

基于.NET的可裁剪式多层 SmartERP 系统

赵洁^{1,2}, 肖南峰¹

(1. 华南理工大学计算机科学与工程学院, 广州 510641; 2. 广东工业大学经济管理学院, 广州 510520)

摘要: 针对中小制造企业个性化和不断变化的需求, 基于.NET 设计和实现可裁剪式多层 SmartERP 系统。该系统使用多种设计模式, 架构可裁剪, 体系内组件可插拔, 提供数据持久化、对象缓存等多种核心服务。实践证明, 该系统具有高度弹性, 能与电子商务系统无缝集成, 提供实用的企业级决策支持。

关键词: 企业资源计划; 中小制造业; 可裁剪式; 多层架构; 决策支持

Tailorable Mutil-tier SmartERP System Based on .NET

ZHAO Jie^{1,2}, XIAO Nan-feng¹

(1. School of Computer Science & Engineering, South China University of Technology, Guangzhou 510641;

2. School of Economics and Management, Guangdong University of Technology, Guangzhou 510520)

【Abstract】 To meet the individual and changing demand of the small and medium-sized manufacturing enterprises, this paper designs and realizes a multi-tier SmartERP system based on .NET. Various design patterns are used. The system can be tailored, its components can be plus in and out and it can provide flexible core services such as common data access, object cache and so on. Practice proves that it is flexible, can be integrated with electronic commerce system seamlessly and provide support for enterprise-level decision.

【Key words】 Enterprise Resource Planning(ERP); small & medium-sized manufacturing enterprises; tailorable; multi-tier framework; decision-support

企业资源计划(Enterprise Resource Planning, ERP)系统至今已经历了多个阶段, 从2层到多层架构再到可重构ERP, 近几年, 基于SOA/Web Service的ERP成为热点。本文通过研究现有的系统, 结合为一家中小汽车配件企业设计并实施ERP的经验, 对中小制造业ERP实施情况进行调研, 针对企业个性化和不断变化的需求, 重新设计和开发基于.NET的面向中小制造业的可裁剪式多层架构的SmartERP系统。

1 SmartERP 概述

相对于大型企业的系统, 中小企业ERP可根据实际需要降低复杂度, 初期可收容到对主体业务的管理。确定设计理念为: 实现面向中小制造业的系统框架, 架构多层次、可裁剪, 层次间柔性结合, 组件可插拔, 在此框架上能迅速进行订制的二次开发^[1-4]。

尽管制造行业的具体业务差别巨大, 但总是由生产、供销、管理等基本部分组成。基于此^[1,3], SmartERP系统实现了生产管理、供应链、人事、财务、基础数据等多个子系统, 主生产业务流程见图1。系统还预留了与电子商务系统无缝连接的接口。

