

面向物流的多源异构数据的整合与互操作技术研究综述文献一：

面向物流的多源异构数据的 整合与互操作技术 研 究 综 述

硕士研究生：陈 永 东

学号：1020379104

班级：Z0203792

专业：软件工程硕士

导师：王东、宋文官

上 海 交 通 大 学 软 件 学 院

2003 年 4 月

面向物流的多源异构数据的整合与互操作技术研究综述

(陈永东)

摘要: 本研究报告对国内外有关面向物流的多源异构数据的整合与互操作技术的研究进行了综述,重点分析了各种面向物流的多源异构数据的整合与互操作技术的设计思想,并阐述了自己对面向物流的多源异构数据的整合与互操作技术的有关观点。为课题“面向物流的多源异构数据的整合与互操作技术的研究”的选题,和开展进一步的研究工作打下了一个良好的基础。

其内容包括:面向物流的多源异构数据的整合与互操作技术研究的背景与意义;面向物流的多源异构数据的整合与互操作的应达到的要求;面向物流的多源异构数据的整合与互操作技术研究综述;本人对面向物流的多源异构数据的整合与互操作的观点。

关键字: 物流; 异构; 数据; 整合; 研究综述

目录:

一、面向物流的多源异构数据的整合与互操作技术研究的背景与意义

- 1、面向物流的多源异构数据的整合与互操作技术研究的背景
- 2、面向物流的多源异构数据的整合与互操作技术研究的意义

二、面向物流的多源异构数据的整合与互操作的应达到的要求

- 1、物流行业数据的多源异构特性
- 2、面向物流的多源异构数据的整合与互操作应达到的要求

三、面向物流的多源异构数据的整合与互操作技术研究综述

- 1、目前常见的多源异构数据的整合与互操作技术
- 2、多源异构数据的整合与互操作技术分析比较
- 3、面向物流的多源异构数据的整合与互操作技术的特殊性

四、本人对面向物流的多源异构数据的整合与互操作的观点

- 1、面向物流的多源异构数据的整合与互操作的研究具有重要的战略意义和实际应用价值
- 2、主系统与子系统的两个层面必须相互协作
- 3、面向物流的多源异构数据的整合与互操作必须做好三种接口设计
- 4、面向物流的多源异构数据的整合的主要三个方面
- 5、先进合理的设计原则和未来技术的开放性是成功实现面向物流的多源异构数据的整合与互操作的关键

一、面向物流的多源异构数据的整合与互操作技术研究的背景与意义

1、面向物流的多源异构数据的整合与互操作技术研究的背景

面向物流的多源异构数据的整合与互操作技术研究是上海市科委现代物流信息系统关键技术集成与应用示范项目(2003-2005 年)的一个子项目研究。

该项目期望通过在战略、技术、标准、示范等方面的工作,构筑物流公共信息平台,并通过应用示范工程以点带面,促进上海现代物流产业的发展。其主要包括以下主要项目:

(1) 基于智能标签的现代物流自动识别技术与应用示范

研究以高频和超高频智能标签为核心的 RFID 射频识别标准、智能标签 RFID 射频通讯和读写器具标准、基于智能标签的集装箱自动识别技术、基于智能标签的单元物品自动识别技术,引导物流信息向全过程管理迈进。要求通过二年的努力,获得符合 ISO15693 规定的高频和超高频智能标签后封装产品和智能标签读写器具,提出有关的标准草案、规范或指导性文件,申请专利或软件著作权,并进行应用示范。

(2) 跨平台物流空间信息服务系统关键技术与应用示范

基于空间信息及数据可视化处理技术,研究开发融合高精度位置信息、物流数据信息、车辆定位信息和动态信息发布的 GGMN (GIS/GPS/MIS/NET) 系统,使得在区域物流信息系统中,让用户能通过多种应用系统及各类应用终端,以 WEB、WIRELESS 等方式获得全面的信息搜索、查询及服务,快速地形成物流解决方案,便捷地进行实时物流数据交换和发布。要求通过二年的努力,重点突破物流地理信息数据加工、转换及集成应用关键技术;跨平台物流动态信息发布系统关键技术;智能导航终端嵌入式软件开发;基于物流特征的动态导航终端关键技术与系统;区域主要物流特征信息模型及城市物流配送分析软件开发;物流系统实时监控技术以及开展跨平台物流空间信息服务系统应用示范。要求所研发技术能够申请软件著作权或专利,能够形成终端产品并进行应用示范。

(3) 企业集团物流信息系统关键技术开发与应用示范

围绕企业集团对物流配送网络优化、订单处理、作业计划、运力资源控制等的要求,集成物流管理、优化、仿真、控制和决策支持等技术,开发面向企业供应链全过程的物流信息管理软件系统,全面支持多客户环境下的第三方物流信息管理业务。要求通过二年的努力,重点解决物流配送网络优化技术、物流作业计划与路径优化技术、动态信息管理技术、物流决策支持技术、物流过程质量控制技术和物流系统绩效评估技术。要求所研发技术能够申请软件著作权或专利,能够结合实际案例进行应用。

(4) 现代物流信息平台关键技术与应用示范

围绕上海国际航运中心建设目标,研究制订现代物流信息技术与产业化发展战略,以物流信息标准化技术为支撑,以物流信息平台技术为基础,通过应用示范工程实现物流信息技术的应用集成,构筑上海城市现代物流信息平台发展框架,推动上海现代物流产业的发展。要求通过一年的努力,提出现代物流信息技术与产业化发展战略、物流信息编码技术标准和信息接口技术标准,研发物流信息共享与数据元联结格式平台技术;要求通过二年的努力,依托物流园区开展陆路运输物流信息平台示范、口岸(海运和空运)物流信息平台示范、公共平台与企业平台在多式联运中的整合示范和物流信息技术标准化示范基地建设。要求形成物流信息技术与产业化发展战略报告、物流信息标准化指南、有关的标准草案和规范或指导性文件,要求所发展的技术能够申请软件著作权或专利,要求完成有显

示度的应用示范。

面向物流的多源异构数据的整合与互操作技术研究是以上总课题中的一个的技术研究部分。

2、面向物流的多源异构数据的整合与互操作技术研究的意义

面向物流的多源异构数据的整合与互操作技术研究将为上海市实现现代物流信息技术与产业化发展战略，以物流信息标准化技术为支撑，以物流信息平台技术为基础，通过应用示范工程实现物流信息技术的应用集成，构筑上海城市现代物流信息平台发展框架，推动上海现代物流产业的发展提供宏观技术分析与支持，从而中国物流信息化的进程。

二、面向物流的多源异构数据的整合与互操作的应达到的要求

1、物流行业数据的多源异构特性

在物流信息化建设过程中，由于各业务系统建设和实施数据管理系统的阶段性、技术性以及其它经济和人为因素等因素影响，导致企业在发展过程中积累了大量采用不同存储方式的业务数据，包括采用的数据管理系统也大不相同，从简单的文件数据库到复杂的网络数据库，它们构成了企业的异构数据源。这些分散的不同业务的数据管理系统虽然能够满足业务数据存储和管理要求，但在许多情况下，为作出一个决策，可能需要访问分布在网络不同位置上的多个业务数据管理系统中的数据。因此，基于整个、集成各个业务异构数据源的综合信息仓库的一个强大的分布式应用系统十分需要。

异构数据源的整合、集成是物流信息化建设过程经常遇到的一个现实问题。也是制约各种应用信息系统建设和数据共享程度，以及信息化建设投资重复或负担重的一个重要因素。

物流行业数据的多源异构特性主要表现在三方面：

(1) 系统异构，即数据源所依赖的业务应用系统、数据库管理系统乃至操作系统之间的不同构成了系统异构。

(2) 模式异构，即数据源在存储模式上的不同。存储模式主要包括关系模式、对象模式、对象关系模式和文档嵌套模式等几种，其中关系模式（关系数据库）为主流存储模式。同时，即便是同一类存储模式，它们的模式结构可能也存在着差异。例如不同的关系数据管理系统的数据类型等方面并不是完全一致的，如 DB2、Oracle、Sybase、Informix、SQL Server、FoxPro 等。

(3) 来源异构，即企业内部数据源和外部数据源之间的异构。

同时，在面向物流的多源异构数据的整合与互操作在实现中，应考虑到技术实现时涉及以下几类行业及机构：

- (1) 陆路运输行业
- (2) 口岸运输行业（海运与空运）
- (3) 主要物流园区
- (4) 物流供应链中的各类参与企业
- (5) 社会机构（银行、海关等）

2、面向物流的多源异构数据的整合与互操作应达到的要求

根据系统要求，信息系统数据交换平台采用多层分布式结构建立系统。系统分为三层：

原始数据层：为原业务系统数据库。数据交换系统不对原系统做任何变动。

数据映射层：在交换平台上通过映射为所有信息系统建立一个统一的、位置透

明的信息系统数据接口。映射方式具有充分的灵活性，可以是原始数据对象的超集或子集。通过应用程序，可以按横向、纵向方式对数据进行扩展或集中。通过重命名机制解决冲突问题。

交换平台层：为一个新的分布式数据库系统，其数据包括独立于各业务系统的新的应用的数据结构、各业务数据库的集成。通过交换平台，为所有的业务系统提供一个统一的入口。面向物流的多源异构数据源的数据整合和互操作的目的是为物流业综合应用系统提供集成的、统一的、安全的、快捷的信息查询、数据挖掘和决策支持服务。为了满足这个需求条件，整合、集成后的数据必须保证一定的集成性、完整性、一致性和访问安全性。

