

文章编号: 1000-050X(2001)04-0303-07

文献标识码: A

GIS 功能部件库和功能集成

徐志红¹ 边馥苓² 孟令奎²

(1 浙江省土地勘测规划院, 杭州市青云街 40 号, 310003)

(2 武汉大学遥感信息工程学院, 武汉市珞喻路 129 号, 430079)

摘要: 从 GIS、计算机、网络和数据库的发展出发, 提出了 GIS 功能部件库, 利用库进行部件的管理, 并从组织结构、分类、检索、评价等方面予以评述, 在此基础上提出了“部件生产”和“部件集成”开发方式。

关键词: GIS 功能部件库; 功能部件; 部件集成

中图法分类号: P208; TP311.13

当今, GIS、计算机及数据库技术都有了长足的进展。横向, 开放式数据库链接 ODBC 解决了关系数据库的链接, Helical 超空间编码 (helical hyperspatial code, HHCODE) 解决了多维数据的存储和管理, 空间数据管理及分析、查询的部分功能均逐步从传统的 GIS 软件中转移到通用的数据库管理系统; 纵向上, 在部件对象模型 COM、通用对象请求代理体系 CORBA 和万维网的支持下, GIS 软件逐步以客户机/服务器为模式对传统的系统实现分解和重组, 如图 1 所示。

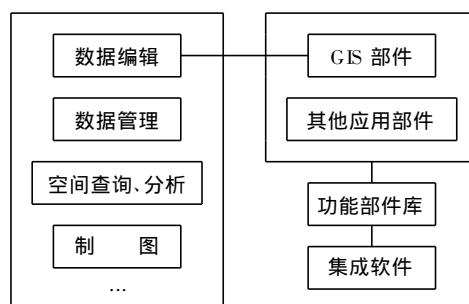


图 1 GIS 应用软件集成

Fig. 1 Integration of GIS Application Software

目前, 已开发出的 GIS 功能部件有如下几种: ESRI 的 MapObjects、Sylvan Ascent 的 Sylvan-Map/OCX、MapInfo 的 MapInfo MapX、Emap 的 GIS OCX Control 等。随着 GIS 功能部件的增多, 如何对它们实施有效的管理, 以方便用户提取和利用, 本文提出了 GIS 功能部件库。

1 GIS 功能部件库

1.1 概述

GIS 功能部件库是 GIS 各功能部件的集合, 也可称为 GIS 功能仓库。这个功能库是由 GIS 各功能部件按一定方式组织的, 它们可以集中存储, 也可以按一定模式在局域网甚至万维网上分布式存储, 正如其他数据库一样, GIS 功能部件库也提供管理、检索等功能, 为用户从库中提取功能部件提供方便, 同时, 由于功能部件不同于数据库中数据的精确匹配, 且 GIS 同时处理空间数据和属性数据的特点, 笔者提出了适用于 GIS 软件开发的 GIS 功能部件库, 这个库中不仅包含 GIS 的基本功能, 如查询、统计、拓扑分析等, 还包括各专业领域的专业功能, 如空间信息量测与分析、多要素综合分析等。这些功能与 GIS 结合紧密, 是 GIS 功能库中不可缺少的部分。

1.2 部件属性描述

为了对 GIS 各功能部件进行管理, 必须首先提取该部件的属性描述, 根据属性描述对各部件进行分类索引。由于 GIS 的地理特性以及空间属性和非空间属性结合的特性, 在 GIS 部件描述中包括的部件属性如表 1 所示。

该属性表中针对 GIS 中空间数据的特点, 提出了 GIS 部件描述表中必备的描述。部件接收的空间数据格式包括: 空间数据的坐标系统、投影系统、存储方式、模型及输入、输出格式; 非空间数

