# 基于中间件技术的制造业需求管理的设计与实现

## Designing and Implementation of Manufacturing Demand Management on Middleware

于海燕　崔君成　王 同　宁殿双　(大庆油田有限责任公司 163514)

**摘要:** 本文通过八个制造项目抽取出制造业需求管理系统模型,并且描述了如何运用中间件技术实现制造业需求管理。本文设计出系统的功能模块和数据库表结构以及表之间的关系;根据 J2EE 三层架构,给出了需求管理软件实现的静态技术框架和动态技术框架。

**关键词:** 需求管理 中间件 J2EE MDS

## 1 项目的背景、目标及实现思路

在制造业务上,尤其在需求管理上,根据我国目前的国情,没有固定的模式。对于西方国家的需求管理,例如 SAP 的管理,很多地方也不符合我国的国情。制造业需求管理模块是根据 ERP 的核心管理思想,在已经开发并投入使用的八个中国的制造项目的基础之上综合出的需求管理,比较符合中国国情,并且有一定的通用性。这八个制造项目分别由爱浪集团、山东鲁峰、河北亚大、南京浦镇、西安西磁、广东金发、日立赛格、深圳华浮等单位承担。

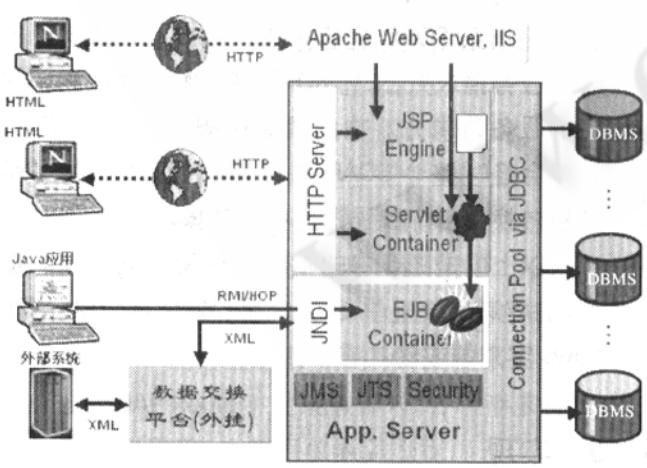


图 1

软件实现的总的思路是按照软件工程的理论,进行模块的需求分析,得出该模块的 E－R 图和初步的数据流程图;进行模块的概要设计和详细设计,得出该模块的数据流程图、功能模块图和数据库模型;进行编码和单元测试;进行集成测试和系统确认测试。

在软件实现技术中,我们采用了中间件技术。参照 J2EE 应用服务器的应用框架(框架图如图 1),构建系统的中间层服务器——应用服务器。

该系统的应用服务器是通过 JDBC 和该系统的数据连接起来的。

该模块的用户与应用服务器之间的交互是:通过 WEB 服务器中的 Servlet 容器,Servlet 容器再与应用服务器的 EJB 容器进行业务数据的处理,然后将结果通过 Applet 返回给用户。

应用服务器的引入简化了企业应用系统的开发、部署和维护。它使开发人员专著于用程序去实现业务逻辑,同时依赖于各种各样的后端服务程序实现基础结构和客户端应用程序,以便用户进行交互操作。

该系统的开发分成三个层次:中间件(应用服务器)的开发;相应的部署工具的开发;业务逻辑实现的开发。在 EJB 容器中,我们使用 Session Bean。

## 2 制造业需求管理的总体功能

制造业需求管理的总体功能模块图(如图 2 所示)。需求管理共分成三个子模块:基础设置、需求获取和需求计划。

## 3　制造业需求管理的实现

### 3.1　动态技术架构

BO：用来完成特定业务领域内所涉及的业务功能，它通过操纵“查询对象”、“数据管理对象”这两类数据处理对象来完成客户端提交的业务逻辑请求。

辅助类；在 WEB 服务器中，主要存放 UI、VO 以及其他辅助类。系统的静态技术架构（如图 4）。

### 3.3　实现需求管理业务层的方法

业务（bs）层的基础类主要指 nc. bs. pub 这个包。业务层基础类框架的设计主旨是为业务代码屏蔽各种 EJB Server 的差别，当系统运行在不同的 EJB Server 上

