一个数据共享系统的实现①

徐济惠, 蒋 宁

(宁波城市职业技术学院 信息学院, 宁波 315110)

摘 要: 针对一个真实的医疗信息数据共享系统的需求, 结合用户需求和数据特点确定数据仓库的主题, 采用了数据采集、数据抽取、联机分析处理等技术, 并且通过 MQ 技术实现了数据交换机制, 具体的实施则是通过结合一些开源工具的有效使用, 最终建立了一套架构先进、功能完善的业务数据共享信息平台.

关键词: 数据共享; 数据仓库; 联机分析处理; 开源工具

Realization of a Data Sharing System

XU Ji-Hui, JIANG Ning

(Information School, Ningbo City College of Vocational Technology, Ningbo 315110, China)

Abstract: This paper intends to study and build data sharing platform for a requirement of medical information data. We defined the database warehouse subject due to the requirement of the user. Then adopt data acquisition mainly, data extraction and OLAP. We realize a data transfer mechanism through MQ technology. The specific implementation mainly consist of utilization of some open source softwaresand finally established a data sharing platform with advanced architecture.

Key words: data sharing; datawarehouse; OLAP; open source software

1 前言

由于省医疗中心各科室和研究机构每年需要开展广泛的疾病监测与调研活动, 汇总出海量数据与文档资料. 中心需要收集、汇总各个业务所的业务数据, 并对这些资料加以编排, 工作量大. 目前大量业务数据分散存储, 且未建立数据库, 关联和共享不便, 难以实现统计分析和辅助决策.

因此, 省医疗中心内部拟建立一套架构先进、功能完善的业务数据共享信息平台, 对医疗中心各科所的监测结构化数据与非结构化数据实现集中、统一的管理, 实现对海量业务数据的汇总、统计、查询、挖掘. 因此需要根据省医疗业务逻辑对现存海量历史数据的全面梳理和结构化, 对新增数据能通过自定义结构化平台自动归档入库, 并通过数据挖掘和商业智能系统进一步提高医疗中心科学化分析决策的能力.

该系统由我们主持设计开发, 平台基于 J2EE 架构, 通过数据仓库技术和数据共享技术, 结合一些开源工具的有效使用, 最终建立了一套架构先进、功能完善的业务数据共享信息平台.

2 系统功能需求分析

省医疗中心“数据共享系统”的开发与实施, 主要为了实现以下几个方面的目标:

(1) 建立完整、完善的省医疗中心多维度信息资源目录体系结构, 来有效管理省医疗中心每年汇总生成的大量业务数据;

(2) 对省医疗中心现存的达上千张的不同类型表单与响应数据进行梳理、分类汇总并结构化, 最终入库;

(3) 提供强大、完善的查询和统计分析功能.

(4) 通过灵活的权限设置, 来实现不同级别的不同人员拥有不同的数据查阅权限.

(5) 实现新监测数据向历史业务数据的自动归档.

① 收稿时间:2013-08-26;收到修改稿时间:2013-09-30

(6) 实现对业务数据的数据挖掘和 OLAP 分析.

3 系统总体设计

3.1 系统架构设计

3.1.1 系统体系结构

本系统基于 J2EE 多层体系结构[1]进行设计, 其中包括:

① WEB 客户端层;

② WEB 服务器层;

③ 数据持久层.

其架构示意图如图 1 所示:

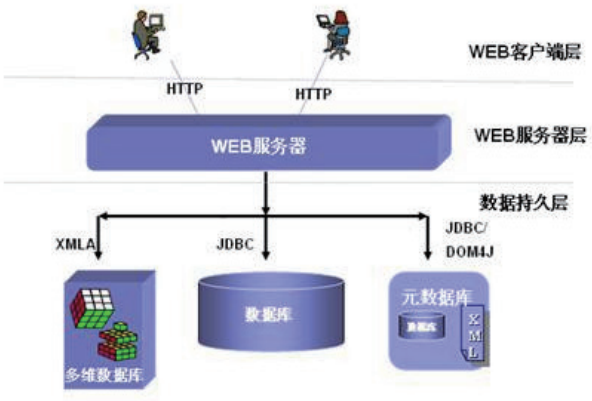

图 1　三层架构示意图

3.1.2 系统数据流程

省医疗中心各科室和研究机构在每年开展的疾病监测与调研活动及日常工作中积累了大量的业务数据和其它的一些文档数据, 数据仓库首先通过数据交换平台从这些数据源中抽取相关的数据, 进行数据集成、转换和综合, 将数据重新组合成面向全局的数据视图, 为统计分析和辅助决策提供数据存储和组织的基础, 解决了数据不一致的问题.

