

基于.NET 架构的产品结构及物料管理系统^①

代乾坤¹, 仲梁维¹, 屈年凯²

¹(上海理工大学 机械工程学院, 上海 200093)

²(上海市离心机械研究所有限公司, 上海 200231)

摘 要: 当前企业信息化建设过程中一个重要方面就是实现生产管理信息的高度集成, 而这这就要求各相关部门能够实现及时的信息传递和资源共享, 以达到提高生产效率, 降低生产成本的目的。根据某制造企业产品数据管理繁杂, 各部门之间数据不统一, 以及已有数据得不到重复利用等问题提出了基于 B/S 模式和微软.NET 架构体系的产品结构及物料管理系统的解决方案, 重点讨论了系统的体系结构、功能以及设计与实现方法。

关键词: B/S; LINQ; 架构; 产品结构; 物料管理

Product Structure and Material Management Systems Based on NET Framework

DAI Qian-Kun¹, ZHONG Liang-Wei¹, QU Nian-Kai²

¹(Department of Mechanics, University of Shanghai for Science and Technology, Shanghai 200093, China)

²(Shanghai Institute of Centrifugal Machine Co. Ltd, Shanghai 200231, China)

Abstract: An important section of an enterprise's CIMS is to integrate manufacturing and management information of a workshop firmly, which demands the rapid information passing and resources sharing in different departments, and workshop management to improve the work rate and lower the cost. This paper mainly mentions a solution of product structure and material management system based on browser/server (B/S) and Microsoft .NET framework system, which resolves the problem of Product data management is complicated, data are not uniform among the various departments, as well as existing data are not re-use and other issues. Its structure, functions and key developing approaches are focused in this paper.

Key words: B/S; LINQ; framework; product structure; materials management

先进技术的运用使得企业的开发周期越来越短, 然而, 信息量却在成倍的增加, 如何更加有效地进行信息的收集、统计和交换成为人们日益关注的问题^[1]。随着企业规模的不断扩大和经营范围的持续扩展, 数据共享、重复利用与各信息相互孤立之间的矛盾日趋明显。通常存在的主要问题有: (1)由于信息的不共享, 企业的知识财富往往变成由个人保管; 人员、岗位的变动, 导致产品数据丢失; (2)纸介质的归档数据因为疏漏、错误等原因导致对归档数据查询困难。(3)对产品的变更设计过程控制不足, 易造成设计数据混淆。(4)财

务人员、相关主管领导由于对产品结构、物料等信息掌握不全面, 从而缺乏对企业成本、经营状况的有效控制。

因而有必要充分利用企业内联网 Intranet 的丰富资源, 建立一个基于 Web 技术的物料管理系统, 以期更好地共享资源, 提高劳动效率, 为企业的信息化建设创造良好的先决条件^[2]。基于此, 在.NET 体系结构下采用动态服务器网页 ASP.NET 技术, 为某离心机制造企业开发了基于 B/S 三层网络体系结构的产品结构及物料管理系统, 实现了产品数据和物料信息在

① 基金项目:上海市教委重点学科建设项目(J50503)

收稿时间:2012-02-13;收到修改稿时间:2012-03-13

内部的高度共享和集成。

1 系统结构及其工作流程

1.1 系统体系结构

系统整体采用浏览器/服务器(B/S)模式。在数据库服务器一端以 Windows Server 2003 和 SQL Server 2008 数据库系统为核心, 为整个系统提供信息服务和数据服务; 以由 Visual Studio.NET 开发出的组件为中间层, 这些组件提供经过封装的各种功能支持; 由 ASP.NET 开发前端的 Web 页面。浏览器/服务器之间采用总线型网络拓扑结构, 用 TCP/IP 协议作为基本的网络通讯协议。

1.2 系统工作流程

物料管理系统的工作流程为: 用户通过浏览器访问 Web 服务器中的系统主页面进行登录, 系统验证用户身份后, 根据用户角色和权限进入相应操作页面。用户可进行各种操作, 如查询、浏览、录入、修改、删除。操作指令和数据经 Web 服务器访问数据库服务器上相关数据库后, 由数据库服务器将处理结果返回给 Web 服务器, Web 服务器再将处理结果生成 HTML 页面返回到客户端浏览器。具体模块见功能分布图和业务流程图:

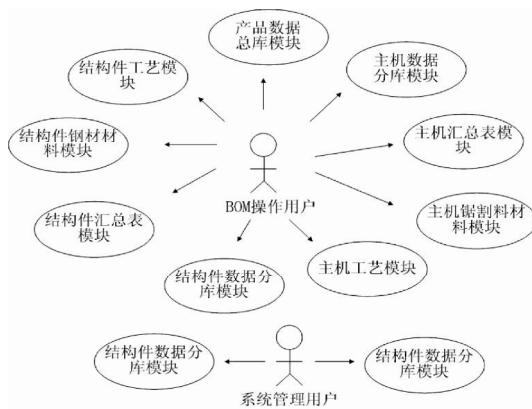


图 1 物料管理系统模块分布图

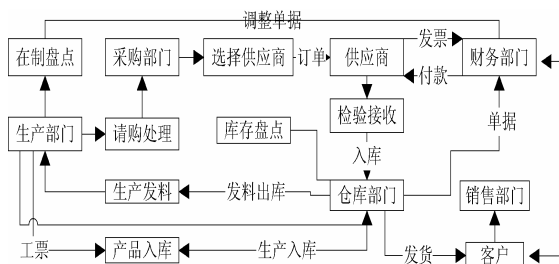


图 2 物料管理信息系统业务流程图

2 系统实现的关键技术

2.1 服务器端应用程序的开发技术

基于 Web 技术的网络体系结构目前主要有客户机/服务器(C/S)和浏览器/服务器(B/S)两种模式^[3]。C/S 模式是一种典型的 2 层结构, 客户机部分负责执行前台功能, 而服务器部分执行后台服务。这种模式具有专用性和交互性强、网络通讯量低以及速度快等优点, 但存在着客户机负担重、移植困难、系统维护复杂、升级麻烦和用户界面风格不一等的缺点。

B/S 模式具有 3 层逻辑结构, 是 C/S 模式的一种扩展。相对于 C/S 结构而言, 3 层的 B/S 体系结构是把原来在客户机一侧的应用程序模块与显示功能分开, 将它放到 Web 服务器上单独组成 1 层, 而客户机上只需安装单一的浏览器, 这样客户机的压力大大减轻了, 把负荷均衡地分配给了 Web 服务器, 从而克服了 C/S 模式 2 层结构负荷不均的弊端, 因此具有易于维护和升级、用户界面具备一致性和友好性、操作使用方便、应用程序开发与管理成本低等优点。

B/S 模式信息管理系统的关键技术在于设计交互式的动态 Web 页面以及通过 Web 访问数据库, 以实现数据库信息发布的开放性、交互性和实时动态性^[4]。因此, 服务器端应用程序开发的关键是 Web 服务器与数据库服务器的连接, 以使用户得到可靠的信息, 并对数据库进行有效的存取和管理。

本系统就是基于 B/S 数据访问模式架构的系统。采用三层软件架构, 层次清晰明了, 并且数据访问层采用了 LINQ 的方式, 大大提高了开发效率。在软件的架构设计中, 分层结构是最常见, 也是最重要的一种结构。微软所推荐的分层结构一般分为三层, 从下至上分别为: 数据访问层、业务逻辑层、页面表示层^[3]。这种分层结构如图 3 所示。将各个层次加以细化, 形成了如图 4 所示的系统架构^[4-6]。

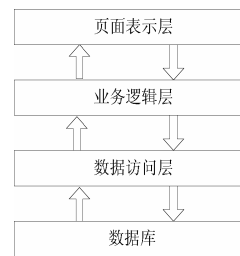


图 3 分层结构图

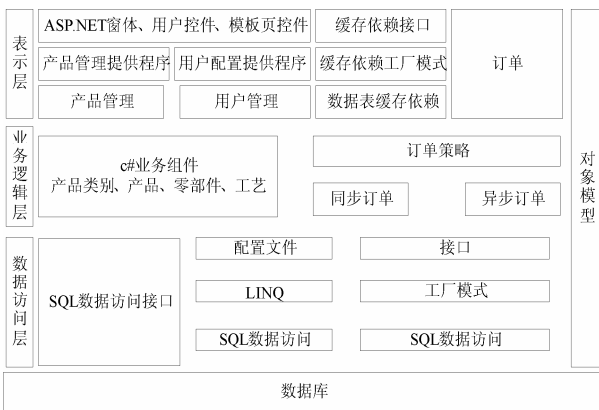


图 4 系统架构图

2.2 LINQ 技术

LINQ 是 Language Integrated Query 的缩写，翻译成中文就是语言集成查询。LINQ 是一系列的变成接口，借助于 LINQ 技术，可以使用一种统一的方式查询各种不同类型的数据。LINQ 是微软公司在 Visual Studio 2008 和 .NET Framework 3.5 版本中一项突破性的创新，它在对象领域和数据领域架了一座桥梁^[7-9]。