表1 部件属性描述表

Tab. 1 Description of Component Attributes

属性	描述
UID	部件惟一标识符
N AM E	部件名称
KEY WORDS	部件功能描述关键词
OBJECTS	部件的对象
SPATIAL DATA FORM	部件接收的空间数据格式
ASPATIAL DATA FORM	部件接收的非空间数据格式
APPLYING OBJECT	部件作用对象
DOMAIN	部件作用领域
APPLYING REGION	部件应用场所
REQ UIREMENTS	有关该部件任何特别需求的信息
ERRORS	错误处理及部件中出现的例外的信息
AUTHOR	部件作者
DATE CREATED	部件完成日期
LAST UPDATE	部件最后一次修改日期
AIDING SOFTWARE	辅助软件 [*]
DOCUMENTATION AND TESTING	包含有关该部件的可用文档的描述及测试实例描述

* 指在MGE, Arc/Info 等GIS基础平台上的开发。

据格式包括属性数据的存储方式、输入、输出格式等;部件作用对象区分属性和空间数据以及不同的数据支撑格式;部件作用领域针对GIS中相同或相近领域具有类似功能需求,提出针对领域的功能部件的描述。同时,由于很多软件的开发是在基础GIS平台上的,因此,辅助软件的描述也是必需的。

1.3 分类

正常地,当用户使用功能库时,首先要存取可能包含所要查找部件的某个库,然后根据该库系统理解的词汇提出被查找部件的描述。同时,用户选择部件时,希望检索到最少修改的最佳候选部件或提出相近部件。所以,建造GIS功能库的首要任务是对功能部件进行分类,使得同一类的成员共享同一组性质。

分类模式有如下要求:分类信息要包括从使用者角度反映的部件之间的关联信息;分类模式能应用于不同精度的重复利用;分类模式不能太复杂。除考虑这些分类基本因素外,还需考虑GIS的特殊性,即它的空间信息和非空间信息处理的差异。由于目前空间信息的处理多是基于目前较成熟的GIS基础软件,而非空间信息处理多依赖于关系数据库管理系统;空间信息和非空间信息之间的关联,不同软件的连接方式也不完全一致,且随着GIS应用范围的扩大,各专业领域又会相应地不断产生一些新的需求。综合以上因素,笔者采用类似于树的功能分面模式和枚举模

式结合方式对这些功能部件进行分类。

GIS的功能描述如前所述,按GIS的功能可把GIS功能库分为输入、编辑、显示、查询、分析、输出、管理、其他等8大类,每类可按空间数据和非空间数据处理再进一步细分。

采用传统的列举方案,随着商业上可利用软件的激增,许多目录对于一些大类,展示出很长的清单,同时还展示出一些没有清单的类。因此,笔者还提出了分面方案,该方案提出一种基于标准专门名词词汇表的部件描述格式,给分面加上引用次序,并为各分面中的专门名词之间的概念距离提供一种度量标准,这种度量帮助选择密切相关的部件。

部件描述通常由描述功能(它做什么)和描述环境(它在何处运行)两部分组成。如果一个描述既要用作分类码,又要用作检索关键字,那么就必须简明。典型的命令语句是用三元组<动作、对象、动因>刻划的。为了表示部件功能,笔者稍作修改,表示为<功能、对象、媒体>,使之更好地描述部件所为。其中,“功能”是部件履行的特定原始功能的名称,该项功能可据GIS的8大功能分类及子类进行设置,从而提供相似的功能聚类,方便检索;“对象”是指部件操纵的对象,视GIS处理空间信息及非空间信息的不同,可为纯空间信息(点、线、面)、网络空间信息、复杂空间信息、地面高程信息、图像信息及属性数据等;“媒体”是指执行动作时所用的实体,即该功能的支持结构,视GIS采用的基础平台及数据组织方式不同,可为不同的媒体,如Arc/Info的coverage、ArcView的shapefile、Microstation的DXF、Geomedia的数据仓库、全关系数据库存储的空间及属性信息以及单纯的属性信息。同样,由于部件应用范围及部件所属类别不同,用<作用领域、应用场所>来描述环境的差异。作用领域指功能上可鉴别、与应用无关的模块,通常包含一个以上部件(如辅助决策分析等);应用场所描述与应用相关的活动,明确定义部件的应用领域(如道路改造、洪水灾情预测等)。

