

元数据驱动的土地信息系统可复用构件研究

岳建伟^{1,2}, 钟耳顺¹, 张秋义³, 姚敏^{1,2}, 方利^{1,2}

(1 中国科学院地理科学与资源研究所, 北京 100101;

2 中国科学院研究生院, 北京 100039; 3 北京超图地理信息技术有限公司, 北京 100085)

摘要: 目前对土地信息系统的研究主要集中于特定项目, 因此, 在土地管理领域中可复用的构件非常匮乏, 这对于土地信息系统的发展十分不利。此外, 对元数据的研究, 主要集中于元数据的标准、元数据的表达形式、元数据的管理和元数据的内容上, 对元数据应用于土地信息系统的开发研究则较少。本文将元数据技术引入构件的开发中, 采用元数据定义业务规则, 通过元数据驱动构件的运行。在业务发生变化时, 只需要改变业务规则在元数据库中的定义就可以使构件满足业务变化的需求, 提高业务构件的可复用性, 扩展了元数据的应用范围, 为土地信息系统的快速开发奠定了基础。

关键词: 元数据; 软件复用; 土地信息系统; SuperMap

1 引言

土地管理的业务是随行业各种因素的变化而不断变化的, 因此, 土地管理信息系统必须能够适应这些变化。为了快速开发出适应业务需求的土地信息系统, 软件复用技术被采用。软件复用发生在构造新的软件系统过程中, 是在新的系统开发中充分地利用已有的开发成果, 消除包括分析、设计、编码、测试等在内的许多重复劳动, 从而提高软件开发的效率; 同时, 通过复用高质量的已有开发成果, 避免重新开发可能引入的错误, 从而提高软件的质量^[1]。

目前对软件复用的研究主要集中于复用已有的软件构件, 通过构件集成(组装)而快速开发出新的系统^[2]。这些研究的确极大地促进了软件系统的开发, 但是当业务需要发生改变时, 利用这些可复用构件迅速构造的系统由于业务规则固化在其中, 很难适应需求的变化。而系统用户所关心的正是这一点: 如何在业务需求发生变化时可以使系统快速反应, 以适应业务的需求。有些研究称其为构件的灵活性, 本文认为这其实也是一种软件复用。与系统开发阶段的软件复用不同, 它是在系统运行中当

业务规则发生变化后, 要求系统能够适应这些变化并继续运转。土地管理的业务是不断变化的, 因此研究系统的灵活性就显得极其重要。本文将元数据技术引入构件的设计开发中, 利用元数据驱动构件的运行, 对提高系统的灵活性进行了有益的探索。

2 系统图形构件设计

土地信息系统通常由三类构件组成: ①通用基本构件, 是特定于计算机系统的构成成分, 如基本的数据结构、数据访问构件等, 它们可以存在于各种应用系统中; ②领域共性构件, 是应用系统所属领域的共性构成成分, 它们存在于该领域的各个应用系统中; ③应用专用构件, 是每个应用系统的特有构成成分^[3]。目前的软件复用研究主要集中在前两类。但是通过对现有系统分析发现, 由于土地管理领域的业务规则包含在后两类中, 因此, 土地管理领域共性构件和应用专用构件决定了土地信息系统的行为。所以, 提高系统灵活性的关键就在于如何使这两类构件适应业务需求的变化。

对现有运行中的土地信息系统分析发现, 在运

收稿日期: 2005-02-23; 修回日期: 2005-10-14.

作者简介: 岳建伟(1975-), 男, 中国科学院地理科学与资源研究所博士研究生, 主要从事国土资源信息系统研究。

E-mail: yuejw@supermap.com

行中变化的主要是业务规则,如统计报表的输出内容、活动拥有的权限、查阅的图形内容等。在现有的土地信息系统图形构件(或称为传统的图形构件)中这些内容大多是在构件中预先定义好的,所以在系统运行过程中当需要改变内容时必须重新修改并编译构件,因此,后期运行中系统的更新几乎是不可能的。解决上述问题的关键在于如何将构件包含的业务规则从构件中分离出来,如何控制构件根据一定的信息找到要处理的数据内容和数据处理的规则,从而能够使之适应不断变化的业务需求。元数据提供了满足这些需求的答案。

