，则入库单实收要减少，如果损坏货品照样入库，由库管员负责处理把损坏货品单独处理（放入残品区）。

(3) 装卸作业反馈；由作业负责人返回给库工部装卸单。

(4) 放货失败；失败原因：未分配储位区货品存放空间不足。解决方法：由库管员联系票据中心或者自己解决，放置到其他区。

(5) 储位分配失败；失败原因：货品体积，重量估计错误。解决方法：主动联系票据中心，重新分配储位（根据实际情况附建议分配方案）。

(6) 库管员反馈；根据实际情况，反馈某区规格储位存放某种商品的最大数量，有利于以后分配方案。一般库管员在接收到储位分配表时每种货品每个储位会表明该分配是由经验数据得到还是优化得到，库管员要负责把表明是优化得到的测试出最安全最大数量并且返回给系统。

(7) 司机反馈；司机负责带回入库单给票据中心，系统接收实际入库数量，打印回执及出门条给司机。

4.2.5 出库作业

出库的概念，是指仓库根据出库凭证，将所需物料发放给需求部门所进行的各项业务活动。出库的基本要求即根据正式的凭证和手续，准确、及时地组织好出库工作，具体如下：

(1) 物资出库必须准确：准确是工作质量的一个重要标志，没有准确就没有质量。没有准确，出库工作就变得毫无意义。所谓准确就是按照出库凭证所列的物资编号、品名、规格、质量、等级、单位数量等，准确无误的进行点交，做到单货相符，避免差错。

(2) 物资出库必须及时：发货及时是保证生产建设和人民需要的重要条件。因此，发货时在手续健全的前提下、力求简便、加快速度及时组织好物资出库作业。

(3) 物资出库必须安全：所谓出库安全，就是在出库搬运点交时注意安全操作，防止物资震坏、摔伤、破损、变形，以保证物资出库时的质量完好。

出库的原则如下:

(1) 先进先出。 为避免物资长期在库存放而超过其储存期限或增加自然损耗,因此必须坚持“先进先出、后进 后出”的原则;

(2) 凭证发货。 物资“收有据、出有凭”是物资收发的重要原则,所谓凭证发货就是指出库必须凭正式单据和 手续,非正式凭证或白条一律不予发放(国家或上级指令的、紧急抢险救灾物资除外)。

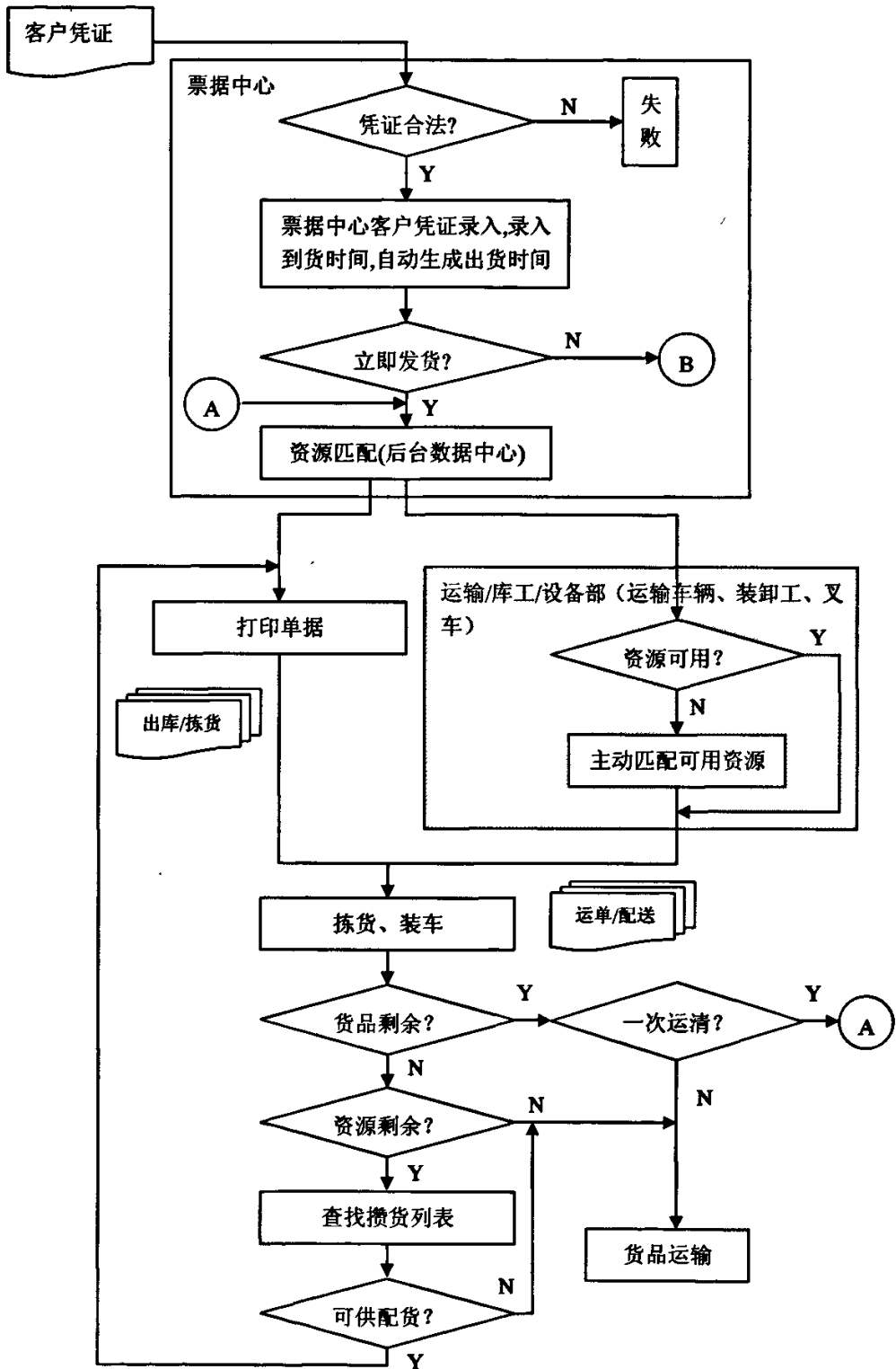
出库方式主要有以下三种:

(1) 自提:即提货单位持出库凭证(提货单)自行到仓库提货,保管员根据提货单上所列的名称、规格、数量当面点交给提货人员;

(2) 送货:仓库受提货单位委托,将其所需物资,按提货单所列内容运送到使用单位,并在使用单位当场点交;

(3) 代运:仓库受外埠用户委托,按单将货配齐后通过铁路、水运、航空、邮寄等方式,将货发至用户所在地的车站、码头、邮局提货。此种出库形式的交接,是与铁路、水运等运输部门进行的,仓库按规定程序办理完托运手续并取得运输部门的承运凭证,将应发货物全部点交承运部门后,责任才开始转移。

具体流程如下图所示:



转到下一页:

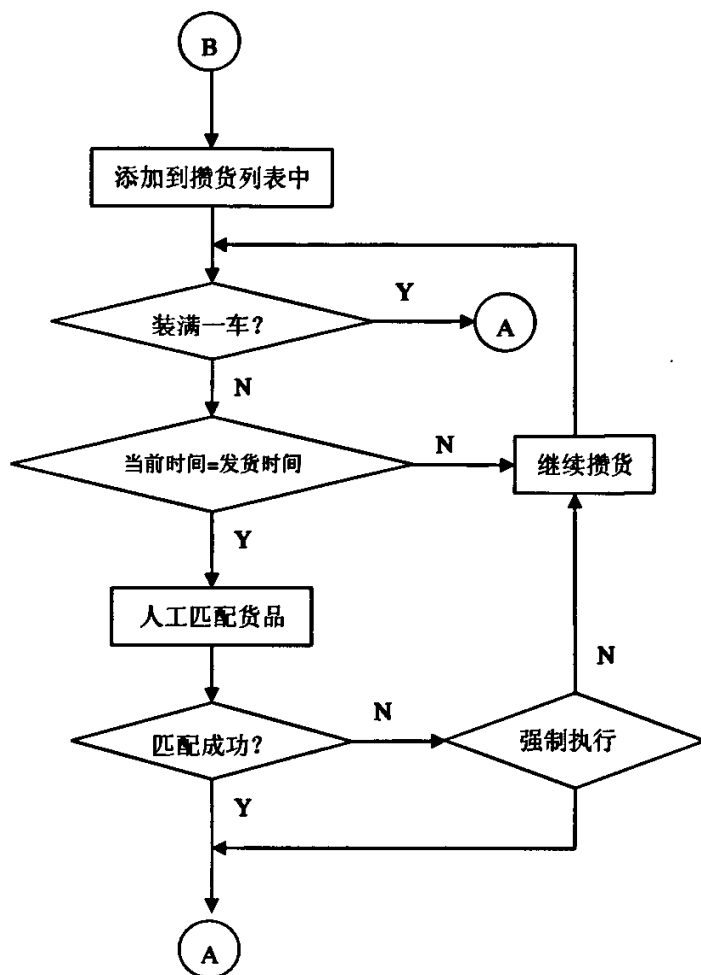


图 4—6 出库作业流程图

4.2.6 仓库管理

仓库管理的目的是建立有效的仓库管理体系，确保仓储物资的储存安全及发放准确，杜绝各项事故的发生，保障大市场的各项业务与生产的正常运作。首先是储存区域的划分，系统会根据先期采集的 GIS 底层数据来生成相应的区域地图，仓库的平面图等。仓库管理员根据仓库的面积、储存物品特性和包装、储存物品进出频率划分仓库的储存区域，保证仓库的储存空间得以最大效率使用。相关的仓库库存物资的位置信息，状态信息也可以由数据交换来保持实时

更新。仓库管理员还可以根据根据产品类别、销售类别、计划产品、订制产品、成品、半成品分区存放，同时根据库区；仓储物资的产品类型、状态等因素对库存物资进行编号，并在电子地图相应位置进行标示。

4.2.7 车辆管理

车辆是物流的血液，如何对车辆进行有效的管理，对物流线路进行优化将决定整个物流系统的优劣。传统的车辆管理主要还是集中在记录车辆的一般属性如：车型、车牌号、司机等。如果系统通过与 GPS、GIS 等子系统进行数据交换，则可以实现更多的功能。车辆管理模块可以实现的主要功能如下：

(1) 车辆和货物跟踪

利用 GPS 和电子地图可以实时显示出车辆或货物的实际位置，并能查询出车辆和货物的状态，以便进行合理调度和管理。用于解决点对点，点对多，多对多的货物运输，并降低物流作业费用，保证服务的质量。

(2) 提供运输路线规划和导航

规划出运输线路，使显示器能够在电子地图上显示设计线路，并同时显示汽车运行路径和运行方法。

(3) 信息查询

对配送范围内的主要建筑、运输车辆、客户等进行查询，查询资料可以文字、语言及图象的形式显示，并在电子地图上显示其位置。

(4) 模拟与决策

如可利用长期客户、车辆、订单和地理数据等建立模型来进行物流网络的布局模拟，并以此来建立决策支持系统，以提供更有效而直观的决策依据。

第 5 章 物流管理辅助决策

5.1 基于 GIS 的物流平台的数据组织

GIS 使用的数据大多为空间数据,其基本构成单元可以分为点(Point)、线(Line)、面(Polygon)、复杂物体(Complex Object)等,并以空间数据库的形式进行数据组织,空间实体的属性信息作为属性项进行存储和管理。传统物流管理采用的数据大多是非空间数据,通常可以以二维表格的形式进行数据组织,可以采用通用的关系数据库进行数据的存储和管理,构成非空间数据库。

GIS 的数据基础与传统物流管理信息系统的数据基础显然有巨大的差异,基于 GIS 的物流信息平台建设的关键问题之一是对异构数据的处理。数据集成需要将异构数据规划为同构数据,如基于元数据的系统集成、基于 RDBMS 的集成等;或者将异构数据规划为同构的过渡数据,如基于 SQL 和 DLL 的集成等,其目的在于形成的数据或过渡数据可以直接被系统所使用。比较理想的方式是基于 RDBMS 的集成方式,在同一数据库中采用同构的方式同时存储空间数据、非空间数据、影像数据等,采用标准 SQL 语言可以实现 DB 与 DBNS 的分离,Oracle、DB2、Informix 等均支持上述操作。