部件分类就是选择最能描述部件的五元组,如<缓冲区,线,矢量数据结构图形文件,缓冲区分析,道路改造>。

在这种描述中存在着一种问题,就是用不同的专门名词元组描述部件会因为同义词而形成不同的主字码,如(delete, line, file)和(remove, line, file)。

这可能是同一部件的不同描述。为避免歧义

及多义,笔者引入词汇控制表。词汇控制表就是将最能描述该概念的名词作为代表性的专门名词,其他近似词作为它的扩展,并将各专门名词与GIS的八大功能类联系起来,从而形成相似功能的聚类,如表2所示。

表2 词汇控制表

Tab. 2 Glossary Control Table

专门名词	隶属于	近义词
Delete	Edit	Remove/ Erase/ Cut/ Throw
Buffer	Analysis	Expand/ Increase/ Swell/ Add
...

同时,为了度量一个分面各个专门名词之间的接近性,引入了概念接近性这个概念,通过概念图表示部件接近程度。专门名词为非环状有向图的叶子,结点就是表示联系两个或两个以上专门名词的一般性概念的超类,边上的权由用户指定,某个专门名词同超类的关系越密切权值就越小,如图2所示。

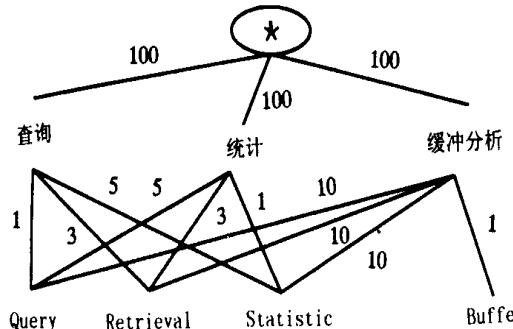


图2 分面加权概念图

Fig. 2 Facet Weight Concept Figure

远近程度的度量主要发生在检索过程中。如果查询特定的专门名词与集合中可以利用的任何描述都不匹配,那么,系统就试用最为密切相关的专门名词去检索由密切相关的专门名词组成的描述。

1.4 GIS 功能部件库的组织

如其他数据库一样,需要一个数据库管理系统(DBMS),而GIS功能部件库又不同于普通的数据库,它的主要目的是功能部件的提取和复用。如何有效组织GIS功能部件库,参照国际上相对成熟的Intermetric研制的可复用软件库(reusing software library,简称RSL),把整个GIS功能部件库划分为GIS功能部件库和4个模块,其组成如图3所示。其中,库管理模块用于管理GIS功能部件库,将检验合格的GIS功能部件,按GIS的输入、编辑、显示、查询、分析、输出、管理、其他

8大类并区分空间数据处理及非空间数据处理输入到GIS功能库中,同时提取部件的属性描述。用户查询模块为查找具有特定属性的部件和生成有关报告提供一个菜单驱动的接口,可以通过查询模块构造查询功能部件的主字码(即描述部件合法的专门名词)。对于枚举模式,可根据GIS功能建立索引,这样的索引重复率极高,表现了相似功能部件的聚类,适用于基于聚类的检索方式。而对于分面模式,采用前面所述的<功能、对象、媒体、作用领域、应用场所>五元组作主关键字进行索引,在检索中,由各分面中挑选的专门名词组成合法主字码 $d_{\sigma 1}$,若集合中没有与 $d_{\sigma 1}$ 相匹配的,那么,就通过计算相应概念图中的距离,选出那些联系紧密的专门名词,形成新的主字码 $d_{\sigma i}$, $2 \leq i \leq n$,随后根据匹配检索与 $d_{\sigma i}$ 描述的部件联系紧密的部件。功能部件检索和评价模块是一个排序系统,它通过对各功能部件与用户需要的功能描述进行评价,从而得到最满足用户需求的功能部件排序。为帮助用户选择合适部件,引入了Score评价体系。