(1) 集成性

物流行业各种原先孤立的业务信息系统数据经过整合、集成后，应该达到查询一个综合信息不必再到各个业务系统进行分别查询和人工处理，只要在整合、集成后的数据信息仓库中就可以直接访问到，即整合、集成后的综合信息仓库的数据是各异构业务数据的有机集成和关联存储（整合、发掘出各业务数据间的内在关联关系），而不是简单、孤立的堆放在一个数据库系统里。

(2) 完整性

包括数据完整性和约束完整性两方面。数据完整性是指完整提取数据本身，约束完整性，约束是指数据与数据之间的关联关系，是唯一表征数据间逻辑的特征。保证约束的完整性是良好的数据发布和交换的前提，可以方便数据处理过程，提高效率。

(3) 一致性

不同业务信息资源之间存在着语义上的区别。这些语义上的不同会引起各种不完整甚至错误信息的产生，从简单的名字语义冲突（不同的名字代表相同的概念），到复杂的结构语义冲突（不同的模型表达同样的信息）。语义冲突会带来数据集成结果的冗余，干扰数据处理、发布和交换。

整合、集成后的数据应该根据一定的数据转换模式和商业规则进行统一数据结构和字段语义编码转换。

(4) 访问安全性

由于数据库资源可能归属不同的单位，各业务数据系统有着各自的用户权限管理模式，访问和安全管理很不方便，不能集中、统一管理，所以保证在访问异构数据源数据基础上保障原有数据库的权限不被侵犯，实现对原有数据源访问权限的隔离和控制，就需要设计基于整合、集成后的综合信息仓库的统一的用户安全管理模式来解决此问题。

系统的实现分为数据级、系统级、应用级。系统的实现原理如图 3 所示。

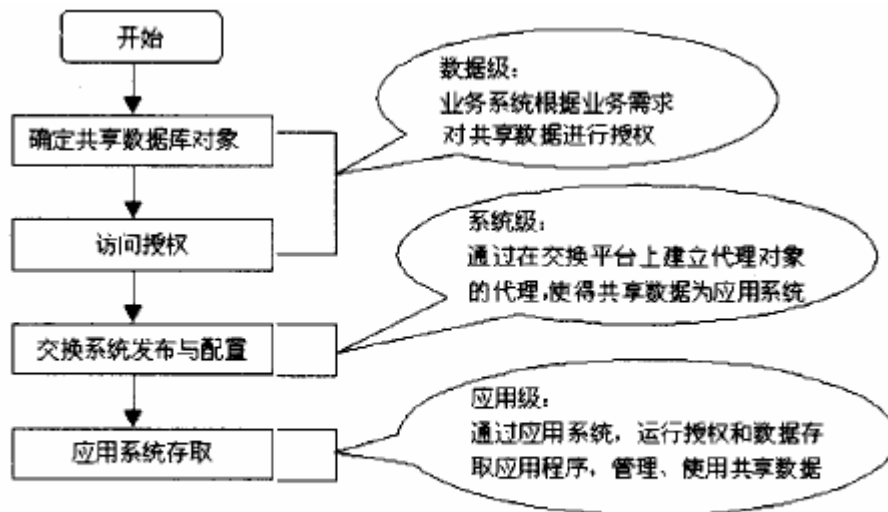


图 3 系统处理流程

三、面向物流的多源异构数据的整合与互操作技术研究综述

1、目前常见的多源异构数据的整合与互操作技术

(1)中间件技术应用现状

中间件的分类：中间件的产品种类很多，根据中间件在系统中所起的作用和采用的技术不同，大致划分为以下几种：

A、消息（通信处理）中间件（Message-Oriented Middleware）

通过一定格式将数据从一个应用程序转发到另一个应用程序，能在不同平台之间通信，实现分布式系统中可靠的、高效的、实时的跨平台数据传输，并能支持多种平台及多种开发工具，称为消息中间件。中间件一般用在非实时的数据交换上，消息中间件就非常适合这种跨平台的异构环境。消息中间件是中间件中唯一不可缺少的，也是销售额最大的中间件产品，目前在 Windows 2000 操作系统中已包含了其部分功能。最典型的消息中间件产品如：IBM 的 MQSeries、Microsoft 的 MSMQ、Talarian 的 SmartSockets 等。

B、交易（事务处理）中间件 Transaction Processing Monitors (TP)

交易中间件具有跨平台、跨网络、工作于多种异构环境下的特点，负责维护事务的完整性、安全性。TP 中间件一般工作于高负载的环境，例如银行，它能非常精确保证多个应用程序同时操作数据时的完整性和一致性。典型产品主要有：CICS, OpenCICS (IBM)、BEA Tuxedo 等。BEA 的 Tuxedo 很著名，它成为增长率最高的厂商。

C、对象中间件（Object Monitors）

对象中间件具有 TP 的所有功能，但是按照面向对象的模式来组织体系结构。在线的电子交易很适合应用这种中间件类型。现在的对象中间件都按照下面三种体系结构中的某一种来构造。这三种体系结构是：Object Management Group 的 CORBA, Sun 的 EJB, 微软的 COM+。

D、应用服务器 (Application server)

应用服务器是基于标准 EJB 体系的中间件容器，它能改善程序的可靠性，降低开发风险及成本。应用服务器最出名的是 BEA 的 WebLogic, IBM 的 WebSphere 和 Borland 的 Appserver。

中间层服务器软件(即数据服务中间件)的设计与实现,其作用是实现分布式应用的数据访问和事务处理,简化多层应用的开发代价。它主要实现以下功能:(1)数据的位置透明性,即数据服务器的物理位置对其上层应用(一般指业务逻辑层)提供位置透明性,使应用逻辑层只需知道数据服务中间件服务器(M i d d l e w a r e S e r v e r)即可,无需知道数据源的具体位置,从而实现了分布式的数据访问。(2)分布式数据查询,由数据服务中间件将企业的应用逻辑中对数据的查询请求转化为数据库查询请求,然后从数据库服务器获得数据并提供给应用层,为企业应用提供统一的数据访问接口。(3)事务处理,在多个应用对数据“同时”进行读写操作时,保证数据的一致性、完整性。(4)异构数据源兼容,能够将企业的各种不同的数据管理系统甚至是数据文件连接起来,为应用逻辑层提供统一的数据访问接口。(5)安全认证,能对系统合法用户进行数据访问认证,增加数据的安全性。

E、安全中间件

一些军事、政府和商务部门上网的最大障碍是安全保密问题,而且不能使用国外提供的安全措施(如防火墙、加密、认证等),必需用国产的产品。产生不安全因素是由操作系统引起的,但必需用中间件去解决,以适应灵活多变的要求。

F、跨平台和构架的中间件

当前开发大型应用软件通常采用基于构架和构件技术,在分布系统中,还需要集成各节点上的不同系统平台上的构件或新老版本的构件,由此产生了构架中间件,功能最强的是 CORBA,可以跨任意平台,但是太庞大;JavaBeans 较灵活简单,很适合于做浏览器,但运行效率差;DCOM 模型主要适合 Windows 平台,已广泛使用。实际上国内新建系统主要是 UNIX(包括 LINUX)和 Windows,因此针对这两个平台建立相应的中间件要实用得多。

(2) CAX——一种有效的数据集成方法

CAX:是三种最新的数据集成方法 CORBA, Agent, XML 的缩写,该方法在基于 CORBA 体系结构的基础上,采用代理技术的机制运作系统,同时在数据语义层采用基于 XML 的数据模型作为公共数据模型实现异构数据的模式集成.这种方法在我们研制的原型系统 Panorama 中被初步证明是有效的和实用的.

A、CORBA 技术的运用

CORBA 优点: 由于在集成性、可用性、透明性和可扩展性等方面具有较大优势,CORBA 技术又显得更胜一筹.

CORBA 缺点: CORBA 对象是静态的,仍将分布式应用中的自主行为实体简单地划分为“客户”和“服务器”两类;客户与服务器之间的交互方式仅限于传统的客户主动请求、服务器被动响应的非对等关系.同时,CORBA 也无法解决局部数据管理系统的异构性,例如不同的 DBMS 在访问控制、约束、查询语言、事务管理以及安全策略等方面都会有很大不同,而且有的局部系统如一些文件系统可能就不具备数据管理能力.

B、代理(Agent)技术的运用: 建立一种有效的运行机制以解决上述问题.一个代理至少有以下各项特性和能力:

- 代理性:代表创建它的用户执行计算;
- 自治性:没有监督独立地定位、执行和移动;
- 智能性:通过它获得的信息学习并且做出决定;
- 主动性:能够遵循承诺采取主动行动,表现出面向目标的行为.
- 递归性:如果需要,可以为子任务创建子代理

表 1 代理技术与其它分布计算技术的比较

Table 1 Comparison of agent and other distributed computing technologies

	交互方式			角色与关系	
	客户	服务器	交互方式	客户/服务器	关系
传统 c/s	客户主动请求	服务器被动响应	不对等	客户/服务器	固定关系
分布对象	客户主动请求	服务器被动响应	不对等	组件	对等关系
Agent	客户主动请求	服务器主动服务	对等	Agent	对等关系

在我们的基于 Agent 的运行机制中,主要包括两种代理:局部代理和全局代理。

一个局部代理是一种静态的用户代理接口,与局部系统相联系,负责维护局部系统的输出模式,完成实际查询操作,同时也是全局事务在局部系统中的事务代理以及信息交换过程中的安全代理。

这个全局代理可以创建多个可移动的子代理,这个移动子代理是个对象,它包括它的状态、它的任务的一些执行结果以及自管理和自派遣的智能行为。这些移动子代理在局部系统主机上请求并获得服务,因而网络通信量、持续的网络连接都能减少。同时,这些移动子代理以局部系统用户的身份与局部代理配合,完成查询处理、事务处理等工作,更充分满足局部系统的自治性。这种基于代理技术的运行机制具有最大限度的灵活性和适应性,能适应于网络分布应用的全部多样性,更易于实现各种异构数据源的在线访问。