元数据帮助人们定位和理解数据^[9]。根据元数据的这一特性,元数据驱动的土地信息系统可复用

构件的设计思想就是将业务办理中的业务规则从构件中剥离出来,利用元数据进行描述并在元数据库中定义;构件运行时通过读取元数据库中定义的相应内容来确定业务规则,根据这些信息对数据进行相应的处理。由于构件的业务规则是在元数据库中定义,而不是在编写构件时就固定,所以在后期系统运行中,在业务发生变化时,只需要修改元数据库中相应的元数据描述信息就可以很容易改变业务规则,从而提高构件的灵活性和可复用性。元数据驱动的构件工作模型如图1中b所示。

(1)参数设计:构件是嵌入在系统中的,必须拥有与外部进行交流的通道,因此我们需要清楚构件必须从系统获得什么信息才能够正确运行。构件运

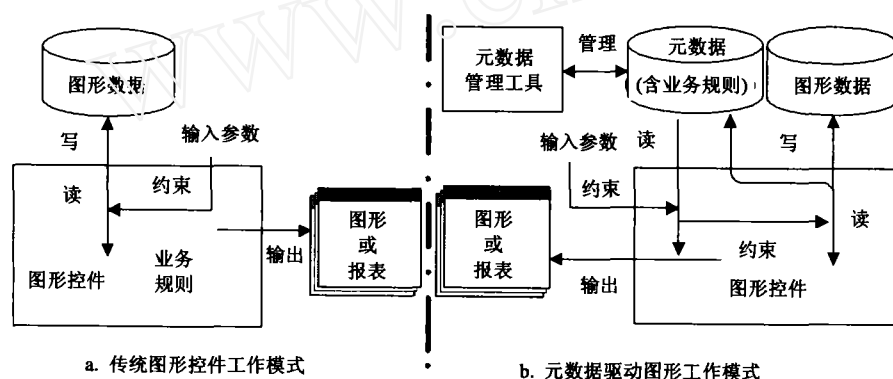


图1 传统构件与元数据驱动构件工作模型对比

Fig.1 The comparison of working model between traditional component and metadata driven component

行中需要知道要处理的图形数据的位置、内容以及如何处理,这些内容定义在元数据库中。在构件运行时,首先应该将有关元数据库的信息传给构件;其次应该将要操作的数据内容在元数据中的定义信息传给构件;再次,应该将对操作数据的操作权限在元数据库中的定义信息传给构件;此外,在业务流程办理时,还应该将项目信息传递给构件。有了这四种基本的信息,构件就具备了正常运行的条件。参数见表1。

(2)工作过程:元数据驱动的构件处理数据的工作过程如图1所示,首先根据系统传递参数来读取元数据库中定义的构件需要处理的数据,然后从业务数据库中读取相应的数据(图形数据或业务办理信息),并加载相应的图形功能模块,完成上述活动后等待进一步的指令;在获得指令后执行指令(如图形平移等不涉及改变输出、输入内容)或者先读取元数据库中执行该指令相关的内容后再根据

元数据内容执行指令,最后将处理结果输出。

表1 图形构件主要参数功能

Tab.1 The primary parameter of graphics component

参数说明	功能
元数据数据源(数据库连接字符串)	指定元数据数据库所在的位置、数据库名称、用户名、密码等
操作权限	根据操作权限显示不同的工具栏组合和图形数据
项目编号(流程性业务中用到)	指定当前办理的项目编号,非业务流程时可以为空
项目名称(流程性业务中用到)	指定当前办理的项目名称,非业务流程时可以为空
项目类型(流程性业务中用到)	指定项目类型,以利于后期查询统计,非业务流程时可以为空

下面以土地管理信息系统图形控件中建设用地审批管理查询申请用地占用土地利用现状的情况为例,进一步说明元数据驱动构件的工作过程。

(1)图形构件从它所在的土地管理系统中获得处理数据的参数,确定元数据的位置,从元数据库中获待处理数据的信息。假设在网页中已经放置名为 prjJSYDControl 的图形控件,prjJSYDControl 获得元数据库信息的代码如下:

```
<script language="vbscript">
.....
window.Form1.prjJSYDControl.strMetaDs="<% =
strMetaDs%>" //元数据库连接字符串
window.Form1.prjJSYDControl.intRightType =<%
=intRightType%> //操作权限
window.Form1.prjJSYDControl.intPrjType =<% =
intPrjType%> //项目类型
window.Form1.prjJSYDControl.strPrjNmb="<% =
strPrjNmb%>" //项目编号
window.Form1.prjJSYDControl.strPrjName="<% =
strPrjName%>" //项目名称
window.Form1.prjJSYDControl.Init
//控件初始化
</script>
```