下图 5-1 显示了基于 GIS 的物流信息平台的数据组织模式。

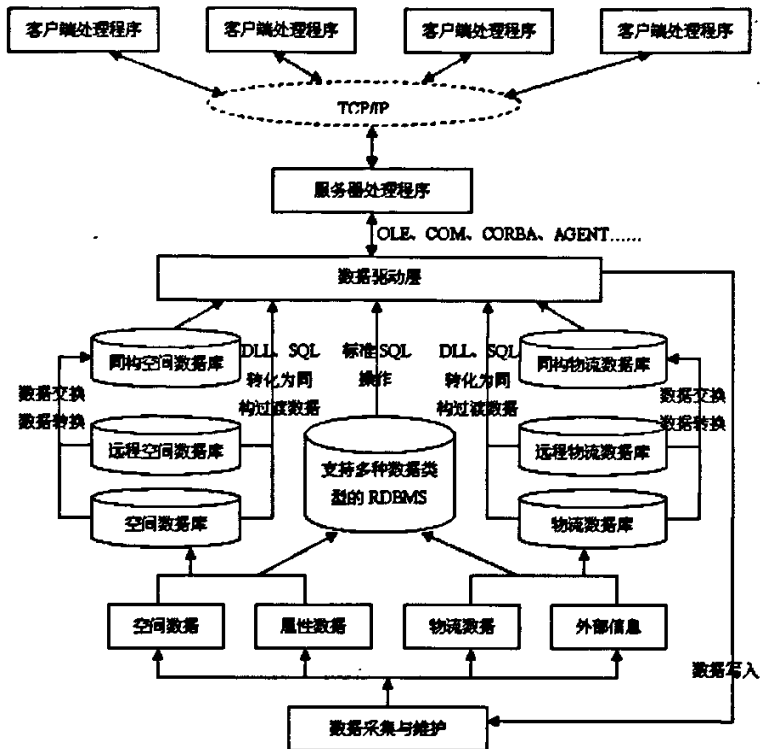


图 5-1 基于 GIS 的物流平台数据组织

5.2 最短路径问题

路径规划是物流辅助决策最基本的功能之一。若想对物流过程中车辆运行路径进行有效的规划使其做到成本最低耗时最少，就必须研究各种基于 GIS 的最短路径问题。最短路径问题有几种常见的形式：单源单目的——常被用来求解甲地到乙地的最短路径，多用于固定两地间的往返运输；单源多目的——也就是在甲地装载一次访问多个目的地，多用于多点物流以及仓储配送业务。对于也单源多目的的最短路径问题通常也被称作旅行商问题——从起点出发一次访问多个城市并返回起点的最短路径问题。以下将结合多种求解最短路径问题的经典算法来讨论基于 GIS 的物流辅助决策。

5.2.1 图的存储实现

对于目前大多数的以图层结构组织的 GIS 来说，存储道路信息的图层称之为道路层。为了在道路层中表示现实世界里的道路状况，我们引入了节点和弧

段的概念。就如同数据结构中的带权有向图一样，道路的交叉路口被表示成节点，路口之间联通的道路被表示成弧段，而弧段上的权值就可表示两个路口之间的距离或者途中所需的时间等。以 Mapinfo 为例，表示数据表结构的 .tab 文件中记录了相应图层的属性数据：

节点表

ID	经度	纬度	节点名称
----	----	----	------

弧段表

ID	起始节点	终点节点	正向权值	反向权值	弧段名称
----	------	------	------	------	------

这种采用双向权值的表示方法可以很好的体现交通网络的有向性，记录了如单行道（某个方向的权值无限大），立交桥（双向权值不相等）等特殊权值的道路。表示拓扑网络的数据结构还有很多，如邻接矩阵，邻接表，十字链表等。虽然用邻接矩阵存储道路信息所产生的图明显是个稀疏矩阵，可能会造成大量的数据冗余，但就目前的计算机硬件发展水平来看，普通 pc 已经可以支持较大量的数据存储和计算。在节点数不超一万个的城市地图中，系统完全可以满足算法的开销。采用以邻接矩阵来表示顶点间的相邻关系及各条弧段的带权，这样的—个 n 阶方阵就可以记录所有节点之间的关系和节点间双向的权值。

设 G 是一个有序二元组，那么有图 $G=(V, E)$ 中 V 是图的非空顶点集合， E 是图 G 的所有存在的边且边 $e_{n,m}$ 中的下标 n, m 表示 e 为一条有向的边，方向为节点 $n \rightarrow$ 节点 m ，那么就有：

$$G = (V, E)$$

$$V = \{v_1, v_2, v_3, v_4, v_5, \dots\}$$

$$E = \{e_{(1,2)}, e_{(1,3)}, e_{(1,4)}, e_{(1,5)}, e_{(2,1)}, e_{(2,3)}, \dots\}$$

对于 G 的每一条边 e 有一个实数 $W(e_{n,m}) \in (0, +\infty)$ 称为 e 从节点 n 到节点 m 的权，权值为 0 时表示从一个节点到另一个节点该方向的道路不存在或目标节点不可达。在动态权值的系统中通过权值的变化还可以表示一条路段某方向车辆堵塞程度，权值越大表示的堵塞情况越严重， $+\infty$ 则是完全堵死。在实际的算法设计中一般由某种数据类型的最大值表示，如 `int` 的最大数值为 32767; `uint` 的

最大数值为 65535。

5.2.2 Dijkstra 算法

Dijkstra 是由荷兰算法学家 Edgar Wybe Dijkstra 于 60 年代初期提出来的, 非常适合在带权有向图中求解单源单目最短路径问题。

已知 V 中有节点 s 及 t , 用 Dijkstra 算法找出从 s 到 t 的最短路径。算法初始化, 源点 s 的路径长度值被赋为 0 ($d[s]=0$), 同时把所有其他节点的路径长度设为无穷大, 即表示不知道任何通向这些节点的路径 (对于 V 中所有节点 v 除 s 外 $d[v]=\infty$)。当算法结束时, $d[v]$ 中储存的便是从 s 到 t 的最短路径, 或者如果路径不存在的话是无穷大。Dijkstra 算法的基础操作是边的拓展: 如果存在一条从 u 到 v 的边, 那么从 s 到 u 的最短路径可以通过将边 (u,v) 添加到尾部来拓展一条从 s 到 v 的路径。这条路径的长度是 $d[u]+w(u,v)$ 。如果这个值比目前已知的 $d[v]$ 的值要小, 我们可以用新值来替代当前 $d[v]$ 中的值。拓展边的操作一直执行到所有的 $d[v]$ 都代表从 s 到 v 最短路径的花费。这个算法经过组织因而当 $d[u]$ 达到它最终的值的时候每条边 (u,v) 都只被拓展一次。