显然，关系型数据库是数据存储的理想选择，但是关系型数据库是根据关系模型以记录的方式来存储数据的，而不是以对象的方式来存储的，为了将程序数据保存到数据库中，开发者不得不在对象模型和关系模型之间进行转换，而为了使用存储在数据库中的数据，同样也需要在对象模型和关系模型中进行转换。这是一个模式化但是容易出错的过程，为了节约开发者的精力，许多公司推出了能够自动在对象模型和关系模型之间进行转换的组件，这些组件统称为 ORM 框架(如 Hibernate)。LINQ to SQL 是微软公司针对 SQL Server 数据库推出的 ORM 框架，是针对 SQL Server 数据库的应用程序编程接口(API)。在目前阶段，LINQ to SQL 只支持微软公司的 SQL Server 数据库。

图 5 是本系统数据库主要表的 ORM 映射关系图，数据字段变为类的属性，可以通过 LINQ 查询得到需要的数据集。在系统设计中，我们还为每个表建立了一个数据库映射类和一个数据库操作类，通过这些类，可以通过 get、set 方法，与映射关系连接，方便的存储于读取数据。真正的实现了数据存、取的对象化。以 bi_product(产品表)为例，数据库映射类如 6 所示：

数据库表操作类命名 productDB.cs，其中根据

categorycode 查询 product 表内容 LINQ 语句如下：

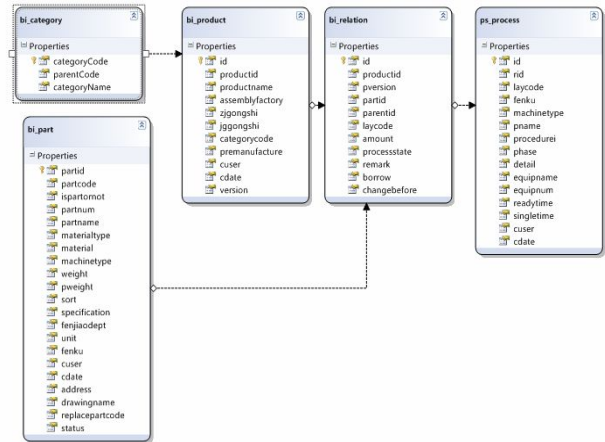


图 5 系统中主要表 ORM 映射关系图

```
private static PbmMapDataContext db = new
PbmMapDataContext();
public static List<Product>getIdandProductname
(string categorycode)
{
    var query = from product in db.bi_product where
product.categorycode == categorycode orderby
product.id ascending select product;
    List<Product> result = new List<Product>();
    foreach (var t in query)
    {
        result.Add(FillRecord(t));
    }
    return result;
}
```

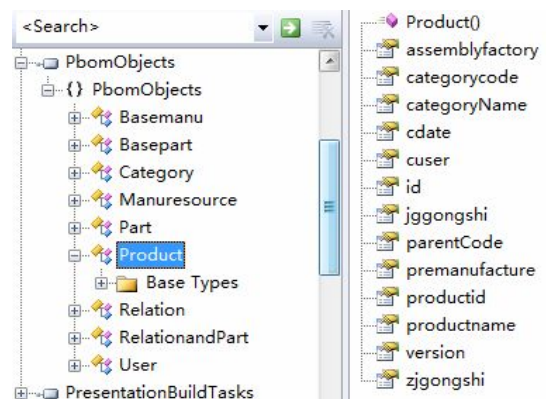


图 6 数据库表映射类

2.3 WEB 打印技术

ActiveX 控件用于 Web 打印的方法是将控件嵌入 Web 页面中, 用户通过浏览器访问该页面。将主页中的 ActiveX 控件下载。并在用户的浏览器上运行, 完成打印的功能。此时浏览器就是这个控件的容器。控件下载一次后就驻留在用户本地机上, 下次再访问相同的 ActiveX 控件时, 不用重复下载该控件^[10,11]。