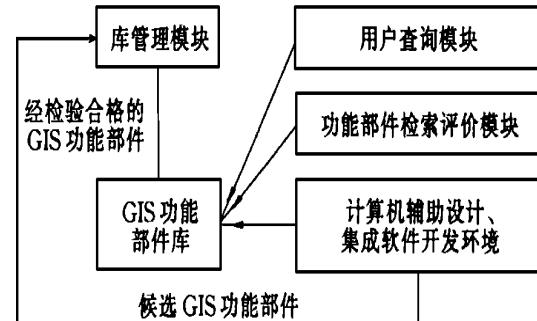


图3 GIS功能部件库组织

Fig. 3 Organization of GIS Functional Component Library

Score评价的中心是辨别软件应用领域,因为在检索复用部件时,应用领域决定了哪些功能属性和性质属性最有意义,通过询问用户对各部件属性的相对重要性,给出符合用户需求的部件列表。考虑到GIS处理空间信息及非空间信息的不同,把部件处理的信息类型列为评价的第二个因素。以规划、土地管理、房屋管理、城建为例,如在做道路Buffer分析时,规划需要知道各地块的影响面积,土地管理需要知道各宗地类型的影响面积,房屋管理需要知道各户型的影响面积,而城建需要知道各户拆迁面积,这些都是通过GIS中对空间信息的Buffer分析之后进行统计得出的。因此通过相近领域的指定功能及处理数据的类型去检索所需功能部件,就可帮助用户选择出最合适复用的软件。

Score 评价过程如图 4 所示。

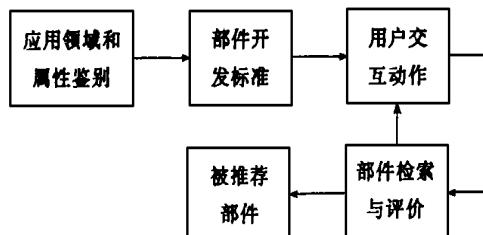


图 4 Score 评价概貌

Fig. 4 General Picture of Score Evaluation

计算机辅助设计集成软件开发环境指有利于 GIS 功能部件进行集成开发的各种软件环境, 如 CASE 等。

1.5 分布及分布字典

随着计算机技术及网络的发展, 单机独立运行越来越少, 渐渐都走向网络化。网络化计算机主要有以下几种系统配置: 对等服务型系统配置; “无盘”型系统配置; 客户机/服务器系统配置; 网络配置。

各种服务在网络中分布存在, 可以通过网络对该服务进行授权使用。GIS 功能部件库可按上述各种配置分布于提供功能服务的各工作站, 如何迅速查找所需部件的位置, 这就需要设置分布字典, 以便于用户查找位于本机或网络上不同位置所需的功能部件。分布字典的功能如图 5 所示。

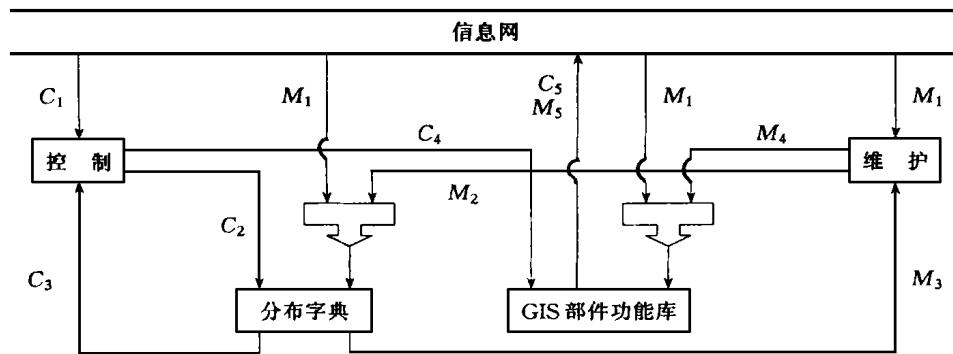


图 5 分布字典功能示意图

Fig. 5 Sketch Map of Distributed Dictionary

分布字典中记录的是功能库中的分类信息和在网络中的确切位置, 本地用 0 表示, 而远程则用 1 表示, 并指出确切网络中的位置。当控制信息 (C) 和维护信息 (M) 发出的时候, 首先访问分布字典, 从分布字典中获得该类功能部件库的位置, 然后采用基于功能的聚类查询或五元组查询访问功能部件库, 获取所需 GIS 功能部件。