算法维护的两个节点集 S 和 V , 集合 S 保留了已知的所有的 $d[v]$ 值已经是最短路径的值节点, 而集合 V 则保留其他所有节点, 集合 S 初始状态为空, 而后每一步都有一个节点从 V 移动到 S 。算法的最终结果是: 集合 S 记录了从 s 到 t 的最短路径的节点集, $d[u]$ 为从 s 到 t 的最短路径 (数值)。若节图的点总数为 n 且所有的节点之间都可以连通, 算法遍历图中所有的节点, 很明显的算法时间复杂度是 $O(n^2)$ 。如果图是一个稀疏矩阵, 用一个二叉堆 (binary heap) 或者斐波纳契堆 (Fibonacci Heap) 对 $d[v]$ 进行优先队列排序, 这样寻找最小节点可以提高算法的效率。当用到二叉堆的时候, 算法所需的时间为 $O((m+n)\log n)$, 斐波纳契堆能稍微提高一些性能, 让算法运行时间达到 $O(m + n \log n)$ 。

Dijkstra 算法流程如下图:

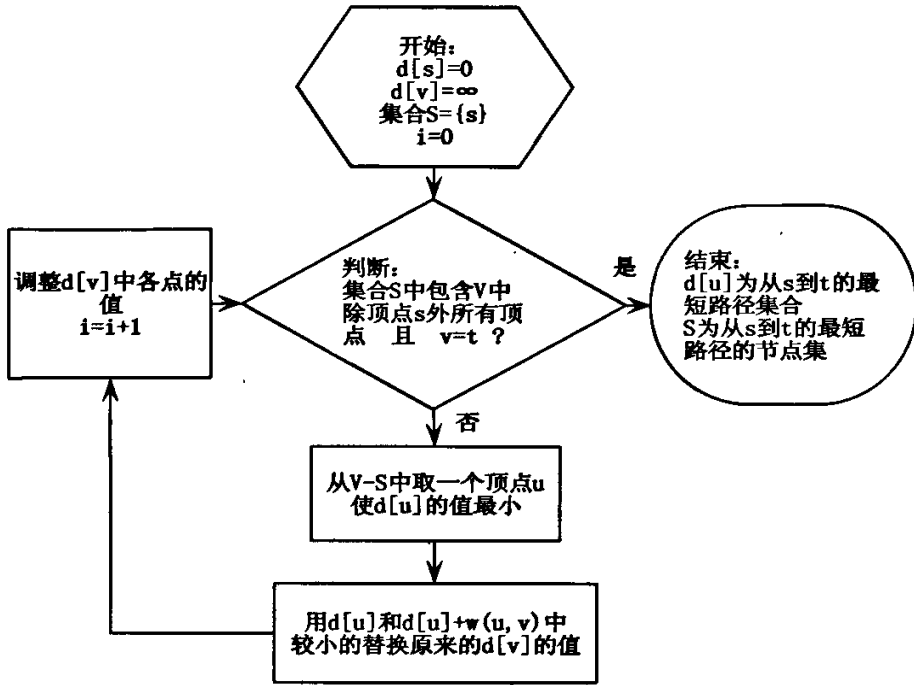


图 5-2 Dijkstra 算法流程图

以邻接矩阵构造的图中 Dijkstra 算法部分代码:

```

void init(GraphMatrix* pgraph, Path dist[])
{
    int i;
    dist[0].length = 0;
    dist[0].prevex = 0;
    dist[0].vertex = pgraph->vexs[0];
    pgraph->arcs[0][0] = 1;
    /* 表示顶点 v0 在集合 U 中 */
    for(i = 1; i < pgraph->n; i++) {
    /* 初始化集合 V-U 中顶点的距离值 */
        dist[i].length=pgraph->arcs[0][i];
        dist[i].vertex=pgraph->vexs[i];
        if (dist[i].length != MAX)
            dist[i].prevex=0;
        else dist[i].prevex= -1;
    }
}
    
```

```

    }

    void dijkstra(GraphMatrix graph, Path dist[]) {
        int i, j, minvex;
        AdjType min;
        init(&graph, dist);
        /* 初始化, 此时集合 U 中只有顶点 v0 */
        for(i = 1; i < graph.n; i++) {
            min=MAX; minvex=0;
            for (j = 1; j < graph.n; j++)
                /*在 V-U 中选出距离值最小顶点*/
                if( graph.arcs[j][j] == 0 && dist[j].length < min )
                {
                    min=dist[j].length; minvex=j;
                }
            if(minvex == 0) break;
        }
        /* 从 v0 没有路径可以通往集合 V-U 中的顶点 */
        graph.arcs[minvex][minvex] = 1;
        /* 集合 V-U 中路径最小的顶点为 minvex */
        for (j = 1; j < graph.n; j++) {
            /* 调整集合 V-U 中的顶点的最短路径 */
            if(graph.arcs[j][j] == 1) continue;

            if(dist[j].length > dist[minvex].length + graph.arcs[minvex][j]) {
                dist[j].length = dist[minvex].length + graph.arcs[minvex][j];
                dist[j].prevex = minvex;
            }
        }
    }
}

```

5.2.3 A*算法

A*算法是在 1968 年发展起来的, 主要利用它来实现人工智能中的路径搜索问题。改进后的 A*算法被应用到了更广泛的领域, Dynamic A*简称 D*就是 A*算法的一种改进算法, 火星漫步者就是在 D*算法的帮助下完成了火星表面的勘探。A*算法的核心是启发式搜索加动态规划, 同样是依靠维护两个表, Open 表和 Close 表来实现的, 它之所以特殊完全是在它的估价函数上。A*算法的估价函数可以表示为: $F = G + H$ 这里的 F 是节点 n 从初始点到目标点的估价函