在 HTML 页面中通过使用<OBJECT>标志表示对 ActiveX 控件的引用。浏览器通过在<OBJECT>标志中指定的参数特性来进行控件的下载和注册。下面是一个嵌有 ActiveX 控件的 HTML 代码示例:

```
<OBJECT CLASSID=" clsid:31291E80-728C-11
CF-93D5-0020AF99504A"
CODEBASE=" http://www.Mysite.com.cn/ocxdir/
MSCHART.CAB#version=1,0,0,0"ID=chart1 WIDTH
=400 HEIGHT=200 ALIGN=center></OBJECT>
```

在<OBJECT>标志中的特性参数的意义如下:

CLASSID 是该控件的 UUID, CODEBASE 指定 ActiveX 控件存放的 URL 地址, CLASSID 是一组独一无二的数字, 是 ActiveX 控件的 ClassID 号。在注册时, 就是通过控件的 CLASSID 来完成控件的注册。当客户浏览嵌有 ActiveX 控件的页面时, 客户端首先在本地的注册表中查找该 CLASSID。如果此 CLASSID 存在, 则不必下载, 而直接使用本地已注册的 ActiveX 控件。如果此 CLASSID 没有在本地上注册。浏览器则按照 CODEBASE 指定的路径搜索该控件并下载。客户端打印的实现过程如图 7 所示。

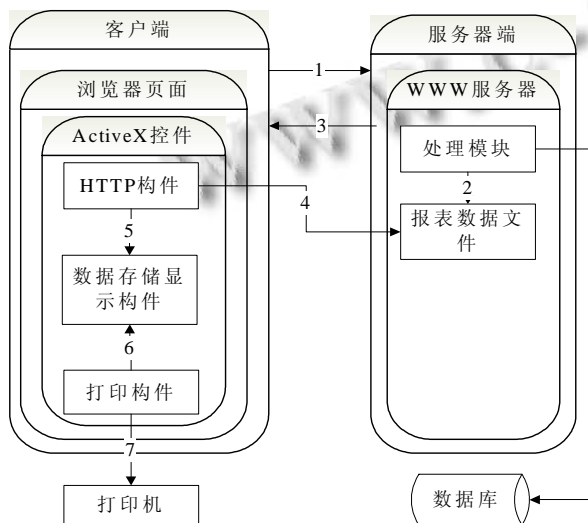


图 7 客户端打印的实现过程

采用 HTTP 协议的 Get 方法下载数据文件的基本过程如下: 图中第一步在 HTML 页面中的<OBJECT>标志里读取 PARAM 标记中的 urlAdd 属性, 获得数据文件的 URL 地址; 图中第二步将读取到的数据文件 URL 地址传送给 ActiveX 里的 NMHTTP 控件; 图中第三步 NMHTTP 根据数据文件 URL 地址调用 HTTP 协议里的 GET 方法下载数据文件存储在本地端; 图中第四步 ActiveX 控件对数据文件进行解析, 将数据添加入数据表格(Grid)中。

通过以上操作, ActiveX 控件的 HTTP 模块从 www 服务器中下载了数据文件(第四步), 并对其进行解析, 并输入到控件的数据存储显示模块里, 用以存储和显示(第五步)。当用户按下打印按钮时, ActiveX 控件调用其打印模块(第六步), 打印模块连通本地打印机, 完成数据的打印输出(第七步)。

3 系统实现

系统基于 SQL Server 数据库系统和 .NET 技术平台开发完成, 按其功能主要分为系统管理、数据管理、执行控制管理和综合查询四部分。

系统管理模块主要功能为新用户的添加和相应角色的赋予、简洁明了的用户查询和高效的组合查询、用户的停用、启用和删除等。

数据管理模块的主要功能为物料基础数据及产品结构 BOM 的导入、编辑、删除和图纸的上传等。



图 8 系统结构界面截图

执行控制模块的主要功能为产品数据、主机数据、主机工艺、结构件数据、结构件工艺的自动计算、自动更新、自动匹配, 确定传阅范围及相关操作人员、根据不同的角色和权限确定相关的可操作数据和物料信息的统计和打印等如图 9 主机零部件总库的执行控制流程。