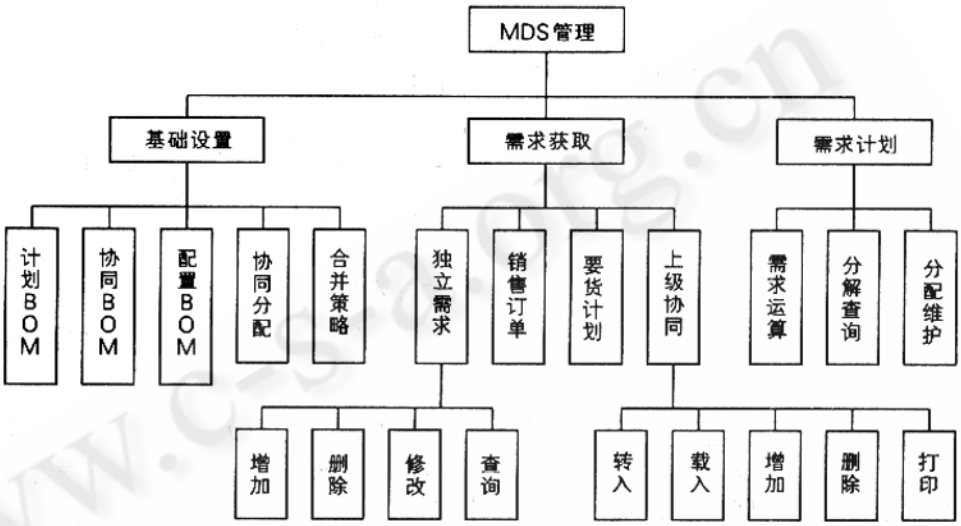


图 2

DMO：（1）、封装对数据库的操作。这样 BO 类中的代码集中于完成业务逻辑，不包含数据库操作的具体代码。（2）、DMO 类中方法的方法名应反映该方法对数据库所执行操作的确切含义。

VO：包装代表业务含义的一组数据，负责在系统各层之间传递业务数据。

用户在浏览器用户界面（Applet）选择和输入条件，点击按钮后（此时，在对应的 WEB 服务器上生成相应的 Servlet 容器），首先生成一个 VO 代码，然后将该 VO 代码传递用 HTTP 协议到 WEB 服务器上，VO 类包括对应字段的 get 和 set 方法。在 WEB 服务器上，通过容器 Servlet 调用应用服务器上的 BO Session，BO 再通过 VO 调用相应的 DMO 中的 get 和 set 方法，在数据源中设置的相应的驱动程序（API 函数）来解释这些方法，并取到相应的数据记录集，放入到该 VO 中。然后 Servlet 服务器将 VO 结果返回到用户的浏览器界面上。

### 3.2　静态技术架构

在数据库服务器中存放 DBMS，包括表、视图和存储过程；在应用服务器中存放 BO、DMO、VO 以及其它

时，只需为该 EJB Server 提供一套业务层框架代码，而不需要修改业务代码。

业务层主要由 BO 和 DMO 类构成，界面层与业务层的交互必须通过 BO 类进行，如图 3 -4 所示。但业务层不仅限于 BO 和 DMO 这两种类，可能还需要各种辅助类，这些辅助类可以继承任何 Java 类，它们辅助 BO 类实现业务功能。

（1）数据管理对象（DMO）。负责对数据库表的增、删、改和查询原子操作，完成业务数据的持久化工作。每个方法中包含且仅包含对数据库的一个操作。public abstract class DataManageObject 是所有 DMO 对象的基类。每个 DMO 类都继承 DataManageObject。DMO 类中每个方法完成一个数据库操作。

通常 DMO 类中应包含 insert（）、delete（）、update（）方法。还可以包括其它的查询方法。对一些特殊的继承类，如处理参数设置的 DMO 类，可能不需要 insert（）和 delete（）方法。