数据仓库包括大量的业务处理系统的操作细节数据和其它的综合数据[2], 而在对一个组织或机构的管理分析与决策中, 人们所关心的大多是综合性数据, 需要从综合性的、总的范围来观察数据.

为此, 我们通过 OLAP 数据仓库使用技术, 可以把数据在一定层次上聚集, 方便用户的快速查询, 以及从多维的角度对数据进行切片、切块、旋转等操作[3], 进一步增加用户对数据的理解.

图 2 为系统的数据流程与相关功能关系图.

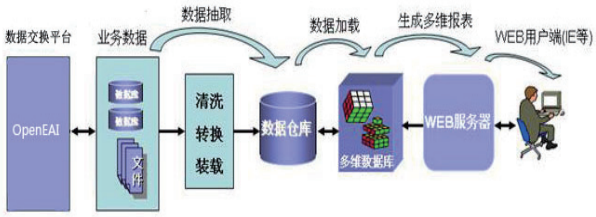

图 2　平台数据流程图

3.2 构建数据仓库

3.2.1 关键技术

数据仓库的构建偏向于工程, 具有强烈的工程性, 其关键技术主要有数据采集、数据抽取等几个方面.

(1) 数据采集

省医疗中心各科室和研究机构在每年开展的疾病监测与调研活动及日常工作中积累了大量的业务数据和其它的一些文档数据, 其中包括主流数据库的结构化数据(如 DB2、Oralce、SQL Server、ODBC 数据源), 还包括 XML 文件、文本文件、EXCEL、Web Service 等非结构化数据[4], 大量需要手工录入的纸质数据, 以及脱机的数据存储介质中的数据等.

对于需要手工录入的纸质数据, 我们将 OCR 扫描软件能够识别的部分进行电子化识别并人工校验; 对于不能识别的部分, 我们将通过自定义表格手工录入.

对于脱机数据, 我们将通过数据存储介质(例如 U 盘, 移动硬盘等)进行采集.

(2) 数据抽取

数据的抽取是数据进入数据仓库的入口. 由于数据仓库是一个独立的数据环境, 它需要通过抽取过程将数据从数据库系统、外部数据源、脱机的数据存储介质中导入到数据仓库. 数据抽取在技术上主要涉及互连、复制、增量、转换、调度和监控等几个方面[5]. 数据仓库的数据并不要求与数据源保持实时的同步, 因此数据抽取可以定时进行, 但多个抽取操作执行的时间、相互的顺序、成败对数据仓库中信息的有效性则至关重要.

数据抽取包括数据清洗、整合、转换、加载等各个过程. ETL 抽取整合数据的好坏直接影响到最终的结果展现. 所以 ETL 在整个数据仓库项目中起着十分关键的作用.

本系统中我们使用一款著名的开源 ETL 工具 Kettle 实现业务数据的抽取.

Kettle 是一款开源的、元数据驱动的 ETL 工具, 纯 java 编写, 绿色无需安装, 数据抽取高效稳定. Kettle 中有两种脚本文件, transformation 和 job, transformation 完成针对数据的基础转换, job 则完成整个工作流的控制. 目前 Kettle 最新的版本号是 3.2.0. 它主要包括四部分, 分别为 Chef, Spoon, Kitchen, Pan.

SPOON 允许你通过图形界面来设计 ETL 转换过程(Transformation). 例如, 从一个 SAP 系统抽取信息, 并把这些信息存储到一个文本文件里的转换任务如图 3 所示.

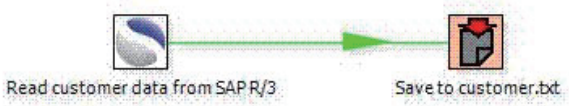

图 3 SPOON 数据转换图

PAN 允许你批量运行由 Spoon 设计的 ETL 转换(例如使用一个时间调度器). 它是一个数据转换引擎, 负责从不同的数据源读写和转换数据. Pan 是后台执行的程序, 没有图形界面.