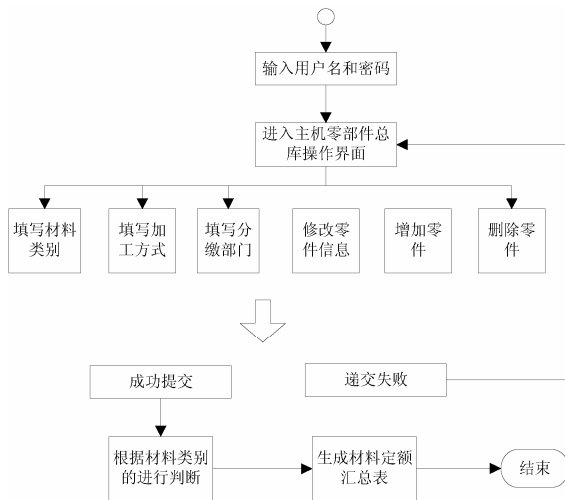


图 9 主机零部件总库的执行控制流程

综合查询模块为所有用户所共享，根据用户权限的不同，系统将自动确定所能执行的操作，用户只需通过选择或输入多种查询条件提交后便可得到所要查找的数据并能通过超链接提供数据相关的操作，如图 10 产品结构 BOM 的检索结果。

序号	代号 (图号)	名称	数量	材料	重量	备注	册量数量		册量重量		分类类型	材料类别	加工类别	工艺性	是否发图
							单台	总数	单台重	总重					
2.1	序1	螺帽	4	Q345B	543.3	3803.1	28	28	15212.4	15212.4	结构件	螺帽	加工工序	冲压	无零件图
2.2	序2	螺帽	1	Q345B	543.3	543.3	4	4	2173.2	2173.2	结构件	螺帽	冷电	冲压	无零件图
2.3	序3	螺帽	1	Q345B	381.0	381.0	4	4	1524.0	1524.0	结构件	螺帽	冷电	冲压	
2.4	序4	螺帽	1	Q345B	381.0	381.0	4	4	1524.0	1524.0	结构件	螺帽	加工工序	冲压	
2.5	序5	螺帽	1	Q345B	265.0	265.0	4	4	1060.0	1060.0	结构件	有色件	冷电	冲压	
2.6	序6	螺帽	1	Q345B	265.0	265.0	4	4	1060.0	1060.0	结构件	螺帽	冷电	冲压	无零件图
2.7	序7	螺帽	1	Q345B	200.0	200.0	4	4	800.0	800.0	结构件	紧固件	冷电	冲压	无零件图
2.8	序8	螺帽	1	Q345B	232.0	232.0	4	4	928.0	928.0	结构件	螺帽	加工工序	冲压	发图
设计日期	校对日期	审核日期	批准日期	分整日期											

图 10 产品结构数据的检索结果

4 结论

本项目针对某离心机制造企业对产品结构和物料

(上接第 72 页)

7 Schwenk H, Bengio Y. Boosting Neural Neural Networks. *Neural Computation*,2000,12(8):1869-1887.
 8 陈军,徐蕾.用一种改进的蚁群聚类算法进行网络入侵检测. *沈阳航空工业学院学报*,2010,(1).
 9 Giacinto G, Roli F. Intrusion Detection in Computer

的管理需求，建立了计算机辅助管理的物料管理系统，该系统以计算机网络、数据库系统为技术基础，现代管理理论和方法为指导，覆盖设计产品结构及物料的信息流通的全过程，最大限度地利用企业的人、财、物、设备、技术和信息资源，加快产品数据流通的效率，提高了数据流通的准确程度，同时降低工作人员的劳动强度，最终以提高企业的经济效益和市场竞争能力为目标。

参考文献

1 Henrik J, Peter A, Kjell O. A system for information management in simulation of manufacturing processes. *Advances in Engineering Software*,2004,35:10-11.
 2 陈伟忠.珠江电信 PDM 系统需求分析及实施.北京邮电大学,2008.
 3 于晓,仲梁维等.基于产品的 BOM 自动生成方法.精密制造与自动化,2006,(3).
 4 贾颖莲.基于.NET 平台的产品数据管理技术的研究与实现,2005.
 5 Yeh RT. Software and Database Engineering:Towards a Common Design Methodology.Issue in Data Base Management,1979.
 6 张婧.基于产品结构与管理配置管理的 PDM 系统研究与开发,2006.
 7 王保健.ASP.NET 网站建设专家.北京:清华大学出版社,2005.
 8 章立民.ASP.NET 开发实践范例宝典(使用 C#).北京:科学出版社,2010.
 9 解本巨,李宗颜,宫生文.LINQ 从基础到项目实践,2010.2.
 10 祖晓东.基于 ASP.NET2.0 实现 WEB 打印方法的探讨.
 11 董培征,杨学良.ActiveX 技术在 Web 应用中实现本地端打印.微计算机应用,2001,22(4):202-204.