数, G 是在状态空间中从初始节点到 n 节点的实际代价, H 是从 n 到目标节点最佳路径的估计代价。这里最重要的一个问题就是关于估价值 H 的信息性。 H 的信息性其实就是在估计一个节点的值时的约束条件, 如果信息越多或约束条件越多则排除的节点就越多, 估价函数也就越好或者说这个算法也就越好。但在实际应用中由于实时性的问题, H 的信息越多, 整个算法的计算量就越大, 耗费的时间就越多。这个时候就应该适当的减小 H 的信息, 即减小约束条件, 但这样算法的准确性就差了。这里就有一个平衡的问题, 如何选取适当的 H 值成为 A^* 算法设计的关键。对于一般的几何路网来说, 可以取两节点间欧几里德距离 (直线距离) 做为估价值, 这样估价函数 F 在 G 值一定的情况下, 会或多或少的受估价值 H 的制约, 当节点距目标点近的时候: H 值小, F 值相对就小, 这样就能保证最短路径的搜索始终向终点的方向进行。这样的搜索方式明显的优于 Dijkstra 算法的毫无方向的向四周搜索。同样当 H 值总为 0 的时候, 就相当于没有启发函数, 而 A^* 算法也就退化为 Dijkstra 算法。其实 Dijkstra 算法和 A^* 本质上是相同的, 只有一点不同, 那就是 Dijkstra 算法没有启发函数 F 。由于没有启发函数, 它在各个方向上的搜索是平均, 只有遍历了所有节点后才能得到最优解。

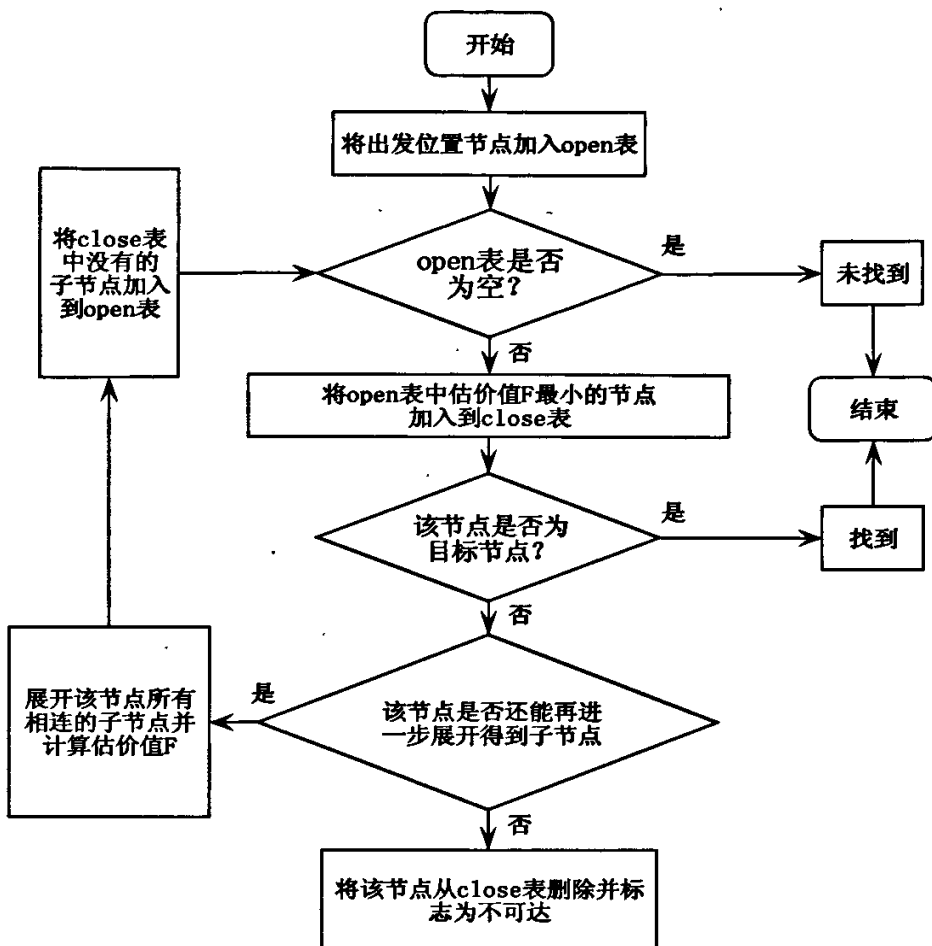


图 5—3 A*算法流程图

如上图所示，算法流程如下：

1. 添加开始节点到 open 表；
2. 重复下面的过程：
 - a) 查找 open 表中 F 值最小的节点，把它放入 close 表。
 - b) 若，和当前节点有连接的每个节点满足下列条件：
 - i. 如果此节点已标志为不可达，或已存在于 close 表，则忽略它。否则执行下面操作。
 - ii. 如果它不在 open 表，就将它添加进去。
 - iii. 对 open 表中所有节点重新计算 G 和 F 值，并按照 F 值从小到大排序。

c) 结束循环，当：

- i. 将目标节点加入到 **open** 表，此时路径已经找到，或
- ii. 没有找到目标节点，并且 **open** 表是空的。此时，没有路径。

3. 从目标节回溯，从每个节点回退到它的父节点，直到抵达开始节点，既得到路径。

就目前的寻路需求来说，优化后的 A* 算法是一种相对较好的解决方案。常用的优化方法有很多种，比较常用的是分别从起点和终点同时相向进行搜索。这样对于静态权值下的单任务寻路来说，理论上可以将搜索时间减半。此外还可以通过维护 **open** 表来进行优化，很明显对 **open** 表里的元素排序将会是整个算法中最耗时的部分之一。引入二叉堆 (binary heap) 以及斐波纳契堆 (Fibonacci Heap) 来动态维持一个有序集，并同时求出最小元素的确是很方便的事情。在大多数解决方案中，使用这类方法会使搜索时间加快至少 2-3 倍，在长路径上将更快 (10 倍以上)。还有一种方法是对地图进行预先规划，找出地图中比较常经过的节点或者是某些必须经过的节点，比如跨江的大桥，隧道，收费站等等。对这些特殊节点的分析可以帮助我们吧地图进行分块，然后对分开的每块小地图求两特殊节点间的最短路径，这样也可以加速整个地图的搜索。在许多游戏中，为了方便 A* 算法实现 AI 的寻路，游戏的设计者会在地图中预先设置些特殊的区域，人为的减小某些节点的代价或加大某些节点的代价。在表现游戏地图中不同的路面情况的同时，避免了过多怪物扎堆的情况。类似这样的设计思路同样也是可以应用到 GIS 地图的设计中去的。