C、XML 技术的运用

上述措施并不能解决被集成的各成员系统的数据在语义层的异构问题。解决这个问题最常用的方法是提供一个公共数据模型(CDM, Common Data Model),每个成员模型被映射到 CDM 上。在传统 MDBS 中,往往采用面向对象模型作为 CDM,但许多结构化和半结构化数据由于没有显式的结构,元数据与数据混合存放,一种能集成这类数据的 CDM 必须能起到某种元数据的作用。

“...XML,描述一个存储在计算机中的数据对象的类以及处理这些对象的程序的行为...”。更重要的是,XML 允许用户自己定义新的可嵌套的标签满足应用需要,具有强大的数据描述能力,因此 XML 能适应任何类型的数据,这样,结构化、半结构化或非结构化的数据都使用 XML 作为扩展的格式来自由交互。XML 源文件是文本文件,因而既是最简单的数据格式,又容易实现平台无关性。

因此,有人提出一种基于 XML 的集成数据模型—XIDM(XML-based Integrating Data Model)作为扩展的 MDBS 公共数据模型,以便实现成员系统模式与全局系统模式之间的映射,解决异构数据表示和关联的同一性问题。虽然 XML 显式的文档结构是树形的,但利用 XML 提供的几种属性类型,XIDM 可以表示图结构,从而能够给没有规则模式的无结构化和半结构化数据建立一种模式,方便 MDBS 中的模式集成。XIDM 建立在 XML Schema 基础上,能表达较丰富的数据类型。同时,由于 XML Schema 提供了键码、单值、域和一般性等约束的定义,因此能方便地实现 XIDM 与关系、对象等其它数据模型之间的转换。在 Panorama 系统采用了四级模式结构:

- 局部模式
- 输出模式
- 全局模式
- 用户外模式

- 局部模式.局部模式由局部系统的数据模型来表示,不同的局部模式有不同的数据模型,如对象模型、关系模型等,对于缺乏模式结构的文件系统,可添加一个基于XIDM的模式.

- 输出模式.通过将局部模式翻译成XIDM模型得到输出模式,这个翻译过程将在局部模式类与输出模式类之间建立一种映射关系.

- 全局模式.集成多个输出模式以产生全局模式,它会产生有关数据分布的映射信息.通过查询管理器将全局模式中的查询传输到相应的输出模式.全局模式建立在XIDM基础上.

- 用户外模式.外模式是为一个用户或一个应用定义的模式,主要用来存放那些不能由输出模式导出的附加信息,包括与用户紧密相关的全局模式的信息子集和一些附加的一致性约束.

(3) 多源数据转换与装载向导开发技术

A、. 功能与结构

多源数据转换与装载向导DEPWizard包括四部分功能:多源数据连接、异构数据类型转换与装载、页面元数据生成、数据页面预览,如图1所示。

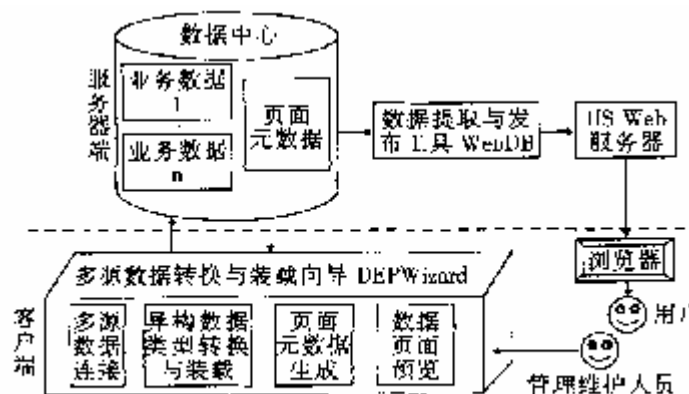


图1 多源数据转换与装载向导功能结构

多源数据转换与装载向导功能结构多源数据连接是指实现与FoxProDBF文件、AccessMDB文件中的表、ExcelXLS文件中的表单,以及各种通过ODBC标准连接的异构数据源(如Oracle,Sybase,SQLServer等)的连接。

异构数据类型转换与装载实现从已建立连接的数据源中读取数据、转换数据类型,并建立与数据中心的连接,实现数据装载。

页面元数据生成包括采集关于数据发布的页面描述信息,生成页面元数据。页面元数据存放了对所发布数据信息的分类、页面的名称及描述,以及从数据中心抽取数据的SQL查询语句描述。页面元数据支持两级数据钻取功能,即一级页面统计查询数据,二级页面关于统计结果的细粒度详细数据。

数据页面预览实现向服务器发送URL,请求浏览数据页面,服务器根据URL调用WebDB生成数据发布页面发送到浏览器。

DEPWizard采用向导方式完成上述所有功能,管理维护人员所要做的全部事情就是按DEPWizard每一页上提供的详细说明输入或选择必要信息,然后按“下一步”按钮。如果想修改前面某一页输入或选择的信息,只需按“上一步”按钮,返回该步进行修改。

B、对象设计

向导管理对象

数据库连接与操作对象

数据类型转换与装载对象

数据页面元数据生成对象

数据页面预览对象

C、主要实现技术

向导实现技术：向导的功能是通过向导管理对象和向导管理对象中管理的属性页对象共同实现的。

数据类型转换实现技术：均通过 D E P W i z a r d 的数据转换对象 T Y P E T r a n s O B J

静态文本超级连接功能扩充技术：我们知道 V C ++对话框上的静态文本控件不支持超级连接功能,且不能接收鼠标事件,但 V C ++支持 U R L 导航,因此,我们对静态文本控件功能进行了扩充: (1)扩充静态文本控件接收鼠标事件功能。(2)改变静态文本控件的外观显示

(4) 基于 COM+异构数据库访问实现技术

目前的多数据库实现方法有多种,基本方式有联邦数据库、网关方式、中间件技术.这 3 种实现方法各有利弊,但中间件技术目前能够比较好地结合 COM+或 CORBA 技术,屏蔽不同平台,不同数据库管理系统的差异.因此,本文采用了基于 COM+的中间件技术来实现异构多数据库互操作系统.

中间件只能通过一个高级查询接口与数据库服务器交互,同时由于它没有自己的 DBMS 引擎,不能管道处理全局查询.然而,可以在一些服务器中创建临时表来保存中间结果或远程表.中间件方法有较好的负载平衡能力,由于全局查询的工作负载被平均分配于所有 DBMS 服务器,中间件的另一优点为它是一个“轻量级”的 DBMS,一个中间件通常比另外两种方式消耗较少的资源,并需要更少的管理.与联邦及网关方式相比,它需要在临时表中存储数据,这导致额外的输入输出负担并增加了服务器的磁盘负载

数据库访问技术主要有 ODBC,OLEDB,DAO,ADO,本文采用目前流行的 ADO(ActiveX Data Object).尽管 COM 是基于二进制标准的,但是各种程序设计语言在使用 COM 的功能方面是不尽相同的.实际上,COM 与 C 语言关系最密切.COM 是基于 C++向量表接口机制的,而 C++是最擅长处理结构与指针的语言.其他程序设计语言如 VB 这方面是无法与 C++相提并论的.如果提供了双接口(该类接口只能适用于自动兼容数据类型)的话,就可以从 VB 中访问组件.OLEDB 的结构最适合 C++程序.为了给 VB 提供易于使用的接口,微软创建了 ActiveX Data Objects(ADO).ADO 看起来非常类似于目前流行的访问 JET 数据库专用的对象模型 Data Access Objects(DAO).新的 ADO 模型有 2 个主要优点.其一是该模型是线性的,因此易于使用,不需要追踪对象分层结构;其二是它是基于 OLEDB 的,因此支持程序员访问各种数据源.本系统采用 ADO 作为异构数据库访问的中间件技术,实现异构数据库的访问

A、库结构信息获取技术：在 VC++单纯利用 MFC 类获取异构型数据库的结构信息比较困难,需将 MFC 和 ADO 编程结合起来.主要利用 ADO 中的 Connection 对象和 Recordset 对象.

B、数据库集成技术：数据库集成技术就是要把一个应用环境中多个参与数据库以信息集成的方法联系起来,实现信息的共享.也就是要将多个参与数据库的信息,在逻辑上集成为一个属于多数据库系统相关的单一定义,即一个全局概念模式(GCS).数据库集成过程可分为两步:模式翻译和模式集成.模式翻译是将参与数据库的模式翻译成

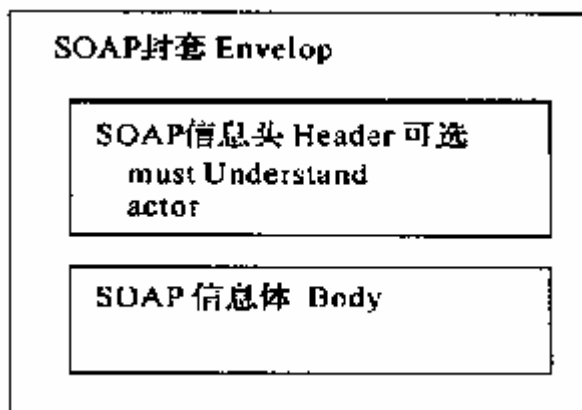
一个用公共数据模型形式的中间表示,即公共数据模型.通过确立出局部模式与中间模式的映射关系,构造出一个翻译器.每个数据库中的数据都是有关实体或者实体之间联系的数据,每个数据模型也是对实体和联系的语义和结构进行描述的.因此,在各种模型之间存在着部分等价关系,再通过建立一个辅助数据库来补充缺乏的语义部分,就可以实现不同数据模型之间的等价交换.

