

MapGIS 数据中心规则设计器设计与实现

单波 胡茂盛 王金玉

摘要 MapGIS 数据中心支持不同种类的数据的可视化功能,并提供扩展机制,支持自定义数据的表达,并提供数据加载工具,支持 ACCESS、SQL、Oracle 数据库并提供数据库表的驱动。遵循 Access 分组驱动的 XML 存储规范,可将 Access 数据库中的一个表或多个表绑定在数据中心的任意一个节点上对应建立一颗子树,实现分组建树,表中的字段对应子树的节点。分组建树的实际意义就是把复杂树抽象成多层次多结构简单的树。数据中心是以 XML 为存储格式的,数据中心的 XML 存储文件主要描述数据中心的属性,数据中心节点样式类型,数据中心节点属性三大部分。为了建立相应的 XML 规则,需要一个规则设计器。

关键词 规则 设计器 XML

中图分类号 P208 文献标识码 A

一、引言

MapGIS 数据中心(以下简称数据中心)是用于组织和操作各种空间数据和非空间数据的资源管理器,集各种数据管理和操作为一体的面向应用的集成框架。数据中心以巧组合,零编程和易搭建为出发点,真正实现不懂软件开发也可以搭建 GIS 系统的梦想。

数据中心的配置文件都是通过 XML 进行存储的,数据中心驱动是数据中心为扩展插件资源而进入的一种机制,是数据中心节点加载子节点系统而提供服务的一种动态链接库。数据中心驱动包括预定义驱动和用户自定义驱动。数据中心预定义驱动包括文件驱动、地理数据库驱动和地理数据库域名解析服务驱动。用户自定义驱动是用户根据自己具体应用的需要编写的驱动。通过此驱动可以设置适合各种业务的规则,在此基础上新建的数据都必须符合此规则。驱动符合规则的 XML 文件可以生成一个目录树,目录树是数据中心组织、管理和操作数据的场所和工具。Access 分组驱动的 URL 的一般格式为 Access 分组驱动为内置驱动,无需用<driver>属性进行声明,使用时为 TmpDrv 协议,用于访问本机中的 Access 数据库。

二、规则设计器

(一)规则。

通过数据中心规则结点驱动,可以将复杂的规则拆分成多层次多结构简单的规则。复杂的规则可以拆分成六种层,根层、联合分组层、定形分组层、单元分组层、纵向分组层和逐记录分组层。TmpDrv:///Access 数据库路径,例如 TmpDrv:///F:\Ar-

cSysDB .mdb,就是指驱动 F 盘下的 ArcSysDB .mdb 数据库。通过对数据库中数据的组合和属性设置,数据中心利用 Access 分组驱动负责将此 Access 表格生成数据中心目录树可以在界面上直接表达,进而实现数据的管理与操作。

目录树是通过目录配置实现的,目录树的数据遵循 URL 格式规范,目录树上的每个节点可以配置一个或多个功能,目录树的子节点可以通过驱动来实现管理。

统一的规则,可以实现数据的共享与集成,

数据中心的属性描述了数据中心的相关信息,例如所属部门、创建日期等。数据中心节点样式类型描述数据中心中一些节点的公有属性,作为样式类型属性的形式提取出来,避免重复设置相同的属性值,这部分可以没有,也可以作为一个独立的文件引用。数据中心节点属性描述了数据中心面向应用的数据属性,包括根节点和其各级子节点。根节点除了没有 name 属性外,其它属性和子节点属性大致相同,它们作为一个部分介绍。

数据中心属性主要描述了数据中心整体的相关信息,其主要属性如表 1 所示:

表 1 数据中心属性

属性名	意义	属性值类型
version	版本号	字符串
name	名称	字符串
department	所属机构	字符串
createtime	创建日期	日期型
extend	扩展属性	一个或多个键值对,都为字符串型
define	预定义属性	取决于预定义的属性
driver	数据中心驱动	属性结构

(二)XML。

XML 是 Extensible Markup Language 的缩写,XML 是一种类似于 HTML 的标记语言,XML 是用来描述数据的,XML 的标记不是在 XML 中预定义的,你必须定义自己的标记,XML 允许你定义自己的标记以及文档结构。XML 使用文档类型定义(DTD)或者模式(Schema)来描述数据,XML 使用 DTD 或者 Schema 后就是自描述的语言。XML 是被设计用来描述数据的,重点是:什么是数据,如何存放数据。XML 数据是以纯文本格式存储的,XML 将被用来存储数据、携带数据和交换数据,而 HTML 则是用来格式化和显示数据的。对于 XML 最好的形容可能是:XML 是一种跨平台的,与软、硬件无关的,处理信息的工具。XML 将成为最普遍的数据操纵和数据传输的工具。基于 XML 的特性,数据中心在数据存储选择了 XML 格式,相应的设计器也采用了此格式。

