

基于XML自定义的通用表单对象管理平台研究与实现

Research and Realization of a Universal Form Object Management Platform Based on XML

梅军 高岩 赵静 / 中国飞行试验研究院

摘要: 航空企业部门众多,各部门拥有大量表单类管理需求,业务逻辑简单、形式变化快,为每类需求开发一个系统会耗费大量的人力、物力和财力。本文研究利用SSH框架及XML技术,建立一个可扩展的、可自定义的通用表单对象管理平台。

Abstract: For most departments of aviation enterprises, there are kinds of similar form information management needs. These forms have simple business logic and quickly changeable form. If aviation enterprises develop a special information system for each department to satisfy its needs, they will spend a large amount of manpower, material resources and money. Therefore, this paper has used SSH framework and XML technology to build an extensible and customizable platform for universal form object management.

关键词: 对象; SSH框架; XML技术

Keywords: Object; SSH Framework; XML Technology

0 引言

航空企业部门众多,各部门拥有大量表单类管理需求,如试飞资源管理、飞行事故分析、质量管理等。这类需求业务逻辑简单,形式变化快,如果为每类需求开发一个系统将耗费大量的人力、物力、财力,后期维护难度也不可想象。基于此,本文利用Struts、Spring、Hibernate及XML技术,建立了一个IT管理员容易掌握,能够根据XML配置文件快速生成业务系统的应用平台,以满足航空企业大量的表单式管理需求,节约了IT投入的人力财力,降低系统维护难度。

1 平台设计思路

在总体设计思路,本文采用目

前较流行的SSH轻量级J2EE框架。一方面是基于SSH开源框架可减少成本、加快开发进度,另一方面是可以提高系统扩展性。Hibernate^[2]负责数据访问, Spring^[3]负责业务逻辑, Struts^[4]负责展现。数据库方面采用的是DB2数据库。平台总体架构如图1所示。

平台建设其核心是:

1) 将企业业务管理表单对象化,一

个表单对应一个对象。

2) 设计对象导航及权限管理模型。

3) 通XML文件配置对象及其属性信息,读取XML配置文件自动生成对象对应的数据库表和自动生成相应的处理页面。

4) 设计一个强大的数据库引擎,处理对象配置文件,生成及校验数据库表。

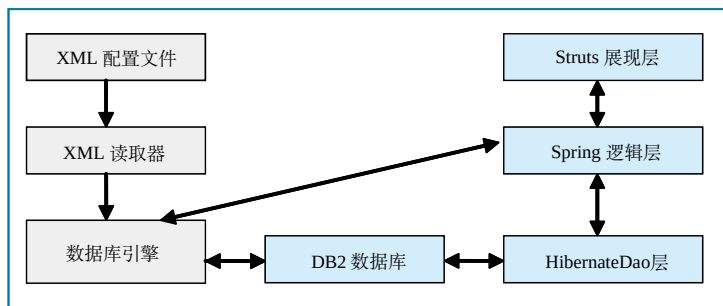


图1 无代码系统开发平台SSH架构图

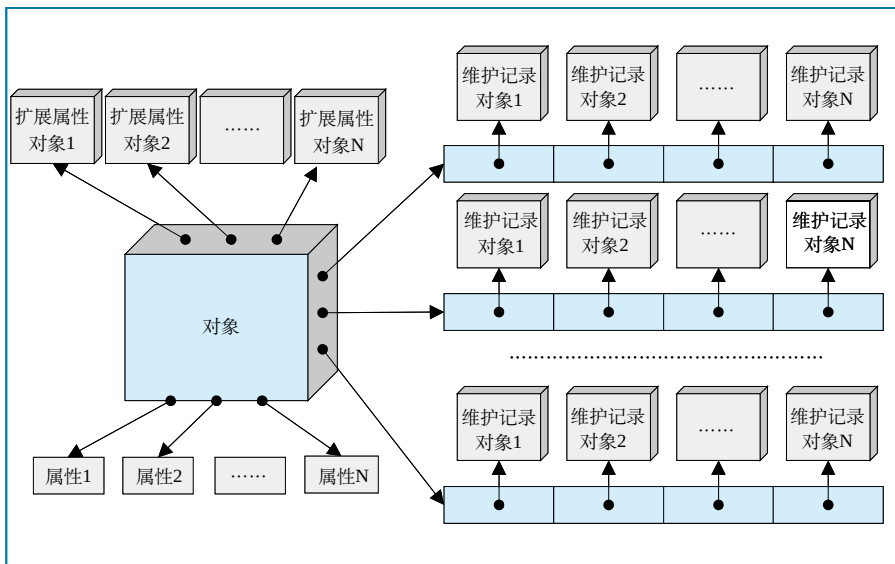


图2 对象数据结构

2 平台数据组织结构

图2描述了整个对象的数据组织结构。一个对象对应一张数据库表,对象属性对应表的列。该对象可能有多个扩展属性关联到其他对象,也就是关联到其他数据表中的某个列值。数据库表中的一行也就是该对象的一条维护记录。