CHEF 允许你创建作业(Job). 下面图 4 是一个 Chef 的作业图:

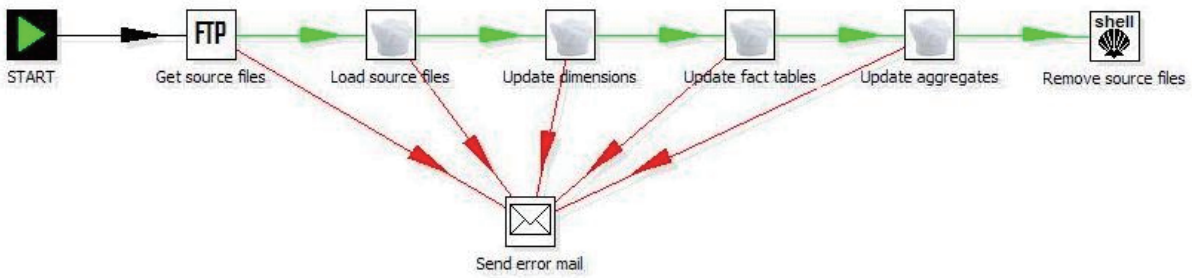

图 4 Chef 作业图

Job 与 Transformation 的差别是: Transformation 专注于数据的 ETL, 而 Job 的范围比较广, 可以是 Transformation, 也可以是 Mail、SQL、Shell、FTP 等等, 甚至可以是另外一个 Job.

KITCHEN 允许你批量使用由 Chef 设计的作业(例如使用一个时间调度器). 它是一个作业执行引擎, 用来进行转换, 校验, FTP 上传. KITCHEN 也是一个后台运行的程序.

3.3 数据仓库模型设计

(1) 确定主题

数据仓库设计首先应明确其主题, 主题的确定必须建立在现有联机事务处理(OLTP)系统基础上, 否则按此主题设计的数据仓库存储结构将成为一个空壳, 缺少可存储的数据[6].

但一味注重 OLTP 数据信息, 也将导致迷失数据提取方向, 偏离主题. 为此, 在模型设计过程中, 需要在 OLTP 数据和主题之间找到一个“平衡点”, 根据主题的需要完整地收集数据, 这样构建的数据仓库才能满足决策和分析的需要. 根据省医疗系统需求分析, 我们决定主要是需要面向突发性、大规模、区域性的医疗分析, 通过分析这些业务数据, 对医疗中心的下一步工作决策进行调整.

(2) 确定度量

在确定了主题以后, 我们考虑要分析的技术指标, 诸如多发流行病预警之类. 它们一般为数值型数据. 我们或者将该数据汇总, 或者将该数据取次数、独立次数或取最大最小值等, 这样的数据称为量度.

量度是要统计的指标, 必须事先选择恰当, 基于不同的量度可以进行复杂关键性能指标(KPI)等的设计和计算.

(3) 确定事实数据粒度

在确定了量度之后, 我们要考虑到该量度的汇总情况和不同维度下量度的聚合情况. 考虑到量度的聚合程度不同, 我们将采用“最小粒度原则”, 即将量度的粒度设置到最小.

例如: 假设目前的数据最小记录到秒, 即数据库中记录了每一秒的交易额. 那么, 如果我们可以确认, 在将来的分析需求中, 时间只需要精确到天就可以的话, 我们就可以在 ETL 处理过程中, 按天来汇总数据,

此时, 数据仓库中量度的粒度就是“天”; 反过来, 如果我们不能确认将来的分析需求在时间上是否需要精确到秒, 那么, 我们就需要遵循“最小粒度原则”, 在数据仓库的事实表中保留每一秒的数据, 以便日后对“秒”进行分析.

在采用“最小粒度原则”的同时, 我们不必担心海量数据所带来的汇总分析效率问题, 因为在后续建立多维分析模型(CUBE)的时候, 我们会对数据提前进行汇总, 从而保障产生分析结果的效率.