5.3 TSP 问题

5.3.1 问题描述

旅行商问题，即 TSP 问题 (Travelling Salesman Problem) 是数学领域中著名问题之一。假设有一个旅行商人要拜访 n 个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

若要解决该问题一个最容易想到的方法是利用排列组合的方法把所有的路径都计算出来，并逐一比较，选出最小的路径。虽然该方法在理论上是可行的，

但路径的个数与城市的个数成指数增长,当城市个数较大时,该方法的求解时间是难以忍受的,甚至是不可能完成的。以每秒 1 亿次的计算速度来估算,如果 TSP 问题包含 20 个城市时,求解时间长达 350 年;如果要处理 30 个城市,则求解时间更长达 $1+10e16$ 年。如此长的时间,在实际中完成是难以想象的。

5.3.2 模拟退火算法

模拟退火算法是解决 TSP 问题的有效方法之一,其最初的思想由 Metropolis 在 1953 年提出, Kirkpatrick 在 1983 年成功地将其应用在组合最优化问题中。模拟退火算法来源于固体退火原理,将固体加温至充分高,再让其徐徐冷却,加温时,固体内部粒子随温升变为无序状,内能增大,而徐徐冷却时粒子渐趋有序,在每个温度都达到平衡态,最后在常温时达到基态,内能减为最小。用固体退火模拟组合优化问题,将内能 E 模拟为目标函数值 f ,温度 T 演化成控制参数 t ,即得到解组合优化问题的模拟退火算法:由初始解 i 和控制参数初值 t 开始,对当前解重复“产生新解→计算目标函数差→接受或舍弃”的迭代,并逐步衰减 t 值,算法终止时的当前解即为所得近似最优解,这是基于蒙特卡罗迭代求解法的一种启发式随机搜索过程。

求解 TSP 的模拟退火算法模型可描述如下:

解空间:解空间 S 是遍访每个城市恰好一次的所有路径,解可以表示为 $\{w_1, w_2, \dots, w_n\}$, $w_1, \dots, w_n$ 是 $1, 2, \dots, n$ 的一个排列,表明 w_1 城市出发,依次经过 $w_2, \dots, w_n$ 城市,再返回 w_1 城市。初始解可选为 $(1, \dots, n)$;

· 目标函数:目标函数为访问所有城市的路径总长度;我们要求的最优路径为目标函数为最小值时对应的路径。

新路径的产生:随机产生 1 和 n 之间的两相异数 k 和 m ,不妨假设 $k < m$,则将原路径 $(w_1, w_2, \dots, w_k, w_{k+1}, \dots, w_m, w_{m+1}, \dots, w_n)$ 变为新路径: $(w_1, w_2, \dots, w_m, w_{k+1}, \dots, w_k, w_{m+1}, \dots, w_n)$ 上述变换方法就是将 k 和 m 对应的两个城市在路径序列中交换位置,称为 2-opt 映射。

模拟退火算法求解 TSP 问题的流程框图如下:

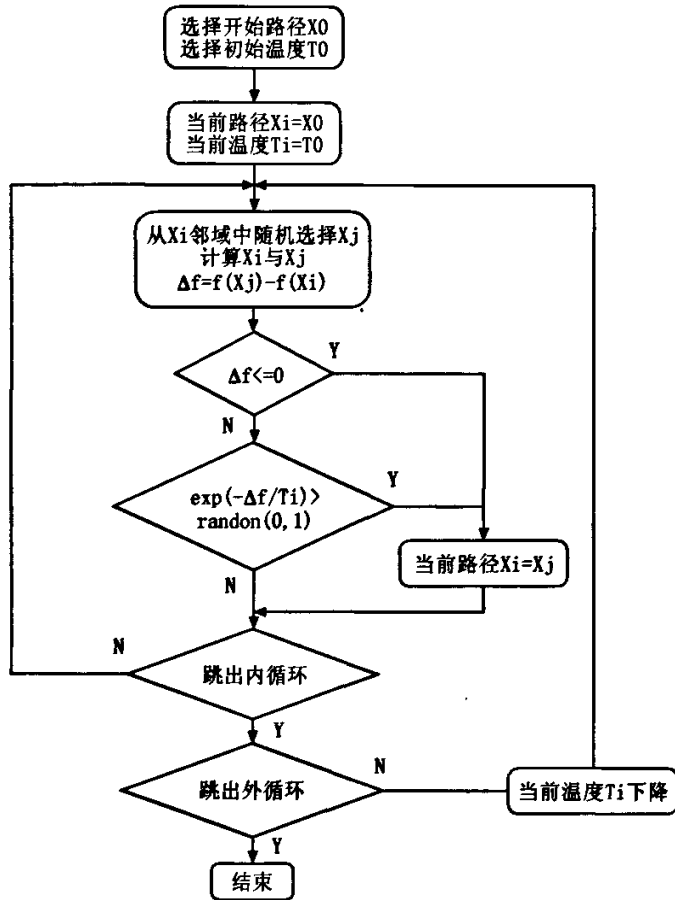


图 5-4 模拟退火算法流程图

对应于流程框图，实现流程的主体函数是 `SAComputation()`，代码如下：

`UINT SAComputation(LPVOID pParam)`

```

{
    while(1)
    {
        double deltatotaldis = 0.0;
        while(1)
        {
            SYRouter SelRouter( ResultRouter.m_CityRouter,
                                NowTemperature,
                                NowExternalIterNumber,