2 功能集成

2.1 集成类型

GIS 的功能集成可分为两种类型: ①采用从下到上的方式。从功能部件库中已存在的功能部件出发, 根据已有的功能部件按一定模式组装成新的应用系统。它只根据现有部件进行组装, 而不考虑用户的需求。②采用由上到下的方式。从用户需求出发, 根据用户需求, 从 GIS 功能部件库中选择所需的功能部件。若功能部件完全满足需要, 则采用“拿之即用”的方法; 若功能部件不完全符合需要, 则需进行局部修改以适应用户需求。这样, 就可以构造出完全满足用户需求的应用系统。第一种类型集成出的应用系统往往不实用,

不能完全满足用户需求; 而第二种类型的集成则完全从用户需求出发, 满足用户需求。

2.2 集成模式结构

在一般的数据库管理系统中, 比较典型的集成模式结构是由 Tschritzis 和 Klug 于 1978 年提出的三级模式结构, 即 ANSI/SPARC 三模式结构, 包括概念模式、内部模式和外部模式。概念模式描述数据库的概念或逻辑结构及其之间的关系; 内部模式描述概念模式内逻辑数据结构的物理特征, 包括存储介质上记录和索引的位置、类型以及逻辑记录之间的物理表示; 外部模式对用户所需存取的部分数据库内容进行描述。Abel 等人曾用三模式讨论过 GIS 集成, 尽管三模式结构明确了设计、实现和使用等各个不同过程所对应的模式, 但是由于 GIS 中空间数据和非空间数据结合这一特性, 这里采用基于三模式的功能集成五级模式结构^[8], 如图 6 所示。该参考结构包括用户模式、集成模式、空间与非空间表示模式、表示模式、局部功能模式。

其中, 局部功能处理系统 (local functional processing system) 因各个系统的功能不同而各异, 可以是建立在不同的 GIS 平台 (如 Arc/Info、