(4) 确定维度及其级别

维度是人们观察客观世界的角度, 它一般包含着层次关系, 这种层次关系有时会相当复杂. 通过把一个实体的多项重要属性定义为多个维度, 可以使用户能够对不同维度上的数据进行分析比较[7]. 从系统需求分析和确定的主题可以确定需要的维度: 区域维度、级别维度、时间维度、指标维度. 通过对主题和维度的分析, 确定数据仓库框架选用星型结构, 结构如图 5 所示.

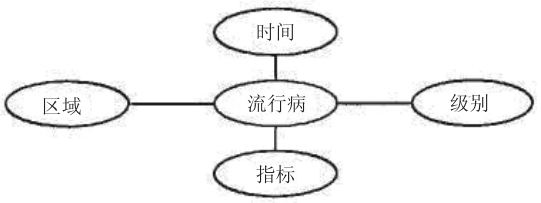

图 5 星型结构图

层次结构是使用有序层次作为组织数据的逻辑结构, 层次结构可以用来定义数据聚集. 例如, 在时间维中, 层次结构能够聚集从 month 层到 quarter 层到 year 层的数据. 一个层次结构可以用来定义导航切片路径, 建立一组结构.

级别是维度层次结构的一个元素. 级别描述了数据的层次结构, 从数据的最高(汇总程度最大)级别直到最低(最详细)级别. 级别仅存在于维度内. 级别基于维度表中的列或维度中的成员属性.

(5) 创建事实表

在确定好事实数据和维度后, 我们将考虑加载事实表.

当大量数据堆积如山时, 我们想看看里面究竟是什么, 结果发现里面是一笔笔突发事件记录, 一笔笔试验检测记录. 这些记录是我们将要建立的事实表的原始数据, 即关于某一主题的事实记录表[8].

我们的做法是将原始表与维度表进行关联, 生成事实表. 在关联时有为空的数据时(脏数据源), 需要使用外连接, 连接后我们将各维度的代理键取出放于事实表中, 事实表除了各维度代理键外, 还有各量度数据, 这将来自原始表, 事实表中将存在维度代理键和各量度, 而不应该存在描述性信息, 应符合“瘦高原则”, 即要求事实表数据条数尽量多(粒度最小), 而描述性信息尽量少.

事实数据表是数据仓库的核心, 需要精心维护, 在 JOIN 后将得到事实数据表, 一般记录条数都比较大, 我们需要为其设置复合主键和索引, 以实现数据的完整性和基于数据仓库的查询性能优化. 事实数据表与维度表一起放于数据仓库中, 如果前端需要连接数据仓库进行查询, 我们还需要建立一些相关的中间汇总表或物化视图, 以方便查询.

3.4 联机分析处理(OLAP)

联机分析处理是指对共享的多维信息的快速分析, 是以总计管理作为基础、以报表为基本骨干、以多维方阵、决策控制为组织形式的.

(1) 构建多维数据集

数据的多维分析是决策支持的支柱, 也是 OLAP 的核心, 在数据仓库中经常要进行复杂查询, 因此数据仓库立方体的有效计算已成为影响数据仓库性能的重要因素. 为了保证数据立方体的有效计算, 可以采用的策略有: 数据的聚集、数据压缩并进行近似查询、合理选择立方体的存储格式等[9].

数据仓库模型基于事实表及维表, 创建多维分析的超立方体模型, 使得用户能够方便的进行多维度的查询.

Mondrian 是开源项目 Pentaho 的一部分, 是一个用 Java 写成的 OLAP 引擎. 它实现了 MDX 语言、XML 解析、JOLAP 规范. 支持的数据库或数据仓库主要有: LucidDb、Oracle、Access、Mysql、Sybase、Ingres、Postgres、Hypersonic、Teredata 等. 它从 SQL 和其它数据源读取数据并把数据聚集在内存缓存中, 然后经过 Java API 用多维的方式对结果进行展示, 同时可以不写 SQL 就能分析存储于 SQL 数据库的庞大数据集, 可以封装 JDBC 数据源并把数据以多维的方式展现出来. JPivot 是 Mondrian 默认的表现层工具, 它是一个 JSP 自定制的标签库, 可以绘制 OLAP 分析图表. 用户可以执行典型的 OLAP 导航, 如下钻、切片. JPivot 使用 Mondrian 作为它的 OLAP 服务器但也支持 XML/A 数据

源访问. 它使用 WCF (Web Component Framework)框架 , 基于 XML/XSLT 来渲染 Web UI 组件.