在 DMO 类中，数据库联结必须通过 getConnection（）方法获得，不允许直接使用 JNDI 查询。

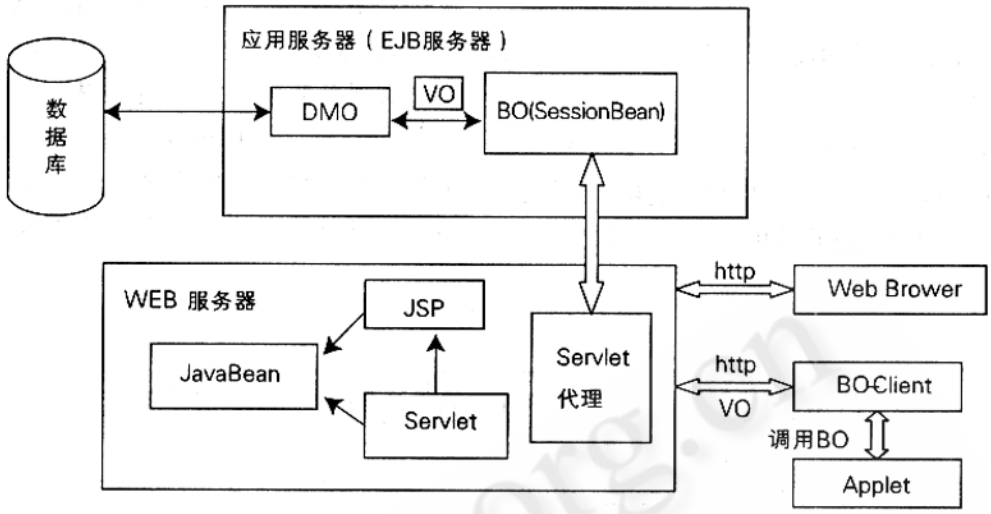


图 3　系统的动态架构

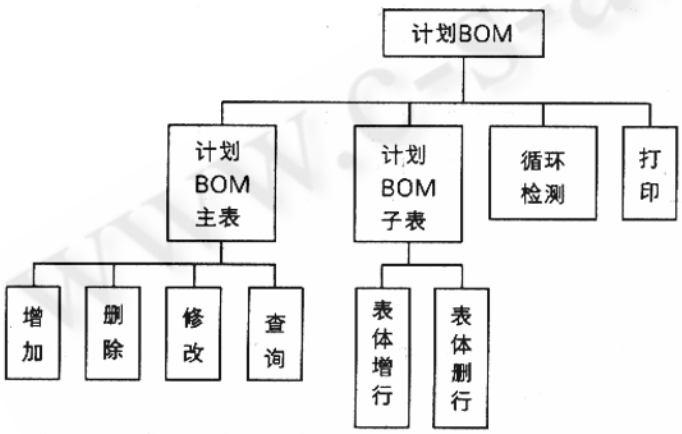


图 4　系统的静态架构

应在 DMO 类每个方法的获得 Connection、PreparedStatement 两种数据库资源，并在方法的结束位置释放数据库资源。

在 DMO 类的一个方法中向数据库插入（insert）数据时，应使用 getOID（）方法获得一个自动产生的库表主键值。

通常 DMO 方法的参数应是 String、Integer 等数据类型，而不是一个 VO 对象。

（2）业务对象（BO）。EJB 中的 SessionBean 用来实现特定业务领域内所涉及的业务功能。public abstract class BusinessObject 是所有 BO 对象的基类，每个 BO 类都继承 BusinessObject 类。BO 对象通过操纵 DMO 对象完成业务逻辑。

（3）数值对象（VO）。包装代表业务含义的一组数据，负责在系统各层之间传递业务数据（如图 5 所示）。通常一个 VO 对应一个数据库表，但这不是绝对的。事实上，一个 VO 对象可以对应多个数据库表，也可以对应一个数据库表的部分字段。public abstract class ValueObject 为所有 VO 对象的基类。

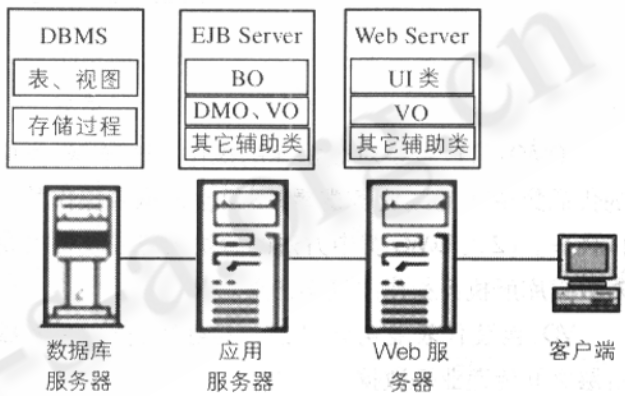


图 5

（4）为 BO 类生成调度码和部署 EJB。当设计完 BO 类后，需调用 NC EJB 开发工具集生成和部署代码，生成瘦客户端，供客户端调用 BS 端的代码。

### 3.4　性能优化方法

（1）制造业需求管理数据库物理规划原则。为了避免 I/O 的冲突，该系统在设计数据库物理规划时应遵循了以下原则（以 ORACLE 举例）：

① table 和 index 分离：table 和 index 应该分别放在不同的 tablespace 中；

② Rollback Segment 的分离：Rollback Segment 应该放在独立的 Tablespace 中；

③ System Tablespace 的分离：System Tablespace 中不允许放置任何用户的 object，不允许放置。（mssql 中 primary filegroup 中不允许放置任何用户的 object）；