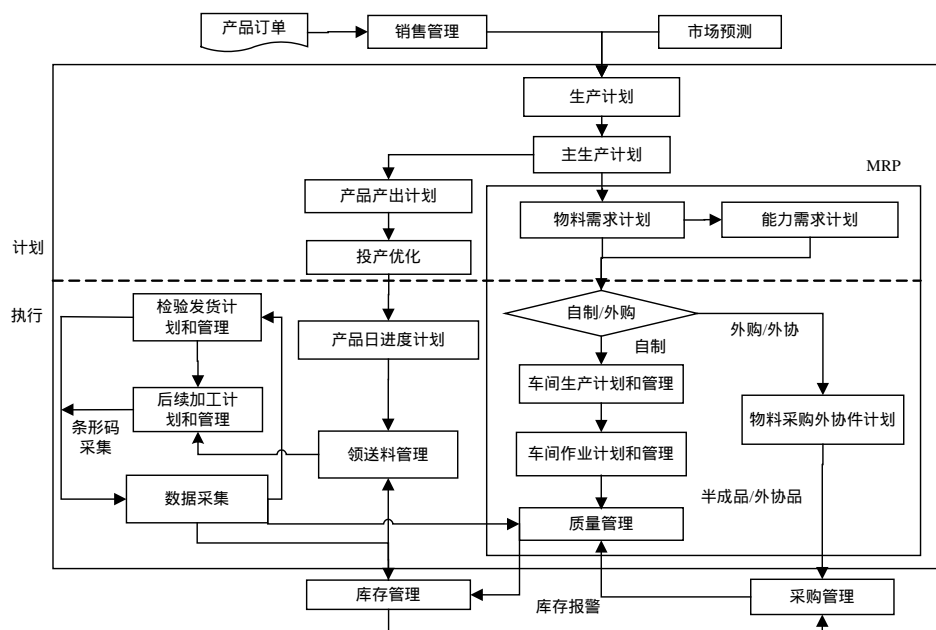


图1 主生产业务流程

基金项目: 国家自然科学基金委员会与中国民用航空总局联合基金资助项目(60776816); 广东省自然科学基金资助重点项目(36552); 广东省科技计划基金资助项目(2007B060401007); 广东工业大学青年基金资助项目(072058)

作者简介: 赵洁(1979-), 女, 博士研究生, 主研方向: 智能计算, 电子商务; 肖南峰, 教授、博士、博士生导师

收稿日期: 2008-02-10 E-mail: kitten-zj@163.com

2 SmartERP 系统的设计

本系统的构架设计^[2-4]如图 2 所示。

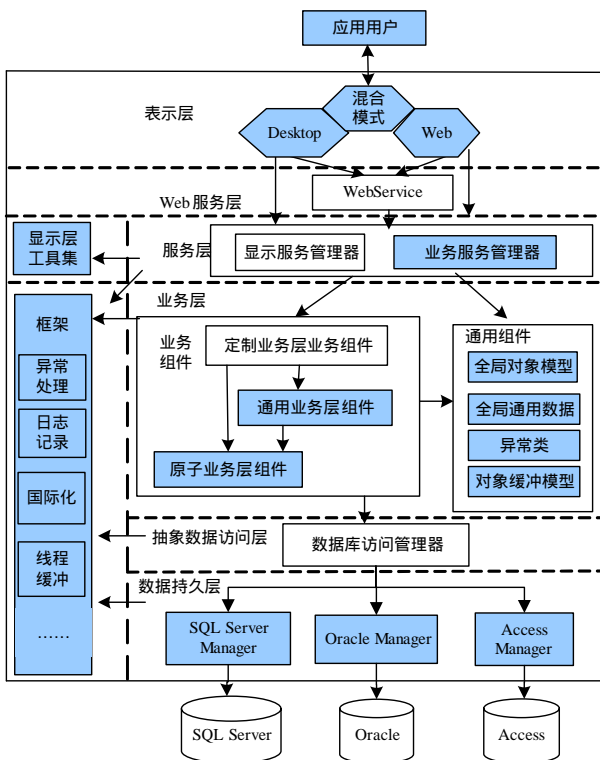


图 2 SmartERP 系统结构

2.1 多层架构设计

系统包括数据持久层(Persistence)、抽象数据访问管理层(DBM)、业务层(Business)、服务层(Service)、Web 服务层(WebService)和表示层(Display)，与其他系统的不同之处在于：

(1)增加了数据持久层。该层位于架构底层，提供对数据库及 XML 配置文件的持久化操作，且完全脱离实际业务逻辑，可作为独立构件供其他系统调用。

(2)设计抽象数据访问管理层。通过修改配置文件就可达到跨数据库平台的目的。

(3)服务层添加双层控制功能。该层中的管理器不但实现业务逻辑控制，还负责控制窗体/页面之间的跳转，使窗体/页面间的耦合度减到最低。

(4)显示层完全独立。显示层与业务逻辑彻底分离，与其他页面耦合度为零，仅与服务层关联及显示数据，不做任何逻辑处理。

(5)增加显示层工具集(DisplayUtil)。为增强显示层的独立性，工具集提供一系列自定义控件，如 SDataGridView, SComboBox，并提供对控件的常用处理功能，如 DataGridView 的动态数据绑定。

(6)设计连接子实现单据间的“推”和“拉”。在实际生产中，纸质单据一单多用，在不同部门间往返。现要求自动产生下一单据。通过设计业务连接子等组件，动态关联、生成各种单据。此处“推”是指由当前单据生成相关联的下一工序单据，“拉”是指由前一工序单据生成相关联的当前单据。当业务变更或者为新企业开发时，只需更新连接子。