```

```

        NowInnerIterNumber );
//从某路径的邻域中随机选择一个新的路径, 邻域映射为 2-optdeltatotaldis=
SelRouter.m_fTotalDistance-ResultRouter.m_fTotalDistance;
//计算新路径与当前路径的路程长度差值
        if( deltatotaldis <= 0.0 )
            ResultRouter = SelRouter;
//如果新路径的路程短, 则用它替换当前路径
        else
        {
            double chgprobability = exp( -(deltatotaldis/NowTemperature) );
            int randomnum = rand();
            double random = ((double)(randomnum%10000))/10000.0;
            if(chgprobability > random )
                ResultRouter = SelRouter;
//如果新路径长于当前路径, 但  $\exp(-\Delta f/t) > \text{random}(0,1)$ , 则仍然替换当前路径
        }
        if( JudgeOverInnerLoop(0) )
            break;
//判断内循环是否结束, 结束则跳出当前温度的内循环
        else
            NowInnerIterNumber++;
//判断内循环是否结束, 不结束则继续内循环
    }
    if( JudgeOverExternalLoop(0) )
        break;
//判断外循环是否结束, 结束则结束模拟退火计算
    else
    {
        NowTemperature = CountDownTemperature( NowTemperature, 0 );
        NowExternalIterNumber++;
        NowInnerIterNumber = 0;
    }

```

//判断外循环是否结束，不结束则计算出下降后的温度，并继续循环

}

}

}

第 6 章 总结与展望

随着网络技术的不断发展和物流仓储中各种不同设备的陆续引进,使得物流管理必须处理越来越多的信息。此时,设计一个功能强大,能与物流中其他子系统有效地进行信息交互的物流仓储管理系统势在必行,而选择怎样的开发平台来设计这个系统自然成为首先要考虑的问题。

通过使用 J2EE 开发平台,大大简化了应用程序的编程代码量,部分地免去了编程的复杂性,提高了软件的生产效率和可维护性。因此,通过使用 J2EE 架构,可以快速地开发出高质量的企业级应用软件。MVC 本身就是一个非常复杂的系统,所以采用 MVC 实现 Web 应用时,最好选一个现成的 MVC 框架,在此之下进行开发,从而取得事半功倍的效果。现在有很多可供使用的 MVC 框架,由于 Struts 有完整的文档并且相对来讲比较简单,所以用它开发 MVC 系统还是非常的方便。通过将 Rational 统一过程方法引入到对物流仓储管理系统的需求分析,将 Struts 开源框架应用于项目的 Web 开发,并在此基础上形成了一个具体的设计方案:Web 层,以一个 Servlet 作为前端控制器,用户的集中访问点和处理用户请求的集中控制点;业务逻辑层,以一个外观会话 bean 封装一个实体 bean,实现了核心业务和数据库访问逻辑;各层组件通过统一接口进行交互,软件结构清晰,层次分明,克服了过去软件开发中所暴露的问题;数据持久化层,将访问数据库的操作全部放在这一层,业务逻辑层通过这层来访问后台的数据库,不仅提高了访问效率,便于根据需要修改数据访问方式,同时有效将逻辑与数据分隔,增强了数据的安全性。

通过优化算法来提高物流管理决策的质量是一种简单而高效的手段。和基于 Web 的 GIS 的进一步结合,可以在物流上实现货物跟踪、路径规划等颇具经济价值的功能。随着计算机信心技术的发展,新的架构新、更优化的算法可以给我们带来更优质的系统;GPS、手机移动通讯等新技术的普及和价格的回归,在硬件层面上给物流仓储系统提供了更高的平台,同时也提出了更高的要求。未来的管理信息系统必将是建立在基于多种通讯协议上的、跨平台的实时系统。

参考文献

- [1] 蒋宝家; 郑荣; 董绍华。可视化仓库管理系统的设计。物流技术, 2005 年 09 期, 178-183。
- [2] 徐常凯; 李继军; 郭志明。基于 Web 的网络数据库技术在物流信息系统中的应用。物流科技, 2002 年 05 期, 24-27。
- [3] 张开碧; 刘焕淋。基于现代物流管理的自动化立体仓库系统开发。山东科技大学学报(自然科学版), 2004 年 03 期, 124-127。
- [4] 李云; 高小强; 郑忠; 何腊梅; 赵克文; 宋国菊; 张大德; 陈小平。运用 ActiveX 技术开发炼钢物流可视化仿真平台。计算机应用研究, 2005 年 05 期, 181-197 转 707。
- [5] 彭洁; 荣鹏辉等。虚拟技术在库存管理中的可视化研究。物流技术, 2004 年 12 期, 24-25。
- [6] 张志刚; 曹西京; 刘昌祺; 王栋生。自动化立体仓库系统仿真的研究。计算机仿真, 2005 年 07 期, 115-117。
- [7] 曹剑东; 郑四发; 余贵珍; 李兵; 耿华; 杨扬; 连小珉。基于 Java 的物流信息系统开发及应用。计算机工程与应用, 2005 年 30 期, 178-181。
- [8] 周亮; 王铁宁; 李胜利; 裴帅。基于 Agent 的物流信息平台模型研究。物流科技, 2005 年 09 期, 51-54。
- [9] 徐丽; 曹元大; 廖乐建; 胡晶晶。基于 Multi_agent 的物流管理系统规划求解。微机发展, 2005 年 09 期, 35-38。
- [10] 林玲; 吴清江。基于电子商务的物流信息管理系统的架构。福建电脑, 2005 年 11 期, 128-129。
- [11] 孟志青; 段智华。商业销售分析数据仓库的数据可视化的一种设计。计算机工程与应用, 2000 年 10 期, 120-168。
- [12] 张成; 袁冰; 王杰; 武超; 胡侨丹。条码技术在立体仓库物料管理中的应用。中国制造业信息化, 2003 年 10 期, 76-78。
- [13] 张杰; 关永; 孙继平。工控组态软件的可视化。中国图象图形学报, 2002 年 02 期, 201-204。
- [14] 陈爱军; 朱绍文; 何克忠。工控可视化用户工具的设计与实现。计算机应用与软件, 2000 年 08 期, 16-20 (转 61)。
- [15] 邓文华。基于 SQLSERVER2000 的数据挖掘方案研究, 物流科技。2003 年 02 期, 54-58。

