

管理信息系统开发平台的研究

付建苏

(南京航空航天大学 民航学院, 江苏 南京 210022)

摘要:管理信息系统(MIS)的开发作为计算机应用的一个重要分支,其开发平台在很大程度上决定了信息应用系统的价值和生命力。通过对管理信息系统平台的研究,采用了基于建模的方法设计和实现了信息管理系统。解决了开发和管理应用系统时遇到的大量的共性的问题,提供了一套完整的应用解决方案。对设计与开发信息管理系统有重要的意义。

关键词:模型驱动架构(MDA);建模;管理信息系统

中图分类号:TP311 **文献标识码:**A **文章编号:**1009-3044(2009)31-8609-04

A Research on the Development Platform of Management Information System

FU Jian-su

(Nanjing University of Aeronautics and Astronautics, College of Civil Aviation, Nanjing 210016, China)

Abstract: As an important branch of computer application, the platform for MIS developing, to a great extent, provides the value and vitality for information application system. Through to platform of management information system research, using modeling-based method to design and realize information management system. Solve out the common problems of encountered in developing and managing application system, and then provide a complete set of application. It has an important signification in designing and developing management information system.

Key words: MDA; modeling; management information system

近年来,由于商业运作受到 Web 的影响越来越大,许多企业和高校都要求建立自己的 Web 业务系统或将现有的业务系统改造成 Web 形式。虽然这些 Web 业务系统因业务不同有所不同,但从软件开发的角度的看,这些系统的开发过程大致是相同的。开发这些相似的应用系统时,如何减少其中的重复劳动,提高生产率对软件企业和高校来说是至关重要的。MDA 的出现很好的解决了软件开发中遇到的问题和挑战。目前 MDA 的研究方向主要有四个:1) UML PIM 阵营;2) MOF 阵营;3) 可执行 UML 阵营;4) MDD(Model-driven Development)阵营。而在这四种阵营中,MDD 阵营不拘于 MDA 规范,使用 MDA 中现有的技术,结合其他优秀框架来构建平台,更适合现实中的 web 应用。因此,本文中的开发平台采用了 MDD 的思想,最终设计了一个基于模型驱动的管理信息系统开发平台,运用此开发平台能够实现管理信息系统的快速设计与开发。

1 基于模型驱动的管理信息系统化平台研究的意义

1)在研究了模型驱动架构相关理论的基础上,结合领域建模技术,并以 XML 定义元数据,设计了一个业务建模工具。业务建模工具提供集系统设计、代码和数据库自动生成的解决方案的可视化设计工具,主要提供模型定义、转换和模型管理等功能。使用业务建模工具对应用系统建立平台无关模型,再通过模型转换引擎把平台无关模型转换为平台相关模型及代码,从而达到系统设计与具体实现分离的目的,解决管理信息系统开发的重复开发等问题。

2)将 MIS 开发平台应用于实际,可以大大加快 Web 业务系统的开发,特别是对于设计这个阶段尤为明显,提高了软件的生产率。

本系统平台基本上已实现了在进行设计之初所想要达到的要求。通过使用本系统,用户无需编程就可以快速生成满足自己需要的管理系统;可以随意编辑操作页面,达到个性化管理的目的;可以设定不同级别的页面,建立用户管理系统,使系统具有一定的安全性;可以方便的对系统进行修改,以适应需求的不断变化。

2 平台的设计与具体实现

2.1 平台设计思想

随着对 MDA 的深入研究和对管理信息系统开发平台的需求分析,最终设计了一个基于模型驱动的管理信息系统开发平台。该平台基于 MDA 思想,建立一套信息化领域的专业业务模型。该业务模型由业务对象模型,权限模型和应用模块模型组成,以 XML 文件的形式保存,开发平台将解析 XML 文件,获取业务模型的信息,通过模型转换引擎生成 SQL 语句,程序原文件等。

由于 XML 在模型描述上具有互操作性和扩展性等优点,所以平台采用 XML 定义元数据。用户在多个业务系统中身份登陆不一致,造成用户需要使用多个账户和密码登陆各个业务系统,给使用者造成很大的麻烦,为实现用户单点登录,平台采用统一身份认证技术进行用户权限的管理。

2.2 平台体系结构设计

基于模型驱动的管理信息系统开发平台由业务建模工具,服务器端运行时、客户端运行时三部分组成,如图 1。其中业务建模工

收稿日期:2009-09-29

作者简介:付建苏(1983-),男,江苏南京人,在读硕士研究生,交通信息工程及控制。

1)创建远程接口;2)创建本地接口;3)创建 Bean 的实现类;4)创建部署描述符;5)打包;6)部署。

1)创建远程接口

远程接口定义了 EJB 客户调用的方法,该方法在 EJB 类中实现。EJBObject 实现远程接口并且客户端通过它访问 EJB 实例的业务方法。下面是创建数据访问服务的远程接口 DataAccServ 的代码。

```
package DataAccServ;
import java.rmi.*;
import java.util.*;
import javax.ejb.*;
public interface DataAccServ extends EJBObject
{
    Public abstract String execSQLEx(String s)
    throws Exception,RemoteException;
    Public abstract String reviseXmlData(String s)
    throws Exception,RemoteException;
    Public abstract String execSQL(String s)
    throws Exception,RemoteException;
    Public abstract String putXml(String s)
    throws Exception,RemoteException;
    Public abstract String getXml(String s)
    throws Exception,RemoteException;
    Public abstract String getSiteItems(String s)
    throws Exception,RemoteException;
}
```

2)创建本地接口

为了方便 EJB 的创建和发现,每种类型的 EJB 都提供一个 home 接口。Bean 开发人员提供一个 EJB home 作为这个 EJB 的工厂。Home 接口扩展了 javax.ejb.EJBHome 接口,并包括 bean 开发人员识别的一些必要的方法,这些方法允许客户创建、查找和删除。

```
package DataAccServ;
import java.rmi.*;
import javax.ejb.*;
public interface DataAccServHome extends EJBHome
{
    Public abstract DataAccServ create()
    throws RemoteException, CreateException;
}
```

3)创建 Bean 的实现类

定义接口后,有一个实现接口后名的业务逻辑的任务,EJB 的业务逻辑在 bean 类中。这个 bean 包含两个部分:业务逻辑本身,包括远程接口中定义的方法的实现和一套允许容器管理 bean 生命周期的方法。

下面是实现类 DataAccServBean 的部分代码。

```
public class DataAccServBean
implements SessionBean
{
    .....
    Public String getXml(String xmlParam) throws Exception
    {Exception exception;
    String s;
    Try
    {
        If(! AuthHandler.isUserInTrustedUsers(getContextUsers()))
            Throw new Exception("拒绝访问");
        PerfTester.middle("begin getxml");
        GetXmlHandler myHandler = new GetXmlHandler(xmlParam);
        PerfTester.middle("new getXmlHandler");
        Org.w3c.dom.Document retDoc = myHandler.execute();
        PerfTester.middle("execute getXmlHandler");
        String retStr = XMLUtil.dom2String(retDoc);
        PerfTester.middle("node2String(retNode)");
        PerfTester.middle("end getxml");
        S=retStr
    }
    Catch (Exception e)
    .....
}
```

4)创建部署描述符

可以手工创建,也可以使用应用服务器提供的基于 JAVA 的工具来自动生成。

5)打包