2.2 可裁剪架构

当下层没有执行完毕，主程序处于等待状态，等待时间随层次的增加而增加，如图 3 所示。系统面向中小企业，在

保证扩展性的前提下，可通过裁剪某些层提高速度。

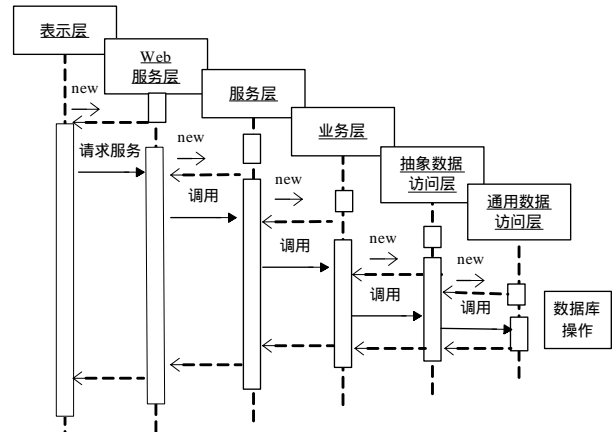


图 3 系统各层间的关系

(1)Web 服务层把服务向外界暴露，供系统扩展使用。而 Web Service 有其适用范围，不使用时可裁剪该层。

(2)当企业所使用的数据库固定、无须跨数据库平台时，可裁去 DBM 层。业务层直接调用 Persistence 层中指定数据库组件的 API。

(3)服务层可全部裁减。但不提倡裁去服务管理器，尤其对于较复杂的系统，因为这会使显示层接触过下层构件。但当企业对速度要求很高而业务逻辑又相对简单时，可以考虑将该层全部裁去。

2.3 可插拔构件

系统中多个构件以 dll 方式供调用，可灵活插拔。图 2 中深色部分是系统的固有部分，其他层次和构件均可裁剪。服务层中的控件操作和页面跳转构件可根据实际插拔。定制构件面向企业特定业务，对于不同企业，需更新这些构件。

3 SmartERP 的实现

3.1 实现平台和系统模式

实现平台选取 .NET+ C#+SQL Server，也可支持其他数据库。C/S 和 B/S 模式各具优势，基于上述框架可迅速开发出 C/S 或 B/S 或混合模式的系统。

3.2 系统架构的重要服务

(1)数据持久服务

该层提供完备的、与业务逻辑无关的数据库持久化操作功能，部分重要类描述如下：

1)IRecordObjectFactory 和 IUpdateFactory 接口：记录对象、记录更新工厂，提供数据记录对象生成和更新的规范。

2)DatabaseAssistant 类：数据库访问协助类，提供查询、更新(增、删、该)入口。

3)DatabaseManager 类：实现数据库查询、更新操作。

4)DBSession 类：封装某次数据库访问的逻辑处理，管理数据库连接、事务等。

5)QueryLogic 类：表达查询逻辑的抽象类。

6)SQLStatementGenerator 类：生成 SQL 语句。

7)TransactionAssistant 类：数据库操作管理辅助类，提供数据库事务级的处理。篇幅有限，略去类中成员描述。

(2)服务对象缓存服务

系统提供对象缓存服务，具体操作为：数据被检索后不释放，临时存于内存中，这样缓冲区中均为热点数据，能快速响应用户请求。但这可能造成数据不一致、数据丢失更新等问题。系统采用数据同步/异步更新机制解决上述问题。系

统提供全局缓存和局部缓存。

缓存区空间有限，需适时刷新缓存对象，系统采用最近没有使用算法(NRU)释放对象。缓冲池空位数量少于 boundToElimiate 个时开始释放，缓冲池空闲位置多于 boundToStopElimiate 个时停止释放，每隔 examine IntervalMills 秒检查空闲位置，数据存在内存中 TimeoutMills 秒未被使用将在下一次缓存收集时被回收。

(3)国际化服务

考虑企业产品大部分外销的情况，系统提供了国际化服务：1)创建若干不同语种的资源文件。2)在文件中添加相关记录。以中文和英文为例：添加 SmartERP.I18N.En.XML 和 SmartERP.I18N.Cn.XML 文档；文档中分别添加记录：name: product; value:camshaft 和 name: product; value:凸轮轴。3)通过修改系统配置文件即可实现动态切换语种。