- [16]蒋代梅; 刘洋; 周小兵。基于 GPS/GIS 的物流运输管理系统的实现技术。北京工业大学学报, 2005 年 04 期, 443-448。
- [17]李一波; 张志勇。基于 GPS 和 GIS 的可视物流管理信息系统设计。冶金自动化, 2005 年 05 期, 11-20。
- [18]刘霜梅; 柳永坡; 刘雪梅; 曹德桥。基于 GIS 的 B2C 物配送管理系统设计。电子商务世界, 2004 年 03 期, 54-55。
- [19]刘志强; 周枫。基于 3G 的物流配送系统设计与实现。计算机工程与应用, 2002 年 18 期, 249-256。
- [20]陈昉。电子商务环境下的物流解决方案——基于 Internet 上的远程物流管理信息系统。中国远洋航务公告, 2000 年 05 期, 15-19。
- [21]付万; 王月; 朱元祥等。基于 GIS 的现代物流管理系统开发及其应用——以物资配送为例。物流科技, 2006 年 01 期, 24-27。
- [22]苏一丹; 李桂。电子商务物流管理信息系统中最优佳径算法的研究。计算机工程与应用, 2002 年 18 期, 182-225。
- [23]李宁宁; 刘玉树。改进的 Dijkstra 算法在 GIS 路径规划中的应用。计算机与现代化, 2004 年 09 期, 13-17。
- [24]毕军; 付梦印; 周培德; 张宇河。基于城市道路网的快速路径寻优算法。计算机工程, 2002 年 12 期, 36-38。
- [25]毕军; 付梦印; 周培德; 张宇河。基于城市道路网的最短路径分析解决方案。小型微型计算机系统, 2003 年 07 期, 1390-1393。
- [26]Michel, Roberto. Extended ERP services, *Manufacturing Systems*, v 17, n 10, Oct, 1999, 7 pp.
- [27]Basu, Vedabrata. An agency theory model of ERP implementation, *Proceedings of the ACM SIGMIS CPR Conference, Proceedings of the 2004 ACM SIGMIS CPR Conference, SIGMIS CPR 2004*, 2004, p 8-13.
- [28]Bowman, Brent . THREE STAGE MODEL OF MIS PLANNING, *Information & Management*, v 6, n 1, Feb, 1983, p 11-25.
- [29]Hoffman, K. Douglas. Marketing + MIS = E-service. *Communications of the ACM*, v 46, n 6, June, 2003, p 53-55.
- [30]Di Lucca, Giuseppe Antonio. Recovering a Business Object Model from Web Applications, *Proceedings - IEEE Computer Society's International Computer Software and Applications Conference*, 2003, p 348-353.
- [31]Kaihara, T. Virtual market emergence for resource allocation in supply web, *IECON Proceedings (Industrial Electronics Conference)*, v 4, 2000, p 2678-2683.

- [32] Arcelli, Francesca. Agent based internet infrastructure for learning commerce. (HICSS-33), Jan 4-Jan 7 2000, Maui, USA .
- [33] Conrad, Tsahat Onesime Oboulhas. Web-based supplier relationship framework using agent systems. Journal of Harbin Institute of Technology (New Series), v 11, n 1, February, 2004, p 54-59.
- [34] Huang, Ming-Hui. Information load: its relationship to online exploratory and shopping behavior. International Journal of Information Management, v 20, n 5, Oct, 2000, p 337-347.
- [35] Shaw, Michael J. Information-based manufacturing with the web. International Journal of Flexible Manufacturing Systems, v 12, n 2, Apr, 2000, p 115-129.
- [36] Khalid, Halimahtun M. Human factors of IT-based solutions for Worldwide Manufacturing Web. Human Factors and Ergonomics In Manufacturing, v 10, n 1, Winter, 2000, p 99-113.
- [37] Ni, Kai. Implementation of shortest path algorithm based on database middle-ware and GIS. Jisuanji Gongcheng/Computer Engineering, v 31, n 13, Jul 5, 2005, p 78-80. .
- [38] Djokic, Dean. Application of GIS network routines for water flow and transport. Planning and Management, v 119, n 2, Mar-Apr, 1993, p 229-245.
- [39] Zhao, Qing-Qi. Shortest-path algorithm research based on GIS for distribution system maintenance , Proceedings of the 2004 Chinese Control and Decision Conference (16th CDC), Proceedings of the 2004 Chinese Control and Decision Conference (16th CDC), 2004, p 346-348+352 .
- [40] Sung, Kiseok. Shortest paths in a network with time-dependent flow speeds. European Journal of Operational Research, v 121, n 1, Feb, 2000, p 32-39.

致 谢

光阴似箭，转眼间两年半的硕士学习生涯即将结束。两年多的学习生活当中，导师夏红霞教授，钟珞教授都给了我无微不至的关怀。他们的严谨的治学态度、求实的工作作风，都使我受益非浅。特别给我树立正确的研究态度，悉心指导我走好每一步。在此，我谨向恩师夏红霞教授，钟珞教授致以最诚挚的感谢和最崇高的敬意！深深地感谢您们为我做出的一切！

在研究生学习期间，邹承明副教授，袁景凌副教授，赵广辉博士在学习和工作中都给了我大量的指导和帮助，在此我向他们表示深深的谢意。

感谢课题组成员刘鹏、黄诗诗同学，以及二年级、一年级的师弟师妹们，正是他们的帮助才使得课题研究顺利进行。本论文的完成离不开他们的帮助。

最后，我要特别感谢养育我多年的父母，我有今天离不开他们的含辛茹苦！

2006 年 5 月

附录：攻读硕士学位期间发表的学术论文

GIS 中最短路径问题的应用研究 (Research on Shortest Path Problem in GIS), 软件导刊, 2007.6.

基于Web的物流仓储管理系统的设计

作者: [俞奕](#)
学位授予单位: [武汉理工大学](#)
被引用次数: 1次

本文读者也读过(3条)

1. [毛向荣](#) [基于ASP.NET和Web服务的物流仓储管理系统的设计与实现](#)[学位论文]2008
2. [王开祥](#) [基于C#的物流仓储系统的设计与实现](#)[学位论文]2009
3. [任振江](#) [潍坊市邮政物流管理系统的设计与实现](#)[学位论文]2009

引证文献(1条)

1. [苏杭](#), [曾盛绰](#) [B/S模式下多层架构的仓储管理信息系统设计](#)[期刊论文]-[物流工程与管理](#) 2009(3)

引用本文格式: [俞奕](#) [基于Web的物流仓储管理系统的设计](#)[学位论文]硕士 2007