(2)图形构件根据元数据库中对权限的定义加载图形控件的菜单栏和图形内容,最后根据参数中项目的信息突出显示项目信息。这是一个复杂的过程,不同的 GIS 软件都有各自不同的处理方式。

(3)查询申请用地占用土地利用的现状,并将申请用地与土地利用现状地类图斑进行叠加分析,然后将其结果以报表形式显示。由于分析报表内容随业务的发展会发生变化,因此在这里我们用元数据来控制报表的内容,而不是在编写控件时将之固定。

假定地类图斑表中有地类编号、地类名称、图斑编号、平差面积、权属单位、所在图幅等 6 个字段,而在分析报表中只需要显示地类名称、图斑编号、平差面积、权属单位等 4 个字段的信息。元数据库中数据结构及数据定义如图 2。

在图形控件输出分析结果时,从表 META_DT_FLD 中获得要输出的内容限制(业务规则),然后再根据分析情况从土地利用数据库的地类图斑表中读取信息输出给用户。

Public Sub ShowAnalyseResult (objRegion As so Geo

METZ_DT (表格定义)		META_DT_FLD(表格字段对照)			META_FLD(字段定义)		
MBBSM	DTMC	DTZWMC	MBBSM	DTID	FLDID	MBBSM	ZDMC
							
			190	161	205	205	DLMC
			191	161	206	206	TBBH
161ZYXZCX	占用现状					207	PCMJ
	查询						
			192	161	207	208	QSDW
			193	161	208	209	SZTF
						210	DLBH
							

图 2 数据定义示意图

Fig.2 Sketch map of defining data

```
Region, objWks As SuperWorkspace, _
    lvwRes As ListView)
'作用显示分析结果
'g_strLuDsa,全局变量,为元数据库中定义的土地
利用现状数据库连接字符串;
'DLTB 为地类图斑的表名;META_FLD 为字段解释
表;META_DT 为数据集表;
'META_DT_FLD 为数据集和字段解释对照表
lvwRes 为显示报表的控件
lvwRes.ListItems.Clear '先将以前结果清空
lvwRes.ColumnHeaders.Clear
```

```
Dim tmpCol As Collection
Dim i As Integer
Dim objDs As soDataSource
Dim objDt As soDatasetVector
'根据元数据字段解释表的内容,动态添加
ColumnHeaders
strSql = "select * from META_FLD where MBBSM in
(select FLDID from META_DT_FLD " & _
"where DTID in (select MBBSM from META_DT
where" & _
"rtrim (ltrim (DTMC))= 'ZYXZCX') order by ZDXH"
```

```

adoRs.Open strSql, g_strMetaDs
adoRs.MoveFirst
Do While Not adoRs.EOF
lvwRes.ColumnHeaders.Add,Trim (adoRs ("ZDMC")),
Trim(adoRs("ZDZWMC"))
adoRs.MoveNext
Loop
'获取和给定多边形相交的地类图斑
Set tmpCol=GetIntersectTDLYXZ(objRegion, objWks)
If tmpCol Is Nothing Then Exit Sub
'添加占地信息的每一条记录
Set objDt=objWks.Datasources.Item(g_strLuDsa).
Datasets ("DLTB")
If objDt Is Nothing Then Exit Sub
For i = 1 To tmpCol.Count
Dim nSmID As String
nSmID = tmpCol(i)
Dim objRs As soRecordset
Set objRs = objDt.Query("SmID=" & nSmID, True)
If Not objRs Is Nothing Then
objRs.MoveFirst
If Not objRs.IsEOF Then
Dim tmpRegion As soGeoRegion
Dim itm As ListItem
Set itm = lvwRes.ListItems.Add(, , i)
For j=1 to adoRs.Fields.Count
If Not IsNull(objRs.GetFieldValue(adoRs(j).
Name)) Then
itm.SubItems(j) = objRs.GetFieldValue(adoRs(j).
Name)
End If
Next j
End If
End If
Next i
End Sub

```

(4)适应变化。通过上面程序可以看到,在程序中显示的数据内容完全由元数据来定义和控制的,如果后来要求在输出的内容,将“所在图幅”信息加入表中,此时只需要在元数据表 META_DT_FLD 中将其定义进去即可(如图 3 所示),而无须改变程序代码。

META_DT_FLD (表格字段对照)		
MBBSM	DTID	FLDID
.....		
190	161	205
191	161	206
192	161	207
193	161	208
194	161	209
.....		

