

doi:10.3799/dqkx.2010.039

数据中心:GIS 功能仓库的关键技术

万波^{1,2}, 杨林^{1,2}

1. 中国地质大学信息工程学院, 湖北武汉 430074

2. 地理信息系统软件及其应用教育部工程中心, 湖北武汉 430074

摘要: 许多组织和部门需要地理信息功能的协同与合作来完成一项工作, 其关键在于解决地理信息系统(geographic information system, GIS)领域的构件复用和协同。采用支持黑盒复用的基于构件软件框架, 探讨了支持多粒度、异构 GIS 构件的功能仓库的关键技术, 提出原子粒度、功能组件、功能插件、功能流程等多粒度构件组织; 定义了统一的功能协议对异构功能资源统一管理 and 发布, 以统一的方式进行配置和系统搭建, 使运行时服务能够通过功能协议动态的发现和调用功能资源, 并实现功能资源的定位与参数传递。功能仓库能够集成各种类型异构、来源异构的功能资源, 实现这些功能资源在数据中心的统一管理和调用, 最终基于工作流引擎与可视化建模工具实现地理信息功能的协同工作。

关键词: 数据中心; 功能仓库; 构件复用; 地理信息系统。

中图分类号: TP311.13; TP311.5

文章编号: 1000-2383(2010)03-0357-05

收稿日期: 2010-01-15

Data Center: GIS Function Warehouse

WAN Bo^{1,2}, YANG Lin^{1,2}

1. Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China

2. Engineering Research Center for GIS Software and Applications, Ministry of Education, Wuhan 430074, China

Abstract: More and more organizations and departments need collaboration of geographic information function to complete work. The key is component reuse and collaboration in geographic information system (GIS) area. This paper adopts the component framework based on black box reuse, discusses the key technologies of function warehouse. Firstly, multi-granularity component structure is presented, including atom granularity, function component, function plug-in and function flow; Secondly, uniform function protocol is defined for unified management and release, unified configuration and system building on heterogeneous function resources. Through unified protocol, running service can discover and call function resource dynamically and achieve the function resource position and parameter transfer. Function warehouse can integrate various types of function resources, implement the uniform management and call in data center, and finally achieve collaboration of geographic information function with workflow engine and visual model building tools.

Key words: data center; function warehouse; component reuse; geographic information system (GIS).

0 引言

软件构件概念自 1976 年在 NATO 软件工程会议上首次提出 (McIlroy, 1976), 至今构件实现的技术已比较成熟, 业内通用的构件模型主要有 COM/DCOM 及 CORBA 技术 (艾萍, 2005)。随着软件构件技术逐渐深入人心, 各种异构的软件构件的数量愈来愈大, 构件开发者关注的重点已经从最初的强

调构件的复用性, 转变到支撑软件构件整个生命周期管理的构件库的建设。构件库为不同领域的构件提供一个开放的注册、管理、检索和交易的场所, 这是当前构件库的主要形式。国外比较有代表性的构件库有欧盟信息技术计划 ESPRIT 中开发的 Reboot 系统、CMU SEI 开发的 Agora 构件搜索引擎和美国 Colorado 大学开发的 CodeBroker 构件库 (Ye, 2001; 潘颖等, 2003)。国内有北京大学青岛

构件库管理系统(JBCLMS)和上海构件库。

借鉴构件库的思想,吴信才(2009)提出地理信息系统(geographic information system, GIS)功能仓库的概念,认为 GIS 功能仓库是内置了 GIS 功能的功能仓库,是对 GIS 构件进行管理并对外提供构件服务的管理系统,它提供构件的分类、存储和查询,负责构件调度。GIS 功能仓库是构件技术在 GIS 领域中的应用, GIS 构件仓库首先是一个存放构件的场所,也是支撑 GIS 应用系统运行的组成部分,向外界的调度提供统一的接口,为用户提供功能服务。