④ Temp Tablesace 的分离：建立单独的 Temp Tablespace，并为每个 user 指定 default Temp Tablespace；

⑤ 避免碎片：但 segment 中出现大量的碎片时，会导致读数据时需要访问的 block 数量的增加。对经常发生 DML 操作的 segemeng 来说，碎片是不能完全避免的。所以，我们应该将经常做 DML 操作的表和很少发生变化的表分离在不同的 Tablespace 中；

⑥ 连接 Table 的分离：在实际应用中经常做连接查询的 Table，可以将其分离在不同的 Taclespace 中，以减少 I/O 冲突；

⑦ 使用分区：对数据量很大的 Table 和 Index 使用分区，放在不同的 Tablespace 中。

在实际的物理存储中，建议使用 RAID。日志文件应放在单独的磁盘中。

合理的数据库物理规划，减少了 I/O 的冲突，因此也在很大程度上保证了系统的性能。

（2）SQL 语句的写法。SQL 优化的原则是将一次操作需要读取的 BLOCK 数减到最低。调整不良 SQL 通常可以从以下几点切入：检查不良的 SQL，考虑其写法是否还有可优化内容；检查子查询，考虑 SQL 子查询是否可以用简单连接的方式进行重新书写；检查优化索引的使用；考虑数据库的优化器；借助 SQL 的性能评测工具 SqlExpert，它能够给出我们的 SQL 语句在编译、运行时的资源消耗情况，并且具体到 SQL 中的各个片段。

（3）代码和数据的缓存技术。为了提高 NC－ERP 制造业需求管理的性能，采用了代码缓存和数据缓存机制。

当客户端首次连接中间应用服务器时，客户从服务器上下载 class 代码文件，并将这些文件缓存在本地的用户目录下。如在 windows2000 系统上，缓存目录是 C：\Documents and Settings\% user% \NC_CODE。当系统升级或服务器上代码更新后，该系统将下载新的代码文件。

（4）JAVA 的编码

① 从制造业需求管理子系统的动态和静态技术架构中可以看出该子系统的开发只需关注 BO、DMO、UI 的开发，程序员不必在代码中关注如何与系统底层、数据库打交道。

② 充分利用有关辅助工具，简化开发过程，同时增强了客户化程度。这种方法其实是软件复用的一种方法。基于 NC 界面统一，常用功能部件统一的思想，抽取公共部分封装成为模板，进行统一控制。NC 目前有单据、查询、帐表、打印、卡片等模板。NC 模板包括两个部分：模板数据和公用控件。模板数据用于存放于数据表中，描述模板的各个属性。例如打印模板就描述了打印格式等信息；公用控件用于程序员在代码中可使用的控件。例如单据和报表控件是一个 UIPanel，查询是一个 UIDialog。模板数据存放在数据表中的主要目的是方便用户自己定制自己的单据、查询、报表和打印。但此种方法增加了系统访问数据库的次数，在某种程度上降低系统的速度。

③ 良好的代码风格，注意代码的效率与健壮性。写类继承的层数不要太多，类中需要继承德内容不能太少；类代码大小最好不要超过两屏；类定义时，功能要明确、简洁、较小，这样有利于将来继承；使用 JAVA 性能评测工具，如 OptimizeIt，Numega，它们能够全面测试我们代码的效率，分析代码各片段对于资源的消耗，如 CPU、内存等，通过它们我们能够很快找到最耗资源的代码片段。

## 4　结束语

本文的重点一是通过八个制造项目抽取出制造业需求管理模型，二是描述 NC－ERP 需求管理是如何实现的。

下一步主要工作有两点：一是进一步规范、完善业务流程，二是改进该系统的性能。

目前，该项目已经完成。在今年上半年邢钢项目中，就是在本产品的基础之上，根据邢钢企业的具体情况，开发补充功能点而进行的。

### 参考文献

1 《精通 JAVA 核心技术》，刘小华等编著，2003.8。
2 《ERP 原理.设计.实施（第 2 版）》，罗鸿、王忠名等，电子工业出版社.2003。
3 《System Analysis and Design Methods》Jeffrey L. Whitten，Lonnie D. Bentley，Kev. 2001.05。