3.5 数据交换平台

目前数据交换技术主要是采用两种方式实现, 一种是分布式数据库方式, 另外一种是 MQ 消息中间件方式[10].

分布式数据库方式是采用数据库自身的远程数据库连接操作技术, 实现不同服务器上数据库之间的一致性操作, 该方式的特点是传输环节少、可靠性高、一致性好, 数据交换的传送速度快, 可以做到实时同步的数据交换. 造成的缺点是对网络要求比较高.

MQ 方式是数据传输的一种工具, 它能够保证数据传输的高效性、一致性、准确性和安全性, 对网络要求稍低, 对数据传输队列有完整的管理机能并可以保证数据的惟一性, 可以实现数据自动重发的功能. 其缺点是系统传输为队列式传递, 没有分布式数据库交换方式快捷. 有关交换数据的获取、组包、拆包、处理等操作仍然需要编制程序实现, 编程工作量大, 环境配置复杂.

本系统采用开源数据交换平台 OpenEAI 作为数据共享系统与省医疗中心其他应用系统如计划免疫系统、慢性病管理系统、传染病管理系统、突发事件应急管理系统等 30 多个业务系统之间的数据交换平台. OpenEAI 提供一种方法使各企业现有的应用程序和数据库能够适用于不同企业间新的环境, 能够将业务流程、应用软件、硬件和各种标准联合起来, 在两个或更多的企业应用系统之间实现无缝集成, 使它们像一个整体一样进行业务处理和信息共享, 简化多个异构系统之间的应用集成, 提高业务效率.

OpenEAI 的应用模型如图 6 所示:

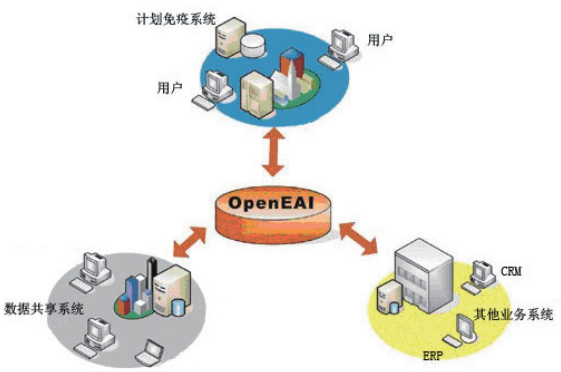

图 6 OpenEAI 应用模型图

4 系统的实现效果

本系统的具体实现则是采用 Struts2+Spring+Hibernate 的框架组合模式作为本项目的应用框架, 因为 Struts 框架具有组件的模块化、灵活性和重用性的优点, 同时简化了基于 MVC 的 Web 应用程序的开发. 而基于 IOC 和 AOP 的 Spring 框架, 能有效地组织 J2EE 应用各层的对象, 将各层的对象以松耦合的方式组织在一起. Hibernate 作为对象关系映射框架, 能提供透明的持久化, 实现了对数据访问方式 JDBC 的轻量级封装, 数据共享平台建设完毕后, 为了验证平台的共享效果, 我们主要对其进行了正确性和互操作性方面的测试, 采用如下测试策略:

(1) 执行新增数据、删除数据、修改数据脚本, 新增、删除、修改各系统数据表. 用于测试的数据应考虑正常和异常情况.

(2) 运行 OpenEAI 中间件, 对数据库表进行操作, 处理并上传数据.

(3) 通过 PL/SQL Developer 查看新增数据、删除数据, 修改数据的完整性和准确性, 数据同步的功能性, 对比同步前后数据的一致性.

① 互操作性

对数据共享平台的互操作性进行了测试, 得到的测试结果如表 1 所示:

表 1 互操作性测试

功能要求	测试结果
主系统与规定系统的交互结果正确	通过
检查子功能之间的交互正确	通过
检查数据的可交换性 , 正确实现接口数据格式	通过

② 正确性

对数据共享交换平台的正确性进行了效果测试, 得到了测试结果如表 2 所示:

最终测试结果表明, 该数据共享平台实现的效果达到了设计目标的要求, 后来的事实也证明, 该数据共享平台在试运行期间, 确实体现了良好的高可靠性、高可用性以及高可维护性.