“数据中心”是基于新一代 GIS 架构技术及新一代开发模式的集成开发平台,该“系统开发者”既管理“数据资源”又管理“功能资源”(叶亚琴等, 2006;周顺平等, 2006;吴信才, 2009)。本文的 GIS 功能仓库即是“数据中心”的功能资源管理中心。“数据中心”的功能仓库技术集成了大量类型异构、来源异构的功能资源,并实现了这些功能资源在“数据中心”的统一管理和调用。这些功能资源包括 MapGIS 功能、ArcGIS 功能、用户开发的系统功能、Windows 操作系统中遵循 COM 规范的组件、网络上发布的 Web 服务功能等。功能仓库将这些资源进行多粒度功能划分,通过功能仓库接口规范进行封装、反射或者通过可视化功能建模工具进行功能搭建,生成符合功能仓库规范的功能协议,再通过功能仓库元数据管理对功能协议进行统一的管理和发布。最后,通过工作流引擎和功能仓库执行体进行统一的调用。

功能仓库是一个高层的统一框架,为维持良好的适应能力和可扩展能力,它提供了一系列的机制和协议,包括为了支持不同层次和不同大小的功能而提供的多粒度功能组织机制;为了支持 GIS 软件功能、用户开发的系统功能、遵循 COM 规范的组件、Web 服务功能等不同种类的功能资源而定义的功能协议;为支持各种功能资源发挥效力的功能调用机制。本文将对这些功能仓库的关键技术进行深入分析。

1 多粒度功能组织

GIS 功能仓库支持对数据仓库提供的数据进行查询检索、空间分析、输出等功能及功能的组合。考虑到应用系统对功能资源使用的粒度依实际需求有所不同,功能仓库按照多种粒度对功能资源进行组

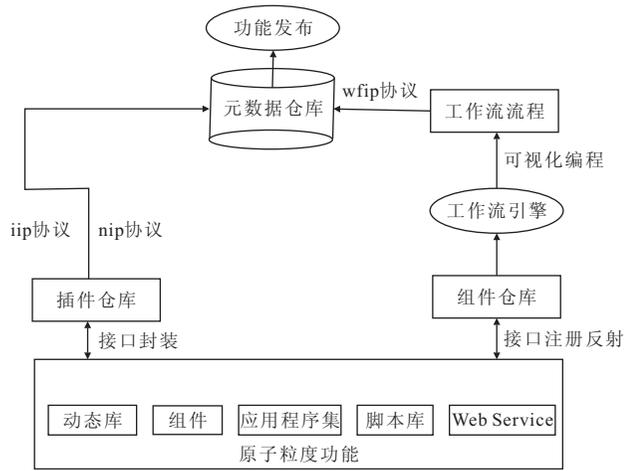


图 1 多粒度功能组织

Fig. 1 Organization of multi-granularity

织,包括原子粒度功能、功能组件仓库、功能插件仓库和功能流程仓库 4 个层次,为应用系统提供不同粒度的功能支持(图 1)。

1.1 原子粒度功能

原子粒度功能包括动态库、组件、应用程序集、脚本库、Web Service 等。原子粒度功能分通用功能和 GIS 功能两种,通用功能为 Window 提供的基础功能;GIS 功能则是 GIS 平台软件提供的原子粒度功能,可能是构建在通用功能之上的,也可能是独立于通用功能的。

1.2 功能组件仓库

功能组件仓库对原子粒度功能里面的组件、应用程序集、Web Service 进行接口反射,将这些功能资源里面的类和方法入库,形成功能仓库最小粒度的功能资源库。例如地理数据库对象的获取、要素类的叠加分析都是功能组件仓库中的功能资源。

1.3 功能插件仓库

功能插件仓库管理更大一级粒度的功能,可能包含一个或多个功能组件。根据功能仓库接口规范对所有的原子粒度功能进行封装生成标准插件,通过系统注册录入功能插件仓库,并发布功能协议。根据功能的需要,功能插件库分为无界面插件和有界面插件。例如要素类的预览、图层的图元输入均为无界面插件;要素类的属性结构设置、要素类的属性统计则为有界面插件。

1.4 功能流程仓库

功能流程仓库管理各种功能流程。功能流程是最大粒度的功能资源,它通过工作流可视化建模工具对小粒度的功能资源库和大粒度的功能插件库进

行组合搭建而成. 功能流程自身又可以嵌套其他功能流程, 形成多级粒度功能组合. 例如针对组件进行搭建, 从统一资源定位符(uniform resource locator, URL)中获取要素类对象、从 URL 中获取地理数据库(geodatabase, GDB)对象、要素类的叠加分析, 这几项功能组件聚合形成一个实现地理数据叠加分析的流程; 针对插件进行搭建要素类预览、要素类属性结构设置和要素类的属性统计, 这几项插件聚合形成针对地理数据的系列操作, 完成针对用户具体需求的一个自定义地理数据处理功能.