每个对象与其属性是一对多的关系,同时与其扩展属性也是一对多的关系。因此一个对象的数据,通过多个属性或关联的其他对象的某些属性进行组织。

3 平台数据库表设计

平台的数据库表分为两部分,一部

分是基本数据库表,这些是平台用户管理、权限管理、对象管理的基础。另一部分是平台通过配置文件自动生成的数据库表,这些表的表名由配置文件指定的名称进行定义。

平台的基本数据库表如下:

- 1) 用户信息表:定义用户信息。
- 2) 组织机构表:定义部门信息,通过组织机构表可形成组织机构树。
- 3) 角色表:定义了角色信息。
- 4) 角色用户:定义某一角色下的用户。
- 5) 访问权限表:定义某节点下某用户或角色对应的具体权限。

6) 对象记录表:定义根据配置文件生成的对象信息。根据对象ID和分组ID可画出导航树。

7) 对象记录审批表:定义某条对象记录的审批过程。

8) 附件表:定义对象挂接的附件。

4 平台业务逻辑设计

导航树按照树型结构进行组织和管理,这样做的好处是管理清晰,并且可以做到灵活的权限控制。在用户登录后,根据用户ID和根节点ID进行递归查找,当找到下一个节点后,在用户权限表中,查找该用户对该节点的权限,有权限则显示该节点,无权限则递归到下一个节点,直到结束(见图3)。当用户取得一个对象节点的权限后,通过递归查找权限表,获取该用户对该对象的具体权限(见图4)。

为了方便系统管理,

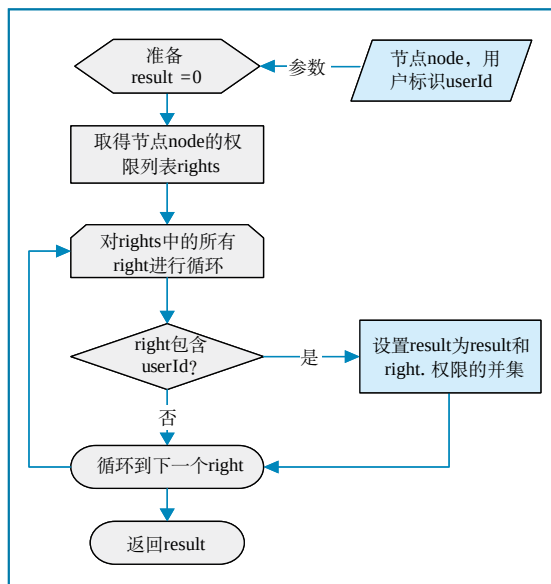
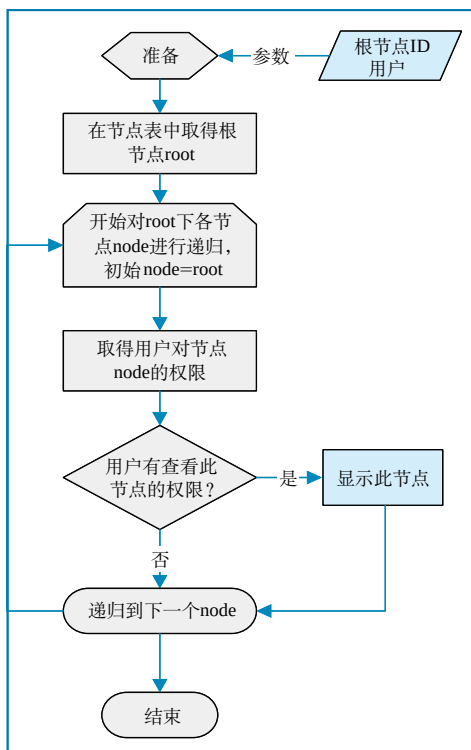