模式集成可按 4 个步骤进行:1 预集成.这一步主要是确定集成规则和集成优先次序.2 比较.确定出各个中间模式之间存在的语义和语法上的冲突.3 同化.这一步主要是解决第 2 步所确定出的如下冲突:()命名冲突.()格式冲突.()结构冲突.()数据丢失或冲突.4 合并和重构.这一步是要建立最后的集成模式.建立过程有二元法和多元法 2 种途径。

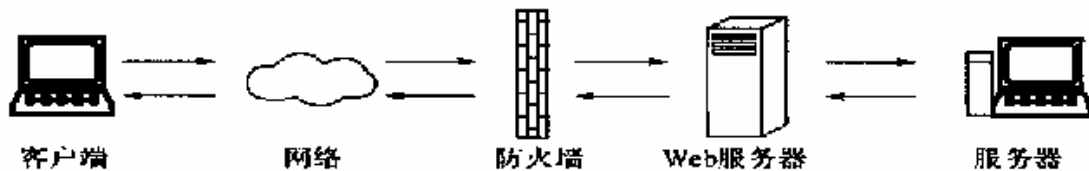
C、数据转储技术异构型数据库相互通信问题中,另一个重要方面是异构型数据库的数据转储.异构型数据库数据转储也是实现数据库信息共享,充分利用已有资源的重要措施.以前的文献中,解决异构型数据库数据转储问题时有非常大的局限性,不能进行双向的转储.为解决该问题,采用如下方法:(1)通过动态光标及动态 SQL 嵌入.应用程序可以根据需要动态构造 SQL 语句,建立相应的纪录集,利用动态光标获取纪录集中的数据信息,进行数据转储.(2)通过动态数据窗口.由于异构性数据库数据转储中,源数据库和目的数据库是根据用户需要来确定的,所以必须以动态数据库窗口的方式显示和操纵数据库.通过动态数据窗口来确定动态的纪录集.(3)建立数据类型映射系统库.在异构性数据库转储中,源数据库和目的数据库的字段类型和字段名称都可能存在着差别,这样在进行数据转储时还必须考虑到数据类型转换问题.为此,建立一个数据类型映射系统库,利用该库可以得知任意数据库间类型的转换关系.

(5) 基于 SOAP 的企业异构信息系统集成

S O A P 协议由 3 部分构成:规定了封装与请求服务有关参数的 X M L 数据包;规定了对封装数据包中请求的响应流程;规定了对封装数据包中的数据类型进行解析的编码规则[1~3]。S O A P 协议中 X M L 封包的构成如图所示,由信息头、信息体、封套构成。封套为服务的名字空间,信息体为服务名称和调用参数,信息头作为数据包的可选部分,利用信息头的属性可以在一次请求中依次调用若干个服务实现服务的流程化。



S O A P 协议中的响应流程:客户端的封包,数据包传递,服务端解包及服务程序定位,服务程序的运行及结果返回,客户端解析获得的结果。S O A P 的具体实现框架如图所示:可以看出在具体的实现中,w e b 服务器实现内部服务的对外发布、将外部对服务的请求转发到具体的内部服务;在这一过程中 X M L 消除了客户端和服务器的差异。



企业在异构的分布式环境中,使用 S O A P 来实现异构信息系统集成的具体方法是:以现有系统中的数据处理功能为基础来构建服务,并发布在 W E B 服务器上,这样对外提供使用本部门数据处理服务的接口,使用者通过调用服务来实现数据的跨部门处理,实现系统间的集成。与传统的分布式协议如 C O M / D C O M 、 C O R B A 等相比采用这一方法的优点有:(1)由于客户端通过服务端发布的接口来调用服务,服务端保持原有的接口不变就可以实现服务功能的更新,实现了系统的并行开发和及时更新;(2)在企业业务流程变更时,利用 S O A P 信息头的扩展功能,连接建立服务提高数据处理的效率;(3)在企业利用 I n t e r n e t 进行业务时,可以方便的通过防火墙调用服务,有助于提高异地办公和业务单位数据交互的效率。

目前建立 S O A P 服务的工具有很多,此次使用的是基于 J a v a 语言的 S u n O N E (O p e n N e t E n v i r o n m e n t 开放网络环境)的体系架构来进行服务器端的开发,它提供了 T o m c a t 服务器和 S O A P 服务的发布机制,以方便 S O A P 服务的建立和部署[4]。业务部和调度室的信息系统在设计时考虑了扩展性,因此采用三层结构实现,这样通过对中间层的业务逻辑的封装,利用业务和调度系统的数据处理功能对外发布服务。

(6) 基于 T I B 的企业应用集成

如果一个企业的 E R P 系统选用的是 S A P 产品, C R M 系统选用的是 S i e b e l 产品, S C M 选用的是 i 2,还有操作系统、数据库以及各种应用软件,如何实现企业应用集成(E A I)方案之一是采用信息总线 T I B (T h e I n f o r m a t i o n B u s) 技术,建立整合业务集成系统,让各种产品通过信息总线 T I B 联为一体,实现实时传递信息,并将各种软件无缝链接起来。

T I B 技术的整合业务集成系统解决方案,目的是将现存的多个系统连接起来以获得更高的效益,其中现存的系统可以包括 O r a c l e 、 S i e b e l 、 S A P 、 i 2 、 M i c r o s o f t 、 S U N 等等的系统。

1、适配器:整合业务集成企业应用的第一步是找到应用程序之间相互交流信息的方法,过去一般采用点对点或者中心辐射的方法,但这两种方法对 E A I 的需要而言都不够灵活。T I B 技术提供一个消息系统平台,让每个应用程序与系统的其余部分有一个连续点,使应用程序有即插即用和方便修改的功能,这样做也能激活不同情况下不同类型的通信,例如发布/订阅、请求/回答、安全、认证和传送。每个应用程序和消息系统的连接叫做适配器。适配器让应用程序插入消息系统中。这些适配器将应用程序的内部事件转换成消息,以便通过消息系统和其它应用程序交换消息,适配器也将收到的消息转换为要执行的操作。整合业务之前要先生成整合系统涉及到的应用程序和数据库的适配器,也可采用现成的产品,如 T I B C O 公司的 T I B C O A c t i v e E n t e r p r i s e 就已包括领先的应用程序和数据库的适配器。

2、数据表示一旦应用被插接到消息平台,他们能够传送和接收消息。通常情况下,应用程序所使用的数据格式是不兼容的,这是因为计算机硬件和操作系统平台在数据表示上使用不同的习惯,所以一个平台通常难于理解另外平台的数据。为了解决数据不同格式的问题,消息必须进行转换。T I B 技术在所有支持的平台上使用统一的数据表

示来交换消息,消息系统内的消息转换使应用程序发送和接收消息时使用它自己的格式,使转换消息独立于应用活动。消息是字段的集合,用来在所有程序中交换数据,每条消息有一个主题名来描述这个消息的目的地。所有通过 T I B 进出程序的数据必须被封装在消息的字段中。消息包括许多自我描述的数据字段,这些字段在程序、进程或线程之间流动。从程序员的角度来看,消息是字段的集合,程序可以使用 A P I 调用来处理消息。程序可以产生、删除一个消息,可以在消息中增加、删除或获取一个字段。在底层,每条消息以字节队列形式存在,这种统一的表示适合在不同的平台中进行网络通信,程序不会直接存取这种表示,这种形式是所有的 T I B 通讯的基础。

3、程序之间交互方式在基于 T I B 的整合业务集成系统中,程序和 T I B 环境之间可以有二种独特的相互作用:

- 基于事件驱动的交互方式-发布/订阅相互作用,例如到多重接受者的信息的分发;
- 基于请求驱动的交互方式-请求/应答相互作用,例如对单个服务的查询或传送。

我们基于 T I B 总线开发了一个异构数据库的集成系统。前端系统是基于 J S P 的,总数据库(M u l t i - u s e r D B)用于存放异构数据库的系统信息(如结构,用户信息等),J a v a B e a n 用于发送消息,L i s t e n e r 用于监听 J a v a B e a n 发出的消息。如果前端想访问不同平台的数据库,就通过 J a v a B e a n 将消息发送到 T I B 总线,而相应的 L i s t e n e r 从 T I B 总线上监听 J a v a B e a n 发出的消息,处理请求,存取数据库,并将处理的结果从 T I B 总线送给前端系统,前端系统直接通过 J D B C 存取总数据库。

采用基于 T I B 的业务整合解决方案,能对企业内各类应用系统进行集成,将客户、供货商和合作伙伴实时地连接起来,完成生产制造、采购、销售服务和信息传送的整合及自动化。这种整合方案相比传统的点对点集成方法和采用中心辐射的方法,都更能灵活的适应 E A I 的需要。

(8) 基于 Web Service 建立异构分布式网络计算平台

Web Service 是建立在 XML 和 HTTP 两种技术基础之上的。

Web Service 是建立可互操作的分布式应用程序的新平台,主要目标是提供跨平台的互操作性。

从最终用户的角度看,Web Service 是一个应用程序,使用 Web 技术实现动态协作的软件。

从应用开发者角度看,它是指一套软件工具,提供了一个开发平台隐藏了下层系统的异构性,使得程序之间的过程调用更加简单、透明。

从系统集成的角度来看,它是通过网络享用过程服务的一种概念,提出了一套标准,它定义了不同的应用程序如何在 WEB 上实现互操作性。

Web Service 是一种把 Internet/Intranet 变成一个统一虚拟计算环境的观念和技术,其基本理念是:把软件做成服务,并且遵从相应的标准,让不同的系统可以跨越平台,彼此相互兼容,具有进行无缝通讯和数据共享的能力。

Web Service 的核心技术如下:

- A、基于 XML 的传输标准
- B、基于 SOAP 的对象通讯标准
- C、基于 WSDL 的服务描述语言
- D、UDDI 发布描述语言:服务请求者搜索所需 Web Service 的工具,之前 Web Service 的提供者必须在公认的 UDDI 注册中心注册。

当前 Web Service 的主要实现可分为两大阵营：

- A、Microsoft 的 .NET 开发平台：只能运行在 Windows 操作系统之上。
- B、J2EE 的 Java 开发平台：可以开发从 PC 机到大型 UNIX 主机上高度可以移植的 Web 服务。

其中，由于系统的异构、地域的分布及安全防火墙的存在，传统的 RPC，COM/DCOM，J2EE 等技术就遇到了很大的挑战。Web Service 就充分显示出其优越性。分别用 .NET 和 J2EE 实现 Web Service 如下：.