2 功能协议

针对各种异类异构功能资源, 功能协议用于统一功能资源定位, 该协议使用类似 URL 协议的方式定义, 它由一系列的段组成, 前一段说明了获得下一段需要采用的方法和参数. 每一段代表不同的功能资源级别, 功能协议最终定位到具体功能, 再由有类似 URL 协议参数的方式传递统一的功能参数.

定义统一的功能协议就能对功能资源进行统一的管理和发布, 以统一的方式对功能资源进行配置和系统搭建, 运行时服务就可以通过功能协议动态地发现和调用功能资源.

2.1 iip 协议

iip 协议用于在应用系统中实现功能插件库里的某项功能, 可以配置到系统菜单、工具条和右键菜单上. 它的结构如下:

```
<iip://PlugIn Name/Main Scene>
```

其中, iip 代表协议名称, PlugIn Name 指示指定插件, Main Scene 表示主场景, 指示响应事件.

主场景下包含多种事件集合, 例如常用的“MainActive”, “SubActive”和“CancelActive”. 在数据中心中执行此功能时, 鼠标左键默认执行“MainActive”事件, 鼠标右键对应“SubActive”事件, 退出该功能对应“CancelActive”事件.

在实行功能配置时, 主要配置在目录树节点上的“关联场景”<com>属性. 另外, 针对某些需要参数的插件, 需要在目录树节点上的“场景参数”<comargs>属性上, 参数的类型由调用的插件决定.

实现右键菜单插件调用示例如下:

```
<subnode name="右键菜单">
```

```
<com>iip://数据中心菜单插件/主场景
```

```
</com>
```

```
<comargs>C:\MapGIS 7.3\调阅数据菜单.xml
```

```
</comargs>
```

```
</subnode>
```

2.2 nip 协议

nip 协议用于在应用系统中实现功能插件库里的某项功能, 可以配置到系统菜单、工具条和右键菜单上. 它的结构如下, 一般是针对 iip 协议下具体的某个事件.

```
<nip://PlugIn Name/Main Scene/MainActive>
```

其中, 主场景下一级的 MainActive 仅仅针对某一个具体的事件. 无论是“主场景”的命名, 还是“MainActive”的命名是可以自主定义的.

插件调用示例如下:

(1) 实现要素类的属性统计.

```
<subnode name="属性统计">
```

```
<com>nip:// Catalog 工具/要素类/属性统计
```

```
</com>
```

```
</subnode>
```

(2) 实现角色用户登录应用系统.

```
<subnode name="登录">
```

```
<com>nip:// 数据中心权限编辑/角色用户管理/测试用户登陆</com>
```

```
<comargs>admin@sa</comargs>
```

```
</subnode>
```

2.3 wfip 协议

wfip 协议用于在应用系统中实现功能流程库里的某项功能, 可以配置到系统菜单、工具条和右键菜单上. 它的结构如下:

```
<wfip://Catalog/flow>
```

其中, Catalog 指示流程中的目录, flow 定义具体的流程名称. 流程目录是用户自定义的某一层目录的名字. 参数是流程中需要操作的对象的路径等, 场景参数会作为流程参数在流程中实现调用.

实现要素类属性统计的流程调用示例如下:

```
<subnode name="相交运算">
```

```
<com>wfip:// 空间分析工具/相交运算
```

```
</com>
```

```
</subnode>
```

2.4 event 协议

event 协议用于在应用系统中实现工作流的流程与数据的交互. 可以配置到系统菜单、工具条和右键菜单上. 它的结构如下:

```
<event ://path/interactive file>
```

其中, path 是交互文件存放的路径, 另外交互文件

是以 xml 的格式存放的, interactive file 描述交互的文件.

交互实现示例:

```

<subnode name="交互">
<url>event://MapGIS7.3/交互.xml</url>
</subnode>

```

2.5 aip 协议

aip 协议用于在应用系统中实现对操作系统中其他应用系统的调用,可以配置到系统菜单、工具条和右键菜单上.

```

<aip://Call Procedure Path / MainActive>

```

其中,aip 指示协议名称,Call Procedure Path 指示需要调用的安装程序的路径. 参数是应用程序要执行的某个文件或者其他的元素. 例如在参数中指定某个文件,可以通过应用程序直接操作这个文件.