图3(左) 节点树加载逻辑图

图4(右) 用户权限加载逻辑图

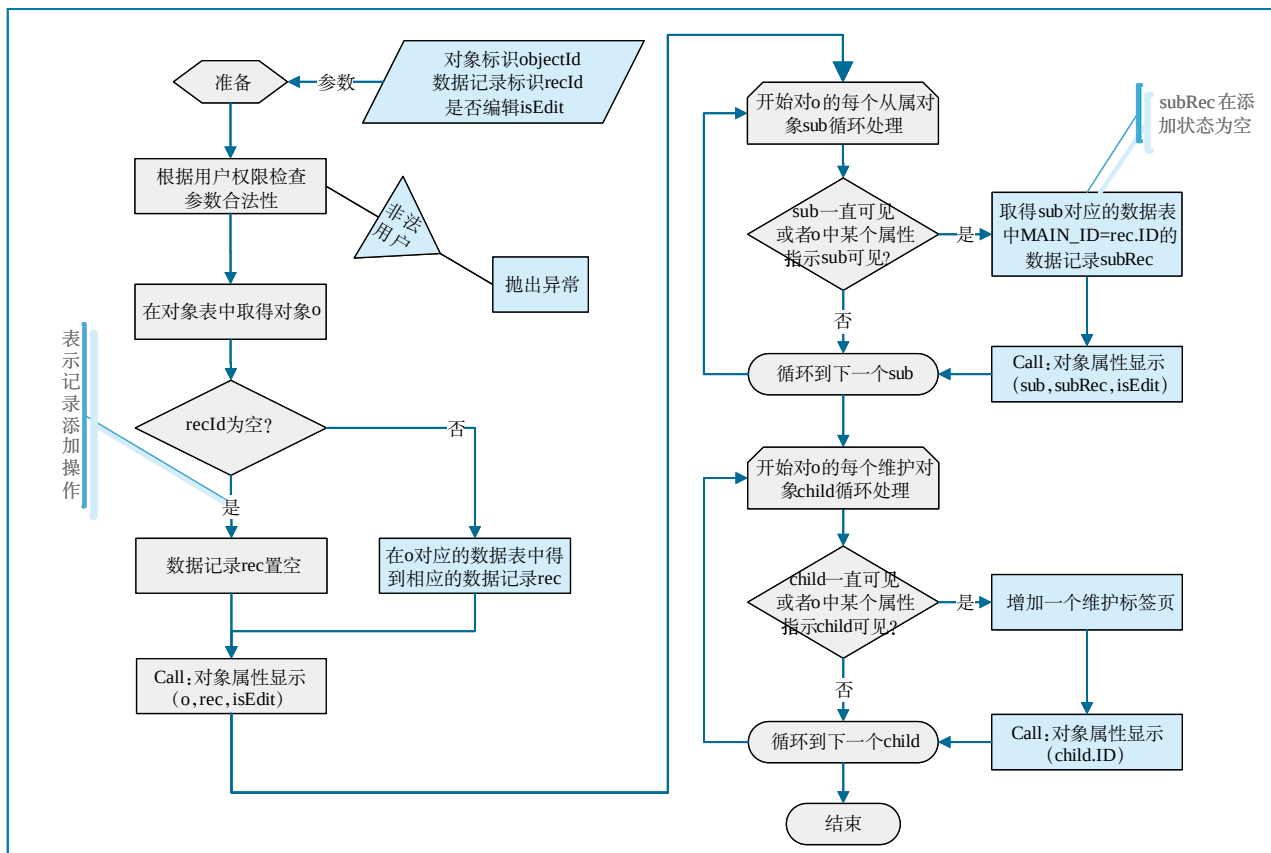


图5 对象内容处理逻辑图

节点的权限可以直接授予,也可以通过父节点继承。对于每个节点,可以直接授权给某个角色组,也可以直接给用户授权。

对象逻辑处理主要包括三个步骤:

1) 对象列表处理:根据对象的ID,在对象记录表中查询该对象,查询该对象的所有列表信息。然后对对象的每个属性进行循环处理,将所有可见属性存储到range。然后通过对象ID及range作为查询条件查询数据库表,得到一个结果集合。将结果集合循环输出。

2) 对象属性处理:当用户具有对象访问权限时,需要对对象每个属性进行处理,根据属性特点显示给用户。如果该属性可见,则判断该属性是否可编辑,当可编辑时显示文本框或复选框等类型显示,当不可编辑时显示为普通文

本。之后判断属性值是否为空,当不为空时将数据作为初始值填充该属性的显示域。

3) 对象内容的处理:当用户通过对象列表链接查看具体某条内容时,根据对象ID、内容ID、是否可编辑参数,首先查找该对象,然后检查该内容ID是否为空,为空说明是添加数据记录,不为空时,找到该条内容,然后调用对象属性显示。之后对对象从属性进行处理,当从属性可见,通过递归查找到从属性所有内容并显示。最后对子对象进行处理,当子对象可见,增加一个子对象维护标签页。图5是对象内容处理逻辑图。

5 配置DTD文件设计

为了通过XML配置文件实现系统自动生成,首先需要对象的各

属性和关联关系进行描述。这就需要自定义DTD文件。文档类型定义(DTD, Document Type Definition)可用于描述XML标记语言,并检验XML文档的有效性^[1]。在实际开发时,笔者定义了2个DTD文件,一个对象配置定义文件object-configuration-1.0.dtd,定义了对象分类,以方便维护管理以及扩展,因为过多的配置文件需要以文件夹方式进行管理。另一个对象类型定义文件object-type-1.0.dtd,定义了对象的基本信息。