A、Windows 平台基于 Microsoft.NET 框架实现 Web Service 的关键步骤：

- (a) 安装 Microsoft 的 .NET Framework 与 .NET 开发集成环境，平台要求为 Windows2000 ServicePack2 或更新的版本。
- (b) 部署 SQLServer 数据库，创建存储数据表
- (c) 首先启动 Windows2000 的 IIS Web 服务器，启动 .NET 开发环境，建立新工程，选择任一语言(vb.net 或 C#.net)，工程类型为 ASP.NET Web 服务
- (d) 编写业务逻辑代码，查询数据库返回产品并以对象形式提供给调用者。注意暴露给调用者的接口需要在声明前加[Webmethod]关键字。
- (e) 编译成功后，.NET 会在 IIS 上创建一个 Web Service，通过网页调试 Web Service 程序。

B、基于 J2EE 架构的 Web Service 实现

- (a) 安装 Java Web Services 开发包。下载 jwsdp-1-0-01-unix.sh 包，直接执行开始安装，安装程序拷贝程序包文件并安装 tomcat 应用服务器，设置相应的环境变量如：JAVA-HOME，CLASSPATH，JWSDP-HOME，PATH 等
- (b) 编写 Web Service 代码。一个 Web Service 需要至少两个文件，一个服务的接口定义文件，一个服务的实现文件。
- (c) 部署 Web Services，使用 ant 工具编译源代码，使用 wsdeploy 工具产生 wsdl 文件，使用 ant-deploy 发布 Web Services 到 tomcat Web 窗口
- (d) 在 tomcat 容器中调试、测试 Web Services

注：在 Linux 下部署、调试 Web Services 的步骤相当繁琐，详细步骤可参见 SUN 的 Web Services 开发手册。

门户网站集成几个 Web Services 向最终用户提供综合服务。

(9) 基于 Web 服务的敏捷供应链系统：

对异构平台的支持

目前企业运行的信息系统各种各样，硬件、操作系统、数据库、开发环境和对外接口千差万别，如何实现异构平台之间的信息交互，这是必须考虑的问题。总而言之，为了保证能够快速高效地构造出一个不断变化的供应链系统，使得各种信息流在企业之间无缝地流动，要求采用的技术必须有易用性、开放性、可伸缩性、跨平台能力。

笔者简单考察和分析现有应用在供应链系统的技术 CORBA。

CORBA 具有以下缺点：

A、开发成本高。CORBA 庞大而复杂，缺乏完整的技术规范支持和快捷易用的开发工具。目前国内掌握 CORBA 开发技术的人员不多，设计也需要丰富的面向对象的知识和经验。

B、CORBA 适合于服务器—服务器间的通讯，但是对于客户端—服务器的通讯十分脆弱，尤其当客户程序分布在 Internet 上更是如此。CORBA 的传输是通过 IIOP

协议完成的，不少企业的信息系统有防火墙，CORBA 的信息会被过滤。

C、不同公司的 CORBA 产品增加了一些技术特性和命令集，增加了不同公司的 CORBA 产品交互的难度。尽管不同厂商的 CORBA 能够相互操作，但是那种互操作性并不能扩展到像安全与事务管理那样的更高级别的服务中去。不仅如此，所有特定于厂商的优化在这种情况下将丢失殆尽。

由于具有以上的缺陷，CORBA 技术并不是敏捷供应链系统采用技术的首选。

相对于其它技术，Web 服务有以下的优点：

A、开放性

Web 服务的开放标准意味着它的运作成本将不会过分高昂，也不会由于技术的私有性而处处受到限制。现在，Web 服务已经成为业界共同认定的方向。许多著名的公司，Microsoft，IBM，Sun 等均已为 Web 服务市场投入巨大的人力物力，研制出各种支持 Web 服务的开发工具。有理由相信，这种技术的未来是光明的。

B、良好的封装性

一个 Web 服务的实现发生变更的时候，调用者是不会感到这一点的。对于调用者来说，只要 Web 服务的调用接口不变，Web 服务的实现任何变更对他们来说都是透明的，甚至是当 Web 服务的实现平台发生变化时，用户都可以对此一无所知。

C、跨平台集成能力

由于 Web 服务采取简单的、易理解的标准协议作为组件接口描述和协同描述规范，完全屏蔽了不同软件平台的差异，无论是何种软件平台，都可以通过这一种标准的协议进行互操作，实现了较高的可集成性。此外，SOAP 协议是建立在 HTTP 之上的，可以无阻碍地穿越企业的防火墙。

清华大学柴跃廷等人提出基于协调决策中心的供应链系统模型，将任意两个供应链实体之间视为供应商与客户的关系，再引入一个协调决策中心，对系统的信息流、物流和资金流进行协调和控制。这样，就可以将供应链内部的任何交互，如客户需求、物资运输等都通过订单来描述与完成，系统的行为表现为围绕订单进行的交易、协调、服务、认证等。

这种模型简化了供应链内部实体之间的关系，为供应链内部的信息共享提供了基础，而且具有一定的通用性。采用 Web 服务来实现这种模型，充分发挥了 Web 服务适合于异构平台之间信息交换的优点。

架构主要定义了三个角色。他们分别是：

A、供应链信息协调中心（服务代理）

- 供应链信息协调中心是整个供应链的核心，它具有以下的主要功能：
- 制定供应链中各种成员（销售商、制造商等）所提供的 Web 服务；
- 中介服务，记录和查询每个成员所提供的 Web 服务；
- 定义统一的 XML 数据格式，确保信息在企业之间的传递；
- 安全服务，对供应链成员企业之间的访问控制；
- 对成员企业信用的评估和记录；
- 工作流管理。

B、服务提供商

服务提供商的职责是根据供应链信息协调中心提供的 WSDL 文档，开发出应用程序，调用自身的信息系统，实现相应的功能，并在信息协调中心登记发布，使其可以被其他应用访问、接收及处理 Web 服务请求者的调用。

C、服务请求者

服务请求者根据自身企业的需要，通过供应链信息协调中心寻找到他们所需的

Web 服务，并且根据服务定义来绑定和调用服务提供者的 Web 服务。

Web 服务开发方式如下：

A、对于供应链信息协调中心，最大的任务是如何制定出供应链中各种成员所提供的 Web 服务。根据敏捷供应链的定义，其所涉及的成员有：供应商、制造商、分销商、客户；业务流程包括：信息流、资金流、物流。同时对于不同类型的企业，虽然其业务或功能不同，但涉及到的业务种类（例如：采购、库存、制造和销售等）却是有限的，具有相对的稳定性。因此，可以采用成熟的面向对象设计技术——UML 来定义和描述供应链中企业之间的经营活动，进而制定出每个成员在经营活动中承担的 Web 服务。除此之外，还需要建立一个私有的 UDDI 商务注册中心，支持 Web 服务的发布和查询，保证 Web 服务只被适当的成员获得。

B、对于服务提供商，由于它所要提供的 Web 服务接口已经由供应链信息协调中心定义。因此 Web 服务开发采用自顶向下的方案：从供应链信息协调中心获得描述 Web 服务的 WSDL 文档和传送数据类型的 XSD 定义，然后根据自身内部系统的实际情况，采用支持 Web 服务开发工作，例如 IBM 的 WSAD，Microsoft 的 Visual Studio.NET，开发出实现的组件。通过内部的单元测试之后，便可以在信息协调中心发布。

C、对于服务请求者，由于敏捷供应链的动态性，Web 服务的提供者也在不断地变化，而且每种类型的成员提供的 Web 服务由信息协调中心统一定义服务接口，所以采用开发时动态连接的方式，动态获得 Web 服务提供商，然后逐一进行服务调用。

目前主流的开发平台提供商 IBM，Microsoft 等都在大力支持和推动 Web 服务的开发，只要根据 WSDL 文档，开发工具便能自动生成相应的服务接口，这样开发者就可以把精力集中在服务的企业业务之上。这使得 Web 服务的开发大大简化。

(10) 网络技术

数据网络的概念来自网格(Grid)，它是网格技术在数据管理方面的应用和实现\$即是为了建立网格环境下，透明访问异构数据资源的新的体系结构。

网格技术的研究目标是实现网络虚拟环境下高性能资源的共享和协同工作\$以解决一致使用各种分散资源的问题。数据网络是为了解决数据密集型计算应用中，方便高效使用分布式数据资源的问题\$其研究内容主要集中在广域、异构、分布环境下如何对数据进行管理；如何从地理分布的各种异构数据资源中获取数据；并通过地域分布的协作和处理，从数据源中获取有用信息。从这个角度，数据网络可以理解为是应用程序!用户等同底层存储系统之间的中间件，它对应用程序提供一致的管理和使用存储系统的界面\$对用户屏蔽广域操作的复杂性和广域网络上存储系统的异构性。

从体系结构上来讲，数据网络有以下特征：

A、异构性(Heterogeneity):数据网络可以包含多种异构的数据资源。其构成的数据源有多种类型，不同类型的数据源在体系结构、数据访问方式、应用程序接口等多个层次上可能有不同的结构。