应用程序调用实现示例:

```

<subnode name="应用程序调用">
<url>aip://c:\Program file\MapGIS7.3\
program\Apploder.exe</url>
</subnode>

```

2.6 脚本库协议

脚本协议用于在应用系统中对脚本功能进行调用. 协议格式如下:

```

<Srip://Script File Path ? op=Function Name>

```

其中,Srip 定义协议名称,Script File Path 定义脚本文件路径,Function Name 定义功能方法名. 例如:

```

<Srip://E:\MapGIS 7.0\Data Center\XXX.py ?
op=Mul>.

```

2.7 Web 服务协议

Web 服务协议用于在应用系统中对网络上服务的 Web 服务功能进行调用. 协议格式如下:

```

<Wsip://Web Service Network Address ? op=
Function Name>

```

其中,Wsip 定义 Web 服务协议名称,Web Service Network Address 定义 Web 服务网络地址,Function Name 定义功能方法名.

3 功能调用机制

3.1 功能仓库在数据中心中的调用机制

功能仓库中功能的调用由数据中心执行体来完成. 数据中心执行体是为数据中心功能仓库中各种异构功能资源设计的一个高度抽象的执行体,它能

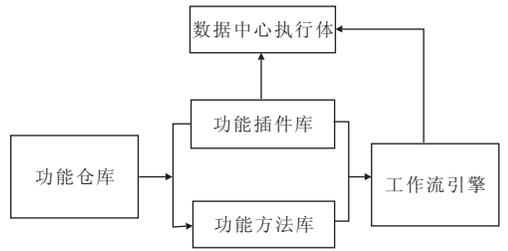


图 2 功能仓库在数据中心的运行机制

Fig. 2 Running mechanism of function warehouse in data center

够屏蔽各种引擎的差异,透明地执行各种功能资源. 数据中心执行体能够以统一的方式驱动功能插件库中的各种功能资源以及工作流引擎(图 2). 功能仓库以功能插件库和功能方法库两种方式提供,功能插件库包括可重命名的组件、直接调用的 API. 工作流引擎则负责执行基于插件库与方法库而创建的工作流插件.

3.2 工作流引擎

集各种异构功能资源于一体的功能仓库得以发挥其效能的关键就是工作流引擎. 一般而言,客户端定义一系列功能流程,通过工作流引擎调用功能驱动引擎,继而通过功能库服务在功能组件仓库中定位正确的原子功能资源. 功能仓库驱动引擎结构如图 3 所示.

工作流引擎,也称工作流运行时服务,为工作流实例提供运行环境,工作流引擎在运行时的主要任务是解释流程图,负责控制过程、节点、工作项的状态转换,处理过程、节点、工作项之间逻辑关系以及负责数据在各个步骤之间的传递. 本文的工作流引擎算法是采用活动状态推进模型,在运行时过程中定义活动有如下 3 种状态:(1)激活状态:活动被激活,指活动是等待实例化的就绪状态,当流程启动实