图 3 改变字段定义示意图

Fig. 3 Sketch map of changing fields

3 应用实例

柳州市土地管理信息系统以地籍管理、建设用地管理和土地监察业务为主体。该系统要求分层叠加显示规划图形数据(如城市总体规划、分区规划、详细规划、专项规划、土地利用总体规划图)、地价分等定级图、现状图形数据(地形图、土地利用现状、现状管网)、规划审批与房地产登记及测量控制点网等各种图形数据,并可以方便地控制图层是否显示。该系统容纳了多种图形,承担了很多的业务,因此在系统运行中在需求发生变化时,该图形控件都有需改变的可能。如果在控件开发中将业务规则固化其中,那么它将无法适应新的需求。为了解决此问题,该系统的图形构件采用了元数据驱动原理设计,在构件式 GIS 软件 SuperMap Objects 2003 基础上进行二次开发。

柳州市土地管理信息系统图形构件需要完成建设用地管理、地籍管理、土地现状和规划利用,其功能强大,为了更好对其进行控制,设计时将前文构件参数的操作权限扩展为编辑权限(如打印、编辑、查询等)和编辑类型(对哪些图层操作)。在元数据库中,将图形控件各功能模块操作的内容进行定义,在图形控件运行时动态从元数据库中读取操作内容。从而达到在业务需求改变时,改变元数据库相应操作的定义就能方便迅速地改变图形构件的功能,以适应业务需求。

为方便元数据的采集和管理,在柳州市土地管理信息系统的开发中编制了元数据管理工具。利用该工具,用户可以非常容易管理元数据的内容,并且在后期业务变化时,用户使用该工具能方便地对

元数据进行修改。

4 结语

本文通过分析土地信息系统中业务构件的工作模式,将业务实现的规则从构件中提取出来,利用元数据进行描述并存储在元数据库中,在构件运行中动态地读取元数据来获得业务的规则,从而使构件可伴随业务的变化而变化,提高了业务构件的灵活性,改善了业务构件的可复用性,为开发可复用的土地信息系统业务构件提供了一个新的途径,为建设基于构件的土地信息系统和元数据驱动的

土地信息系统进行了有益的探索。随着元数据技术的发展和元数据在土地信息系统开发中应用的深入,元数据驱动的土地信息系统开发是一重要的技术方法。

参考文献

- [1] 陆耀宾. 构件重用技术在雷达软件中的应用. 现代雷达, 2004(4):51~53.
- [2] 杨芙清, 梅 宏, 李克勤. 软件复用与软件构件技术. 电子学报, 1999, (2):68~75.
- [3] David Marco(美)著, 张 铭, 李 钦等译. 元数据仓储的构建与管理. 北京:机械工业出版社, 2004:2~3.

Study on Metadata-driven Reusable Components of Land Information System

YUE Jianwei^{1,2}, ZHONG Ershun¹, ZHANG Qiuyi³, YAO Min^{1,2}, FANG Li^{1,2}

(1 Institute of Geographic Sciences and Natural Resources Research, Beijing 100101, China;

2 Graduate School of the Chinese Academy of Sciences, Beijing 100039, China;

3 SuperMap GIS Technologies, Inc, Beijing 100101, China)

Abstract: Current research on land information systems mainly focuses on particular projects, thus causing a great lack of reusable components in the land management field. This is unfavorable to the development of land information systems. Besides, research on metadata is mainly concentrated on metadata standards, ways of metadata expression, metadata management and metadata contents. Research on and development of metadata applications in land information systems are quite inadequate. This paper introduces metadata technology into the development of components. It adopts metadata to define business rules and drives the running of components through metadata. When business changes, it only needs to change the definition of business rules in the metadata base and use components to meet the needs of business changes. This will improve the reusability of business components, expand the scope of metadata applications and lay a foundation for the rapid development of land information systems.

Key words: metadata; software reuse; land information system; SuperMap