B、可扩展性(Scalability) 数据网络的规模可以变化，从只包含少量数据资源的局域系统，发展到包括成千上万数据资源的跨洲际的广域大数据网络。由此可能带来的一个问题就是随着数据网络资源的增加、地域分布的更广泛而引起的性能下降及网络传输延迟，数据网络必须能适应这种变化。

C、自适应性 (Adaptability):在广域系统中,有很多数据和存储资源, 这些资源发生故障的概率很高\$因此数据网络必须能对用户和应用程序屏蔽这些故障,并动态地适应这些情况;此外,数据网络资源因地域分布和系统复杂使其整体结构经常发生变化,数

据网格应能适应这种不可预测的结构。

D、多级管理域，由于构成数据网络的资源和存储系统通常属于不同的机构或组织，并使用不同的安全机制，因此需要各个机构或组织共同参与，以解决多级管理域的问题。

根据上述特点及应用的实际情况，在设计数据网络时要根据以下原则进行：

A、与底层实现机制无关性：数据网络的结构应该与底层实现机制无关。包括与数据存储方式、元

B、数据访问方式等无关：系统可以定义一些界面或接口。以封装不同存储系统在数据存储、目录管理、数据传输算法等方面的异构性

C、与应用策略无关性：数据网络结构在设计中。应与与性能相关的一些重要功能的实现方式留给用户或是应用程序来选择。而不是由系统封装执行。这样既保证了用户实现策略的灵活性。又提高了应用系统的性能。如"数据移动和复制文件目录管理是数据网络的一个基本功能，而建立在其上的复制策略应由用户或应用程序来决定！

D、与网格框架兼容：网格作为广域异构分布资源互联的基础设施，为上层系统提供了许多基础服务。如：认证、资源管理、信息服务等，数据网络的设计应基于网格框架"并与底层网格机制相兼容。

E、一致的信息结构，在网格中"用户以一致的方式访问资源的结构信息和状态信息。同样，在数据网络中，用户也要能以一致的方式访问数据网络中的各种信息；如，元数据、复制文件目录等！

2、多源异构数据的整合与互操作技术分析比较

(1)数据库访问技术比较

A、直接访问特定的数据库：其采用了专用的技术，针对选定的对象，访问效率最高。但，其缺点是显而易见的，采用的技术难以掌握，通用性差，开发成本相对较好，无法在一般的应用系统中使用。

B、DAO (Data Access Objects)，适用于单系统应用程序或小范围本地分布使用，这种技术在新的应用中已经不再使用了，但由于很多老的桌面系统采用这种技术，在系统升级时还会用到。

C、RDO (Remote Data Objects)，是一个基于 ODBC 的面向对象的数据访问接口。它的目的是访问大型数据库，不能很好地访问 Jet 或 ISAM 数据库。

D、ODBC API，是到现在仍然广泛使用的通用数据库接口技术。ODBC 的引入，帮助应用系统将数据层、数据访问层与应用层有效地隔开，极大提高了编程的效率。从早到晚 ODBC API 比较难学，而且使用时很容易出错。

E、ADO (ActiveX Data Object)是 DAO/RDO 的后继产物。尽管 ADO 在结构上与 RDO 看起来很像，但在性能和执行效率上有极大的提高，并且采用 ADO 可以访问各种数据库，使程序的可移植性大大提高。直到现在，ADO 仍然是一个可以选择的数据访问方案。

F、RDS，是 ADO 的一部分，目的是通过 IIS 实现数据库的远程访问。使用它可以为用户提供许多方便，如不用在本地配置 ODBC 数据源。RDS 在 ADO 的基础上通过绑定的数据显示和操作控件，提供给客户端更强的数据表现力和远程数据操纵功能。随着 XML，Web Service 的应用，新的应用不建议使用 RDS。

G、JDBC，是执行 SQL 的 Java API。JDBC 为 Java 提供了数据库开发的标准 API，使用户可以用纯 Java API 来编写数据库应用。通过 JDBC 与 ODBC 的桥接可以访问大多数数据库。

(2) XML 的优势

- A、使用简单
- B、开放性强
- C、易于扩充
- D、国际化
- E、适合于系统间的数据交换
- F、提高 Web 客户端的功能
- G、有利于信息内容的修改化定制

(3) Web Service 在的优缺点:

Web Service 的主要目标是跨平台的可互操作性。为了达到这一目标, Web Service 完全基于 XML (可扩展标记语言)、XSD (XML Schema) 等独立于平台、独立于软件供应商的标准, 是创建可互操作的、分布式应用程序的新平台。由此可以看出, 在以下四种情况下, 使用 Web Service 会带来极大的好处。

A、跨[防火墙](#)的通信

如果应用程序有成千上万的用户, 而且分布在世界各地, 那么客户端和服务端之间的通信将是一个棘手的问题。因为客户端和服务端之间通常会有[防火墙](#)或者代理服务器。在这种情况下, 使用 DCOM 就不是那么简单, 通常也不便于把客户端程序发布到数量如此庞大的每一个用户手中。传统的做法是, 选择用浏览器作为客户端, 写下一大堆 ASP 页面, 把应用程序的中间层暴露给最终用户。这样做的结果是开发难度大, 程序很难维护。

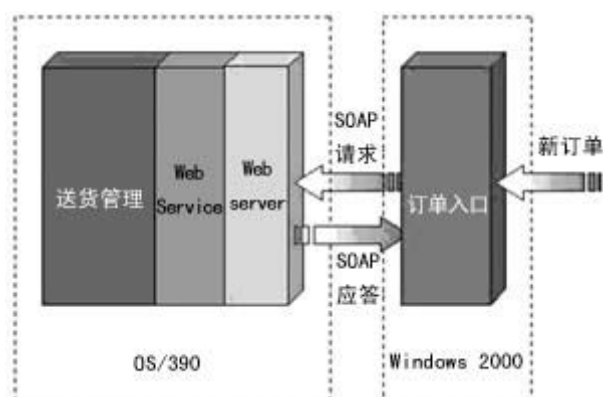


图 1 通过 Web Service 集成应用程序

举个例子, 在应用程序里加入一个新页面, 必须先建立好用户界面(Web 页面), 并在这个页面后面, 包含相应商业逻辑的中间层组件, 还要再建立至少一个 ASP 页面, 用来接受用户输入的信息, 调用中间层组件, 把结果格式化为 HTML 形式, 最后还要把“结果页”送回浏览器。要是客户端代码不再如此依赖于 HTML 表单, 客户端的编程就简单多了。

如果中间层组件换成 Web Service 的话, 就可以从用户界面直接调用中间层组件, 从而省掉建立 ASP 页面的那一步。要调用 Web Service, 可以直接使用 Microsoft SOAP Toolkit 或 .NET 这样的 SOAP 客户端, 也可以使用自己开发的 SOAP 客户端, 然后把它和应用程序连接起来。不仅缩短了开发周期, 还减少了代码复杂度, 并能够增强应用程序的可维护性。同时, 应用程序也不再需要在每次调用中间层组件时, 都跳转到

相应的“结果页”。

从经验来看，在一个用户界面和中间层有较多交互的应用程序中，使用 **Web Service** 这种结构，可以节省花在用户界面编程上 20% 的开发时间。另外，这样一个由 **Web Service** 组成的中间层，完全可以在应用程序集成或其它场合下重用。最后，通过 **Web Service** 把应用程序的逻辑和数据“暴露”出来，还可以让其它平台上的客户重用这些应用程序。

B、 应用程序集成

企业级的应用程序开发者都知道，企业里经常都要把用不同语言写成的、在不同平台上运行的各种程序集成起来，而这种集成将花费很大的开发力量。应用程序经常需要从运行在 **IBM** 主机上的程序中获取数据；或者把数据发送到主机或 **UNIX** 应用程序中去。即使在同一个平台上，不同软件厂商生产的各种软件也常常需要集成起来。通过 **Web Service**，应用程序可以用标准的方法把功能和数据“暴露”出来，供其它应用程序使用。

例如，有一个订单登录程序，用于登录从客户来的新订单，包括客户信息、发货地址、数量、价格和付款方式等内容；还有一个订单执行程序，用于实际货物发送的管理。这两个程序来自不同软件厂商。一份新订单进来之后，订单登录程序需要通知订单执行程序发送货物。通过在订单执行程序上面增加一层 **Web Service**，订单执行程序可以把“**Add Order**”函数“暴露”出来。这样，每当有新订单到来时，订单登录程序就可以调用这个函数来发送货物了。

C、 B2B 的集成

用 **Web Service** 集成应用程序，可以使公司内部的商务处理更加自动化。但当交易跨越供应商和客户、突破公司的界限时会怎么样呢？跨公司的商务交易集成通常叫做 **B2B** 集成。

Web Service 是 **B2B** 集成成功的关键。通过 **Web Service**，公司可以把关键的商务应用“暴露”给指定的供应商和客户。例如，把电子下单系统和电子发票系统“暴露”出来，客户就可以以电子的方式发送订单，供应商则可以以电子的方式发送原料采购发票。当然，这并不是一个新的概念，**EDI**(电子文档交换)早就是这样了。但是，**Web Service** 的实现要比 **EDI** 简单得多，而且 **Web Service** 运行在 **Internet** 上，在世界任何地方都可轻易实现，其运行成本就相对较低。不过，**Web Service** 并不像 **EDI** 那样，是文档交换或 **B2B** 集成的完整解决方案。**Web Service** 只是 **B2B** 集成的一个关键部分，还需要许多其它的部分才能实现集成。

用 **Web Service** 来实现 **B2B** 集成的最大好处在于可以轻易实现互操作性。只要把商务逻辑“暴露”出来，成为 **Web Service**，就可以让任何指定的合作伙伴调用这些商务逻辑，而不管他们的系统在什么平台上运行，使用什么开发语言。这样就大大减少了花在 **B2B** 集成上的时间和成本，让许多原本无法承受 **EDI** 的中小企业也能实现 **B2B** 集成。