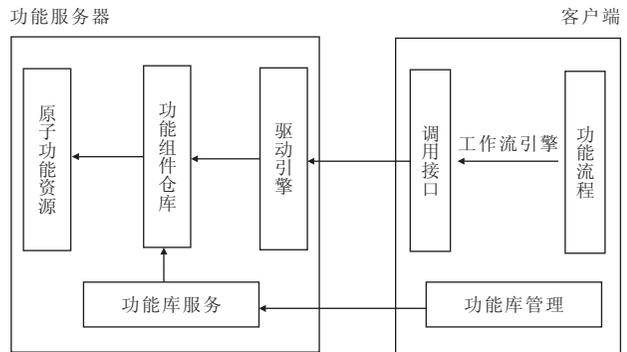


图 3 功能仓库驱动引擎结构

Fig. 3 Function warehouse driven engine structure

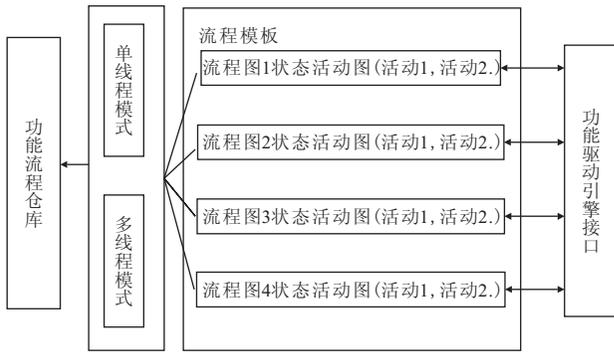


图 4 工作流与功能驱动引擎

Fig. 4 Workflow and function driven engine

例时,开始节点被激活;(2)实例化完成状态:活动只有被激活后才实例化,实例化就是完成活动上的所有功能;(3)实例化异常状态:指活动在实例化的过程中发生异常情况。

在运行过程中以各种动作来驱动活动实例状态的转变,当前驱节点处于实例化完成状态时,根据活动迁移条件的规则定义,决定是否激活后继节点,当迁移条件满足时,自动激活后继节点。

在功能仓库丰富的功能资源的支持下,流程模板提供地理信息处理任务的流程图的定制环境,一个独立的流程图包含一系列的活动,对应于一系列的功能资源。流程图通过功能驱动引擎的接口驱动,代表地理信息处理任务的功能流程可以作为一个完整的整体执行与处理(图 4)。

4 总结与展望

功能仓库提供了对各种不同类型功能资源的开放管理,与数据仓库集成新一代开发模式的集成开发平台——数据中心。在功能仓库的基础上能够迅速、方便地构建应用模型与业务系统。功能仓库降低了 GIS 应用系统的开发难度,为多领域应用系统的集成及功能复用提供手段,增强 GIS 应用系统适应需求不断变化的能力。功能仓库后续可以面向异构 GIS 功能管理主题域,以各个 GIS 平台功能库为基础,建设统一的功能仓库管理的规范与平台的概念模型,在此基础上构建应用模型与业务系统。采用工作流技术及可视化建模技术将不同 GIS 平台的空间分析及操作功能有机地融合在一起,实现 GIS 应用系统框架、表单和功能可视化智能搭建,并能够在多平台异构环境下的运行。该技术能够使现存的众多基于多种平台的异构 GIS 应用系统快速实现协

作,并根据实际业务快速构建综合应用系统,为支持多 GIS 系统协同工作提供良好的技术支撑。

References

- Ai, P., 2005. Research on the formal method of description for the flexible component composition and its application to the water resources domain (Dissertation). Hehai University, Nanjing (in Chinese).
- McIlroy, D., 1976. Mass-produced software components, 1968 NATO conference on software engineering. Van Nostrand Reinhold, Germany, 88—98.
- Pan, Y., Zhao, J. F., Xie, B., 2003. The research and development of the technologies in component library. *Computer Science*, 30(5): 90—93, 156 (in Chinese with English abstract).
- Wu, X. C., 2009. Data center integration development technology: the next generation GIS architecture and development model. *Earth Science—Journal of China University of Geosciences*, 34(3): 540—546 (in Chinese with English abstract).
- Ye, Y. Q., Zuo, Z. J., Chen, B., 2006. Orient-entity spatial data model. *Earth Science—Journal of China University of Geosciences*, 31(5): 595—599 (in Chinese with English abstract).
- Ye, Y. W., 2001. An active and adaptive reuse repository system. In: 34th Annual Hawaii International Conference on System Sciences (HICSS-34)-Volume 9, Hawaii, 9065.
- Zhou, S. P., Li, H., Du, X. P., 2006. Building the topology of spatial entities. *Earth Science—Journal of China University of Geosciences*, 31(5): 590—594 (in Chinese with English abstract).

附中文参考文献

- 艾萍, 2005. 构件柔性组装描述的形式化方法研究及其在水利领域的应用(博士学位论文). 南京: 河海大学.
- 潘颖, 赵俊峰, 谢冰, 2003. 构件库技术的研究与发展. *计算机科学*, 30(5): 90—93, 156.
- 吴信才, 2009. 数据中心集成开发技术: 新一代 GIS 架构技术与开发模式. *地球科学——中国地质大学学报*, 34(3): 540—546.
- 叶亚琴, 左泽均, 陈波, 2006. 面向实体的空间数据模型. *地球科学——中国地质大学学报*, 31(5): 595—599.
- 周顺平, 李华, 杜小平, 2006. 空间实体的拓扑构建. *地球科学——中国地质大学学报*, 31(5): 590—594.