5 结束语

本文在研究过程中虽然取得了一定的成绩, 但是我们认为还需要在以下几个方面做进一步的研究:

(下转第 76 页)

General Meeting, 2010 IEEE .Minneapoils, MN: IEEE. 2010. 1–6.
5 李俊娥,罗剑波,刘开培,周洞汝.电力系统数据网络安全性设计.电力系统自动化, 2003,27(11):56–60.
6 吴德州,武君胜.面向电力系统的分布式实时数据库设计.科学技术与工程,2008,8(4):929–934.
7 胡喜.支付宝三年光棍节高可用系统架构的演变 .http://wenku.baidu.com/view/91cb072d58fb770bf78a558b.html. 2012-09–08.
8 Alexandros G. Dimakis, P. Brighten Godfrey, Yunnan Wu, Martin J. Wainwright. network coding for distributed storage systems. IEEE Trans. on Information Theory, 2010, 56(9): 4539–4551.
9 罗志明,张大华,王电钢,常健.电力分布式云存储关键技术研究.2012 年电力通信管理暨智能电网通信技术论坛论文集.北京:2013.314–318.
10 张国良,丁岳伟.一种共享 IP 流记录分布式平台.计算机系统应用,2011,20(6):38–43
11 Buyya R, Yeo CS, Venugopal SK, et al. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility . Future Generation Computer Systems, 2009, 25(6): 599–616.
12 金弟,庄锡进,杨俊.存储虚拟化在石油物探的应用.计算机系统应用,2012,21(1):13–16,76.
13 杨燚,李善金,孙禹.电力通信网管理系统的安全防护和系统灾备研究.信息通信,2012,(6):187–188.

(上接第 43 页)

表 2 正确性测试

功能模块		功能要求	测试结果
数据传输模块	实时传输	提供当源表数据进行部分操作时, 实时修改目标数据的功能.	通过
	定时传输	提供间隔一定时间, 对源表和目标表的数据进行一次同步功能.	通过
	可靠性传输	提供保证数据传输的可靠性和确保数据的完整性的功能.	基本通过
数据校验模块		提供检查数据是否符合相应的标准, 对不符合的数据进行备份, 并记录错误原因的功能.	基本通过
主键生成模块		提供利用主键表生成目标表的主键, 使符合统一编码规则的功能.	通过

(1) 继续完善数据仓库的 ETL 操作, 建立一个比较完整的数据仓库, 实现一个功能齐全的医疗数据分析系统;

(2) 将来打算结合多种算法, 如将关联规则、模糊算法、实例推理算法综合运用到医疗数据库中, 能够比较充分、全面对大量医疗数据分析, 从而挖掘出的“知识”要比采用单一算法的挖掘结果丰富.

参考文献

1 徐冠华.J2EE 架构设计.电子计算机,2009,(1):25–32.
2 彭木根.数据仓库技术与实现.北京:电子工业出版社,2010, 10–15.
3 Inmon WH. Building the Data Warehouse (3rd Edition). John Wiley& Sons, 57Inc, 2002, 21–30.
4 Trujillo J, Palomar M, Gomez J, Song I. Designing data warehouses with OOConceptual models. IEEE Computer, 2001, 34(12): 66–75.
5 樊明辉,陈崇成,涂建东.医疗科学数据仓库及 WEB 联机分析处理的设计与实现.福州大学学报,2009,32(5): 163–170.
6 Chaudhuri S, Dayal U. An overview of data warehousing and OLAP technology. SIGMOD Record, 2008, (26): 23–28.
7 王燕萍,夏琳,李黎明.数据仓库实现过程中的关键技术,电子计算机,2009,(3):30–34.
8 Agrawal R,.Srikant R. Fast algorithms for medical association rules in largedatabase. Proc. 20th VLDB, 2010, (9): 487–499.
9 杨胜,孙翱.OLAP 技术的发展新动态.计算机应用与软件, 2011,20:20–21.
10 左建,陈警.数据共享服务的实践与应用.先进制造与数据共享国际研讨会议论文集.北京.2010.633–638.