(三)规则设计器。

目录树是树形展示,再基于数据的冗余性、复杂性和多变

性,为了使用户方便的设置需要的数据,采用了C#的PropertyGrid控件,此控件可以方便的修改相应的数据,可以描述树结点的信息。使用C#提供的数据库操纵工具和XML工具,可以生成和读取规则。基于以上考虑,采用了此种设计。

为了实现设计器,设计了两个模块的功能:树生成主模块和向导模块。由于XML是以文本格式存储的,所以在数据处理过程中,采用了文本化处理。读取数据库和注册表信息,数据库的信息用于结点的分类和信息采集;注册表的信息可以以COM组件的形式添加到 workflow 当中,数据中心设计器可以直接使用,COM组件的功能可以直接实现。

设置每层的信息,数据和分类信息后,可以保存规则,再调用相应的规则驱动,就可以实现个性化的GIS软件了。

三、解决方案

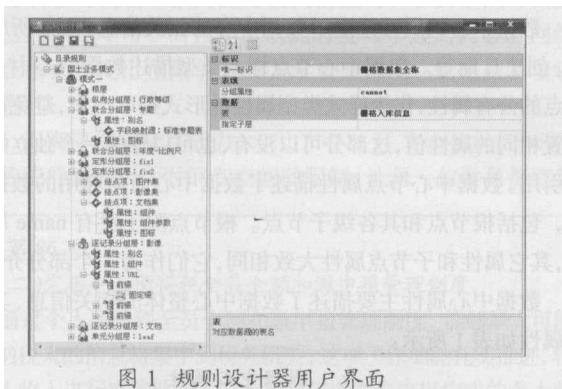


图1 规则设计器用户界面

通过规则设计器设计后,可以得到遵循Access分组驱动的XML存储规范和响应的一个驱动XML,如下:

(一)驱动XML:

```
<? xml version="1.0" encoding="gb2312"?>
<datacenter name="数据中心" version="1.0" department="
zondycyber" createtime="2008-10-28">
<extend key="ConfigPath" value="{ConfigRoot}" />
<extend key="ResPath" value="{ConfigRoot}res" />
<driver type="subnodesystem" name="TplDrv" src="
DcDrv_template" />
```

```
<root>
<URL>TplDrv://</URL>
<template>file:///国土业务模式/模式一</template>
</root>
</datacenter>
```

(二)遵循Access分组驱动的XML

```
<? xml version="1.0" encoding="utf-8"?>
<dcml>
<business name="国土业务模式">
<template name="模式一">
<metaDB>file://{$MainRoot}DataCenter{database}\驱动建树
```

```
数据库\ArcSysDB.mdb</metaDB>
<Group>行政等级;专题;年度</Group>
<Layer name="root" regroup="root" table="专题入库信息">
<Type extended="true"/>
<attribute name="AliasName">【数据库】</attribute>
<attribute name="Icon">数据库 ico</attribute>
</Layer>
<Layer name="行政辖区" union="行政代码" subLayer="fix">
<filter table="文档入库信息">
<fieldMap src="行政代码" dst="行政代码">
</fieldMap>
</filter>
<attribute name="AliasName" map="行政名称"/>
<attribute name="Icon">目录行政区划 ico</attribute>
</Layer>
</template>
</business>
</dcml>
```

四、实例

使用此规则设计器设计的江西国土业务规则,通过驱动,可以实现如下效果。

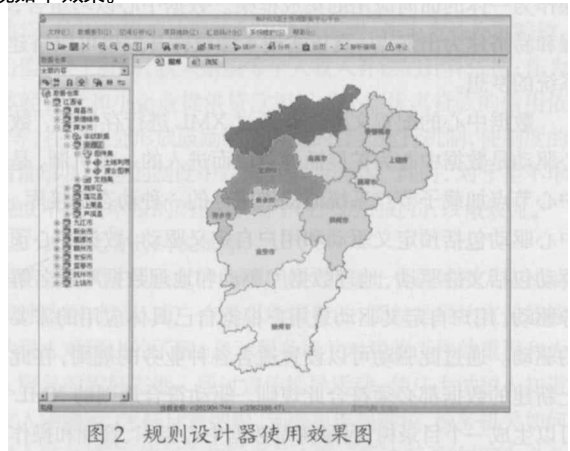


图2 规则设计器使用效果图

(作者:单波:中国地质大学(武汉)在读研究生,研究方向:GIS理论与软件开发;胡茂盛:中国地质大学(武汉)在读研究生,研究方向:空间数据库;王金玉:中国地质大学(武汉)在读研究生,研究方向:WebGIS理论与软件开发)

参考文献:

[1] Abraham Silberschatz, Henry E. Korth & Sudarshan 著,杨冬青,唐世渭等译.数据库系统概念.机械工业出版社,2008
 [2] 吴信才.地理信息系统原理与方法.电子工业出版社
 [3] Sams Teach Yourself XML in 21 Days, April 1999
 [4] Microsoft Corporation, "C# Language Specification" White paper, 2001: <http://www.microsoft.com/msdn>