D、 软件和数据重用

软件重用是一个很大的主题，重用的形式很多，重用的程度有大有小。最基本的形式是源代码模块或者类一级的重用，另一种形式是二进制形式的组件重用。

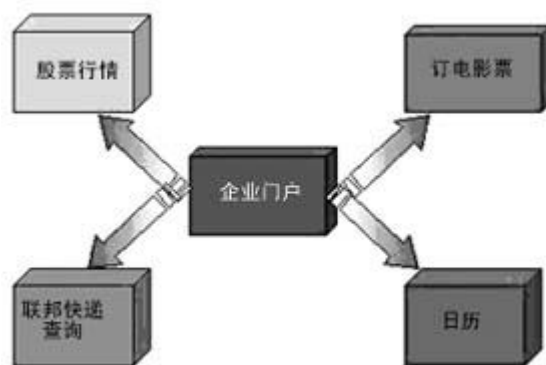


图 2 用 Web Service 集成各种应用中的功能，为用户提供一个统一的界面

当前，像表格控件或用户界面控件这样的可重用软件组件，在市场上都占有很大的份额。但这类软件的重用有一个很大的限制，就是重用仅限于代码，数据不能重用。原因在于，发布组件甚至源代码都比较容易，但要发布数据就没那么容易，除非是不会经常变化的静态数据。

Web Service 在允许重用代码的同时，可以重用代码背后的数据。使用 Web Service，再也不必像以前那样，要先从第三方购买、安装软件组件，再从应用程序中调用这些组件；只需要直接调用远端的 Web Service 就可以了。举个例子，要在应用程序中确认用户输入的地址，只需把这个地址直接发送给相应的 Web Service，这个 Web Service 就会帮你查阅街道地址、城市、省区和邮政编码等信息，确认这个地址是否在相应的邮政编码区域。Web Service 的提供商可以按时间或使用次数来对这项服务进行收费。这样的服务要通过组件重用来实现是不可能的，那样的话你必须下载并安装好包含街道地址、城市、省区和邮政编码等信息的数据库，而且这个数据库还是不能实时更新的。

另一种软件重用的情况是，把好几个应用程序的功能集成起来。例如，要建立一个局域网上的门户网站应用，让用户既可以查询联邦快递包裹，查看股市行情，又可以管理自己的日程安排，还可以在线购买电影票。现在 Web 上有很多应用程序供应商，都在其应用中实现了这些功能。一旦他们把这些功能都通过 Web Service “暴露”出来，就可以非常容易地把所有这些功能都集成到你的门户网站中，为用户提供一个统一的、友好的界面。

将来，许多应用程序都会利用 Web Service，把当前基于组件的应用程序结构扩展为组件/Web Service 的混合结构，可以在应用程序中使用第三方的 Web Service 提供的功能，也可以把自己的应用程序功能通过 Web Service 提供给别人。两种情况下，都可以重用代码和代码背后的数据。

从以上论述可以看出，Web Service 在通过 Web 进行互操作或远程调用的时候是最有用的。

不过，也有一些情况，Web Service 根本不能带来任何好处。

A、 单机应用程序

目前，企业和个人还使用着很多桌面应用程序。其中一些只需要与本机上的其它程序通信。在这种情况下，最好就不要用 Web Service，只要用本地的 API 就可以了。COM 非常适合于在这种情况下工作，因为它既小又快。运行在同一台服务器上的服务器软件也是这样。最好直接用 COM 或其它本地的 API 来进行应用程序间的调用。当然 Web Service 也能用在这些场合，但那样不仅消耗太大，而且不会带来任何好处。

B、局域网的同构应用程序

在许多应用中，所有的程序都是用 VB 或 VC 开发的，都在 Windows 平台下使用 COM，都运行在同一个局域网上。例如，有两个服务器应用程序需要相互通信，或者有一个 Win32 或 WinForm 的客户程序要连接局域网上另一个服务器的程序。在这些程序里，使用 DCOM 会比 SOAP/HTTP 有效得多。与此相类似，如果一个 .NET 程序要连接到局域网上的另一个 .NET 程序，应该使用 .NET remoting。有趣的是，在 .NET remoting 中，也可以指定使用 SOAP/HTTP 来进行 Web Service 调用。不过最好还是直接通过 TCP 进行 RPC 调用，那样会有效得多。

总之，只要从应用程序结构的角度看，有别的方法比 Web Service 更有效、更可行，那就不要用 Web Service。

因此，对于 Web Service 要正确地看待。

3、面向物流的多源异构数据的整合与互操作技术的特殊性

物流行业信息综合平台，在进行多源异构数据的整合与互操作，有以下一些特殊性：

- (1) 数据源种类多，涉及许多行业与机构，对集成带来一定的复杂度。
- (2) 数据源对应的信息系统管理水平参差不齐，对集成带来一定的难度。
- (3) 面向物流的多源异构数据的整合与互操作技术要求必须达到一定的响应速度。
- (4) 面向物流的多源异构数据在整合，要利用许多新的技术，如 RFID、GPS、GDS 等信息识别与定位方法。

四、本人对面向物流的多源异构数据的整合与互操作的观点

1、面向物流的多源异构数据的整合与互操作的研究具有重要的战略意义和实际应用价值

面向物流的多源异构数据的整合与互操作技术研究将为上海市实现现代物流信息技术与产业化发展战略，以物流信息标准化技术为支撑，以物流信息平台技术为基础，通过应用示范工程实现物流信息技术的应用集成，构筑上海城市现代物流信息平台发展框架，推动上海现代物流产业的发展提供宏观技术支持，从而中国物流信息化的进程。

成功实现面向物流的多源异构数据的整合与互操作将在以下四个方面具有实际的意义：

- (5) 适应信息技术的发展，增强中国物流行业的竞争力。
- (6) 充分利用物流行业的信息资源。
- (7) 降低中国物流行业的总体成本，从而降低物流成本在社会总成本中的比例。
- (8) 为中国物流及类似行业提供技术经验。

2、主系统与子系统的两个层面必须做到相互协作

面向物流的多源异构数据的整合与互操作要求必须在主系统与子系统两个层面做到相互协作。

(1) 子系统：包括前述的各类行业及机构，如陆路运输行业、口岸运输行业（海运与空运）、主要物流园区、物流供应链中的各类参与企业、及各类社会机构（银行、海关等）。

这些子系统，必须做到子系统内部的信息管理系统要建立，并能提供真实、完整、有效的数据源。对于某些信息管理落后的行业及机构，应在主系统进行整合之前，对这些子系统进行必要的完善与优化。有些行业或机构的信息效率低下的原因是没有认真地调查分析信息管理的需求与环境，很多时候，这些子系统的信息管理中，技术不是主要的问题。

(2) 主系统：即总的物流信息平台，它要将上述的子系统进行整合，并能提供有效的互操作能力。

主系统的建设主要应建立在子系统完善的前提下，通过以物流信息编码技术的标准的物流信息接口的标准为基础，利用现有的先进的中间件等技术设计综合的物流信息平台。

3、面向物流的多源异构数据的整合与互操作必须做好三种接口设计

(1) 对被整合信息资源对象的接入接口，即对各类行业与机构子系统的多源异构数据源整合时必须有统一的接口，从而方便主系统的整合。

(2) 对使用新信息系统的各类用户的输出接口。这是为了让物流综合信息平台的信息，能及时返加给加入综合信息平台的各类行业与机构，使这些子系统的数据库能及时更新。

(3) 对新平台上运行的各种应用系统的支持接口。即对整合后的数据，要求能给物流综合信息平台的各种应用系统提供可使用的接口。

4、面向物流的多源异构数据的整合的主要三个方面

面向物流的多源异构数据的整合的主要应注意三个方面的整合：

(1) 从应用系统整合看，已有十几年的历史，从单个应用与单个应用之间进行消息传递出现中间件技术到实现信息总线（Info Bus）到大规模专业化企业应用集成服务（EAI）到企业内和企业外的应用集成（B2Bi），与此相伴还有工作流管理技术（Workflow）以及发展到今天的业务流程管理技术（BPM）。

(2) 从数据资源整合看，也有十几年的历史，从单一小型数据库发展到跨多种数据库的驱动访问到海量管理的 ETL 技术、数据仓库技术、商业智能数据分析技术（BI）乃至今天在元数据技术的驱动下，实现实时抽取与个性化数据展现的虚拟数据库技术。

(3) 从内容管理整合看，伴随着互联网技术的高速发展，以内容管理为主要内容的各种技术，如内容管理技术（CMS）、门户管理技术（EIP）、企业门户技术（EP）、知识管理技术（KM）乃至最近提出的智能企业套件技术（SES）。

5、先进合理的设计原则和对未来技术的开放性是成功实现面向物流的多源异构数据的整合与互操作的关键

对于面向物流的多源异构数据的整合与互操作而言，其成功实现的关键是必须要有先进合理的设计原则和良好的控制体系结构。

一方面，设计原则必须是先进合理的，既在多源异构数据集成的同时，要使数据达到完整性、一致性及安全性。

另一方面，整个设计必须有一定的前瞻性，使整个系统平台对未来具有良好的开放性，能尽可能容纳今后信息技术发展中出现的新的技术与标准。

参考文献:

- 1, Greg Barish, etc., 2002, *A Case Study In Information Integration*, [M], TheaterLoc, ,
- 2, Junfeng Qu, 2002, *An Introduction to Data Mining Technology*, [M], CSCI 6900, ,
- 3, Francisco Curbera, 2002, *Building Web Service Demand Management For/From Web Services Model*, [M], CSFB, ,
- 4, David Theriault & Jacqui Cadell, 2001, *Business considerations for Enterprise Application Integration* , [M], GITA 2001 , ,
- 5, Vijayshankar Raman, etc., 2002, *Data Access and Management Services on Grid*, [M], IBM Almaden Research Center, ,
- 6, Jiawei Han and Micheline Kamber, 2003, *Data Mining: Concepts and Techniques*, [M], University of Illinois , ,
- 7, DataMirror, 2001, *Data Warehousing Uncovered: The Inside Framework of Real-time Data Warehouses*, [M], DataMirror Corporation White Paper, ,
- 8, William L. Scherlis, 2002, *Information Technology Research, Innovation and E-Government*, [M], CSTB, ,
- 9, Kenneth C. Laudon, Jane P. Laudon, 2001, *MANAGEMENT INFORMATION SYSTEMS - Organization and Technology in the Networked Enterprise (Sixth Edition)*, [M], Pearson Education, , P368-397
- 10, Satine, Consortium, 2003, *Semantic-based Interoperability Infrastructure for Integrating Web Service Platforms to Peer-to-Peer Networks for Travel Industry*, [M], METU, ,
- 11, Fibre Channel Industry Association, 2001, *Storage Virtualization Brief*, [M], Business Solutions Whitepapers, ,
- 12, IBM, 2001, *Supply Chain Management Solutions*, [M], IBM Corp., ,
- 13, Vic Werner and Craig Abramson, 2003, *The Critical Business Need to Reduce Elapsed Time*, [J], Business Intelligence JOURNAL, Vol8, No2,2003,
- 14, Fred M. Domke, 2004, *TranXML - XML for Transportation*, [M], Washington University, ,
- 15, Jay Unger, Matt Haynos, 2003, *A visual tour of Open Grid Services Architecture*, [M], IBM DevolperWorks, ,
- 16, 宋文官, 2002, 电子商务实用教程(第二版), [M], 高等教育出版社, 2002 年第 2 版, P206-222
- 17, 常晋义等, 2002, 管理信息系统, [M], 中国电力出版社, 2002 年第 1 版, P71-82
- 18, 朱道立等, 2001, 物流和供应链管理, [M], 复旦大学出版社, 2001 年第 1 版, P53-66
- 19, 彭望勤等, 2003, 物流实务手册, [M], 立信会计出版社, 2002 年第 1 版, P640-644
- 20, 现代物流管理课题组, 2002, 物流信息管理, [M], 广东经济出版社, 2002 年第 1 版, P29-86
- 21, 赵刚, 2002, 物流信息系统, [M], 四川人民出版社, 2002 年第 1 版, P145-171
- 22, 周城, 2002, 物流信息化解决方案, [M], 四川人民出版社, 2002 年第 1 版, P82-169
- 23, 甄镭, 2004, 信息系统升级与整合: 策略、方法、技巧, [M], 电子工业出版社, 2004 年第 1 版, P10-40,217-253
- 24, 李兵等, 2003, CAX——一种有效的数据集成方法, [J], 小型微型计算机系统, 第 24 卷第 8 期, P1488-1490
- 25, 李贵荣, 2003, WebGIS 的三层应用结构模型及其实现技术, [J], 南方冶金学院学报, 第 24 卷第 1 期, P33-36

- 26, 何忠等, 2004, XML 映射器的实现, [J], 计算机工程与应用, 2004 年第 4 期, P137-139,187
- 27, 王东等, 2003, 城市社区现代物流运作与物流信息平台规划, [J], 港口经济, 2003 年第 2 期, P48-49
- 28, 杨柳等, 2003, 第三方物流企业信息平台的研究与开发, [J], 港口装卸, 2003 年第 3 期(总第 149 期), P8-9
- 29, 周庆佐等, 2004, 多信息系统数据交换的研究与实现, [J], 计算机工程与应用, 2004 年第 3 期, P164-165,197
- 30, 邓铁清等, 2003, 多源数据转换与装载向导开发技术, [J], 计算机应用研究, 2003 年第 9 期, P85-87
- 31, 洪志全等, 2003, 分布式数据服务中间件实现技术研究, [J], 计算机应用研究, 2003 年第 1 期, P46-49
- 32, 王家华等, 2003, 基于 COM+异构数据库访问实现技术, [J], 西安石油学院学报(自然科学版), 第 18 卷第 6 期, P79-82
- 33, 张立明等, 2003, 基于 CORBA 的数据集成中间件的设计与实现, [J], 华中科技大学学报(自然科学版), 第 31 卷第 11 期, P11-13
- 34, 张俊等, 2004, 基于 SOAP 的企业异构信息系统集成, [J], 南京工业大学学报, 第 26 卷第 1 期, P95-97
- 35, 袁秀梅等, 2003, 基于 Web Service 建立异构分布式网络计算平台, [J], 北京工业职业技术学院学报, 第 2 卷第 4 期, P40-45
- 36, 肖颖浩等, 2004, 基于 Web 服务的敏捷供应链系统, [J], 计算机工程与应用, 2004 年第 7 期, P224-226
- 37, 叶蕾等, 2003, 基于 XML 的异构数据集成领域的应用研究, [J], 吉林师范大学学报, 2003 年第 3 期, P21-23
- 38, 田绪红等, 2003, 基于 XML 的异构数据库查询技术研究, [J], 计算机工程与应用, 2003 年第 22 期, P182-184
- 39, 王忠群等, 2003, 基于 XML 的异构数据源的集成研究, [J], 安徽工程科技学院学报, 第 18 卷第 4 期, P37-43
- 40, 李焕荣等, 2004, 基于 XML 的战略网络信息平台研究与实现, [J], 计算机工程与应用, 2004 年第 7 期, P172-175
- 41, 王操等, 2003, 基于 XML 和虚拟数据中心的网上城市数据集成, [J], 计算机工程, 第 29 卷第 21 期, P61-63
- 42, 周晓峰等, 2003, 基于多 Agent 的分布式异构数据源访问机制研究, [J], 计算机工程与设计, 第 24 卷第 4 期, P5-8
- 43, 杨立等, 2003, 基于分布式异构数据库的企业信息集成平台, [J], 计算机与现代化, 2003 年第 1 期(总第 89 期), P34-37
- 44, 李仁庆等, 2003, 基于神经网络技术的异构数据库集成查询, [J], 大连轻工业学院学报, 第 22 卷第 2 期, P146-150
- 45, 张蒲生等, 2003, 基于政务数据仓库的数据集成及关键技术, [J], 计算机工程, 第 29 卷第 9 期, P179-182
- 46, 郭权等, 2004, 浅析网格相关应用技术, [J], 计算机工程与应用, 2004 年第 6 期, P14-17
- 47, 周恒等, 2003, 区域物流信息系统和平台及关键技术分析, [J], 交通标准化, 2003 年第 7 期(总第 119 期), P50-54
- 48, 蔡淑琴, 2003, 区域性物流信息平台结构的研究, [J], 武汉理工大学学报·信息与管理工程版, 第 25 卷第 3 期, P94-98

- 49, 邓致格等, 2003, 商业企业物流信息系统(LIS)的分析与设计, [J], 云南财经大学学报, 第 17 卷第 6 期, P157-159
- 50, 何杰等, 2003, 省级物流信息平台体系结构方案分析, [J], 交通科技, 2003 年第 6 期(总第 201 期), P72-74
- 51, 石玉晶等, 2003, 使用 XML 进行异构数据库间数据传送, [J], 现代计算机, 2003 年总第 175 期, P79-81
- 52, 何戈等, 2003, 数据网格技术研究, [J], 微电子学与计算机, 2003 年第 4 期, P3-10
- 53, 徐慧等, 2003, 信息系统集成技术与开发策略的研究, [J], 苏州大学学报(自然科学版), 第 19 卷第 4 期, P40-47
- 54, 姜大立等, 2003, 虚拟物流信息平台的结构与功能分析, [J], 铁道运输与经济, 第 25 卷第 8 期, P18-21
- 55, 王逢春等, 2004, 一种基于数据集成的 EAI 解决方案, [J], 工业工程与管理, 2004 年第 1 期, P14-18
- 56, 李晖等, 2004, 异构分布环境下数据获取与多态发布, [J], 计算机工程与设计, 第 25 卷第 1 期, P111-114
- 57, 贺卫红等, 2004, 异构环境下分布组件的对象互操作性的研究, [J], 计算机应用研究, 2004 年第 2 期, P79-82
- 58, 马桂芳等, 2003, 异构环境下基于 Web 的信息系统的设计与实现, [J], 计算机工程与应用, 2003 年第 12 期, P200-202
- 59, 马淑娇, 2004, 异构数据库集成中的 XML 技术探讨, [J], 计算机应用研究, 2004 年第 1 期, P94-96
- 60, 胡忠望, 2003, 异构网络环境中跨平台共享数据存取的解决方案, [J], 现代计算机, 2003 年总第 171 期, P40-43
- 61, 李建飞, 2003, 运用 TMIS 系统构建铁路物流信息平台, [J], 事业财会, 2003 年第 6 期(总第 86 期), P59-62
- 62, 迟永林, 2003, 现代物流信息系统规划与设计的研究, [J], 西南交通大学硕士学位论文, ,
- 63, 桑作军, 2002, 现代供应链管理系统研究, [J], 南京理工大学硕士学位论文, ,
- 64, 赵艳, 2003, 西航公司技术改造项目管理信息系统研究, [J], 西北工业大学硕士学位论文, ,
- 65, 徐艳, 2002, 区域物流系统的信息平台研究, [J], 武汉理工大学硕士学位论文, ,
- 66, 顾强, 2002, 面向制造企业供应链管理系统集成平台的体系结构及开发研究, [J], 浙江大学硕士学位论文, ,