(4)多种其他核心服务

框架还提供异常处理、日志记录、报表和性能监视等多种服务。

3.3 多种模式的使用

设计模式可以解决从细小乃至大规模架构级问题，系统综合使用了多种设计模式^[5]。

(1)Factory(工厂)模式。应用该模式，设计 Persistence 层中 IRecordObjectFactory 和 IUpdateFactory 接口，上层通过实现这些接口，完成特定对象模型数据的组装和更新。

(2)Reflection(反射)模式。其核心类是 System.Type。系统实现利用该模式表示层中窗体/页面的完全分离。任意窗体之间不产生联系，窗体跳转须向 SFormController 发请求，Controller 接收请求字符串后经处理构成窗体名(命名规则为 namespace+窗体名)并生成响应窗体。

综合应用 Reflection 和 Factory 模式系统实现跨数据库平台。采用 Factory 模式，DBM 中的 DBManagerFactory 提供访问数据库操作的标准接口。首先实现前述接口，并增加一个配置文件，通过修改该文件，应用 Reflection 模式，动态生成不同的 DBManager，实现跨数据库平台。

(3)Façade(门面)模式。系统中 InterService(服务层)作为整个体系服务的入口，将用户界面与各种业务逻辑分离。InterService 提供多种服务供调用，并把上层请求委派到一个或者多个下层子系统中。

(4)Dispose(处理)模式。系统遵循 .NET 标准模式处理非内存资源，实现 IDisposable 接口以释放非受控资源。Common 中的数据库事务类实现 Dispose()方法以释放资源，调用 SuppressFinalize (this)防止调用析构函数对资源重复处理，实现 Dispose(bool)以分解共同事务。

(5)Singleton(单件/单态)模式。Common 中的 DatabaseManager 需要有且只有一个实例，并提供一个全局访问点，需采用 Singleton 模式。具体实现为：使用静态初始化方法定义静态私有变量 _instance、私有构造函数及公共静态方法 GetInstance()，确保 DatabaseManager 在系统中存在唯一实例。

(6)Decorate(装饰)模式。采用该模式抽象类 QueryLogic 定义抽象方法 GetQueryString。AndQuery, NotQuery, BasicQuery 和 ORQuery 等类扩展 QueryLogic 类，生成复杂查询逻辑。

4 系统运行与测试

系统运行部分截图如图 4 所示。



(a)生产子系统主界面



(b)主生产计划管理界面



(c)非法操作提示

图 4 系统运行截图

以生产子系统为例，介绍系统的功能测试用例设计。在编码阶段进行单体测试，在测试阶段进行组合测试和总体测试，使用黑盒测试并综合使用等价类划分、边界值分析等方法，用重复递交相同事务、不按常规顺序执行等方法对关键操作进行检测。测试内容涵盖：基本功能，数据处理，状态控制，业务流程，其中，业务流程是测试重点，测试用例包括用例说明、业务流程、特殊需求、前置条件和扩展点。设计多项测试数据，例如添加 2 月主计划，计划生产 100 个链条张紧器滑块(由 1 个张紧器滑块底座、1 个链臂支撑件、2 个垫块组成)和 50 个链条张紧器(由 1 个凸轮、2 个销、1 个线圈弹簧 1、1 个线圈弹簧 2、1 个活塞组件、1 个张紧器组件组成)，测试数据如图 5 所示。

主计划单	加工单	领料单	质检单	入库单
2 月生产链条张紧器滑块 100 个； 链条张紧器 50 个	1.0 链条张紧器滑块 100 个	1.11 张紧器滑块底座(100 个)， 1.12 链臂支撑件坯(100 个)， 1.13 垫块(200 个)	1.2 链条张紧器滑块 (100 个)	1.3 链条张紧器滑块(50 个)
	2.0 链条张紧器 50 个	2.11 凸轮(50 个)，2.12 销(100 个)，2.13 曲轴(50 个)， 2.14 线圈弹簧(50 个)， 2.15 线圈弹簧 2(50 个)， 2.16 张紧器组件(50 个)	2.2 链条张紧器(50 个)	2.3 链条张紧器(50 个)
1 条记录	2 条记录	9 条记录	2 条记录	2 条记录

图 5 业务流程数据表

(下转第 30 页)