对象配置定义文件object-configuration-1.0.dtd定义的元素包括:顶层根、对象实例分类、对象类型、对象列表。代码示意图图6。

对象类型定义文件object-type-1.0.dtd定义的元素包括对象类型、对象

```
object-conf...ion-1.0.dtd 起始页
<?xml version="1.0" encoding="GB2312"?>
<!ELEMENT object-configuration (category*,type-list?)>
<!ELEMENT category (access-rights?, ((url?, category*)|(type, group*)))>
<!--@TLIST category id ID #REQUIRED-->
<!--@TLIST category name CDATA #REQUIRED-->
<!--@TLIST category visible (true|false) #IMPLIED-->
<!ELEMENT url (#PCDATA)>
```

图6 对象配置DTD

```
object-type-1.0.dtd
<?xml version="1.0" encoding="GB2312"?>
<!ELEMENT type (property*, sub-types?, child-types?, approval?)>
  <!ATTLIST type id CDATA #REQUIRED>
  <!ATTLIST type name CDATA #IMPLIED>
  <!ATTLIST type table CDATA #IMPLIED>
  <!ATTLIST type id-field CDATA 'ID'>
  <!ATTLIST type main-id-field CDATA 'MAIN_ID'>
  <!ATTLIST type parent-id-field CDATA 'PARENT_ID'>
  <!ATTLIST type active-str CDATA #IMPLIED>
  <!ATTLIST type flag-bits CDATA #IMPLIED>
  <!ATTLIST type add-alias CDATA #IMPLIED modify-alias CDATA #IMPLIED delete-alias CDATA #IMPLIED query-alias CDATA #IMPLIED report-alias CDATA #IMPLIED approval-alias CDATA #IMPLIED>
  <!ATTLIST type disp-tmp CDATA #IMPLIED>
  <!ATTLIST type report-tmp CDATA #IMPLIED>
  <!ATTLIST type dao-bean CDATA #IMPLIED>
  <!ATTLIST type ui-bean CDATA #IMPLIED>
<!ELEMENT property EMPTY>
  <!ATTLIST property id ID #REQUIRED>
```

图7 对象类型DTD

图8 效果示意图

属性、关联子对象、审批流程等。代码示意见图7。

6 数据库引擎设计

在Spring业务层中,创建了一个ConfigurationAssistantImpl类来处理数据库表的生成和校验逻辑,称之为数据库引擎,主要完成业务逻辑包括:创建对象表、删除对象表、检查哪些对象表需要重建、执行SQL语句、清除所有对象表及表空间、清除数据、检查所有对象类型、重建对象表、判断对象表字段是否为主键、重命名数据表、复制对象表数据到备份表、重建某个类型合并

其所有数据。

7 应用案例

当平台管理员收到一个表单管理需求时,首先根据需求编制XML配置文件,将XML配置文件上传至服务器指定的文件夹,然后进入管理页面,生成对象数据库表,最后进入管理页面,配置用户权限。经验证表明,无代系统开发平台可以通过编辑对象配置文件迅速生成一个系统,熟练的IT管理员三天即可配置生成一个管理系统。下图是某航空企业质量管理体系效果图,如图8所示。

8 结束语

随着因特网的不断发展和普及, Web应用变得越来越广泛,企业对信息化的需求也越来越快速,而每个企业同时渴望信息化的成本越来越低。本文通过面向对象的思想,将航空企业表单式的信息化需求虚拟成对象进行设计,提出了一个基于对象的管理模型,并利用XML配置文件实现对象数据库表的自动生成,其适用范围广泛,一方面可通过快速配置生成可满足常用的表单管理类需求的系统,另一方面,也可以以该平台为基础,开发更为复杂的业务系统,加快项目进度。目前,该平台在为航空企业配置生成了质量安全、事故分析、涉密人员管理等十几个信息系统,节约了大量经费。

参考文献

[1] 慕容雪,蔡月茹. 文档类型定义(DTD)的对象化模型[C].第二十一届中国数据库学术会议论文集.2004.

[2] 陈天河.Hibernate项目开发宝典
[M].北京:电子工业出版社,2006.

[3] 林信良.Spring技术手册[M].第一版.北京:电子工业出版社,2006.

[4] 郭继成.Struts与Hibernate实用教程—构建基于MVC模式的高效Java Web应用[M].北京:电子工业出版社,2006.

作者简介

梅军, 硕士, 工程师, 从事网络应用与开发工作;

高岩，工程师，从事雷达电子试飞工作：

赵静，工程师，从事雷达电子试飞工作。