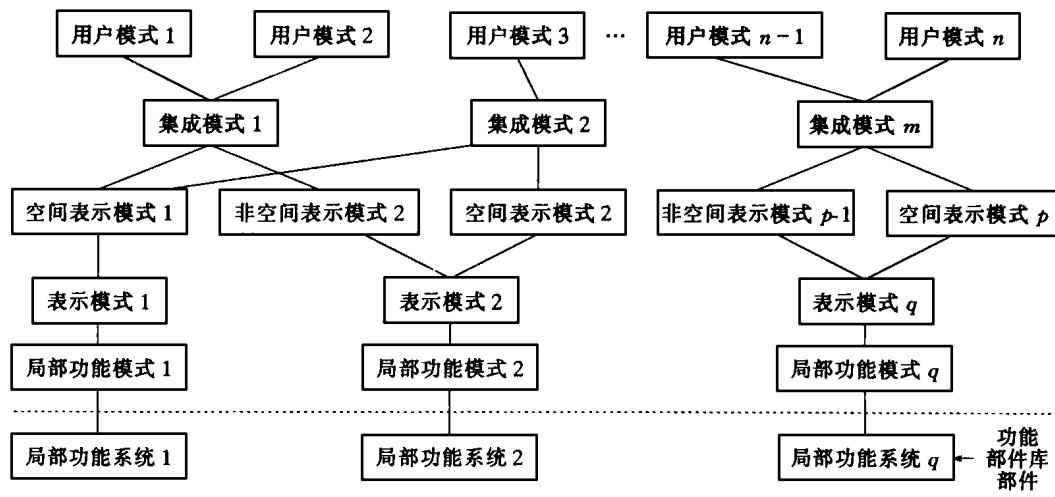


图 6 功能集成五级参考系统结构

Fig. 6 Five-level Reference System Construction of Function Integration

MGE、GENAMAP 等)之上的应用系统,也可以是 DPS、图像处理系统、专家系统、数据库管理系统等,它是由存取器和相应的数据库及 GIS 功能部件库等构成的专用功能处理系统。局部功能模式 (local functional schema) 是局部功能处理系统各功能的一种概念描述,反映了各局部功能与对象数据之间的逻辑关系。不同的系统具有不同的局部功能模式描述方法,包括模式目标及其之间关系的描述等。局部功能模式的表示可与 GIS 功能部件库的索引相对应,以方便索引和查询。表示模式 (express schema) 是利用某种集成标准或公共表示模型 (common express model, 简称 CEM) 表示的一种功能和数据语义一致性模式,即所有集成对象,无论其本身的局部功能模式有多大差异,它们在表示模式上应取得一致性的表示,这种一致性主要包括模式目标描述方法、语义规范格式、功能和数据存取的表示、继承关系的表示等,这是达到功能集成的第一步。空间与非空间表示模式 (spatial and aspatial schema) 是表示模式的子集。对于不同的用户,并非表示模式中的所有模式目标都是必需的。从功能优化角度看,当一个集成系统有多个完全一样的功能模块时,为了减轻系统负荷 (如网络传输) 和避免存取竞争,并考虑到用户要求的时效性以及所授予的存取权限,也需要对各个参与集成的表示模式进行某种限制,这种限制的最终目标是达到全局优化。根据不同的优化策略和限制条件,可对表示模式的模式目录进行筛选 (过滤),从而产生不同的空间与非空间表示模式,该级别的模式目标将全部参与集成。集成模式 (integration schema) 是多个空间及非空间表示模式的集成。功能集成系统内可以有多个集成模式,每个集成模式对应一类

集成用户,每个集成模式还可为多个集成用户提供服务,因而可以针对不同的用户需求提供合适、有效的功能服务。由图 6 可看出,各集成模式及其下级模式均处于分布式计算环境下,因而在一个以网络为连接纽带的大型系统中,通过合理组织便可形成一个并发程度高、功能服务迅速的高效集成化系统。用户模式 (user schema) 是用户使用的模式定义。从用户角度考虑,应为不同用户规定一个符合其使用习惯的模式,这有助于对用户的需求变化迅速作出反应,并使得增加或删除一个用户模式成为一项十分简单的活动。

转换器分为功能模式转换器和数据格式转换器,分别实施对功能模式和数据格式的转换,这种转换对终端用户而言是透明的。通过转换器,完成从局部功能模式到表示模式的转换;过滤器主要实现对模式目标的子集的若干限定和过滤,包括进行功能模式和数据类型的语法检查、语义完整性检查以及存取权限的检查等。通过过滤器,完成从表示模式到空间/非空间表示模式、集成模式到用户模式的转换;集成器完成两种功能模式分解和功能模式集成。功能模式分解负责把一种功能模式分解为由下级处理器能够正确接收的功能子模式,它提供了单一模式与多模式的映射功能,是功能集成中分布式处理、并发控制、并行传输以及对用户透明存取的关键。功能模式集成负责把下级各处理器产生的各种模式目标进行归并处理,为满足功能模式的功能要求提供完整、一致的模式表示,并消除不必要的冗余功能。通过集成器,完成从空间/非空间表示模式到集成模式的转换。

3 实 例

为了验证部件式开发的便利性,作者在基础 GIS 中部分进行了尝试,验证了其有效性和实用性。GIS 软件环境是 Intergraph 公司的基础 GIS 软件包 MGE, 包括 Microstation 95, MGE Basic Nucleus (MGMUC), MGE Basic Administrator (MGAD), MGE Base Mapper (MGMAP), Relational Interface System (RIS)。开发环境是 Microsoft Visual Basic 5.0 企业版和 Microsoft SQL Server 6.5。

在该实例中,应用了本文提出的五级模式结构。局部功能处理系统是基于 MGE 开发的中山市规划管理信息系统,在此基础上,提取局部功能模式,即对图形、属性的编辑功能,描述为 LFS (edit, graphics & attributes, MGE, urban planning), 通过转换,形成统一的表示模式,即 ES (edit, graphics & attributes, MGE graphics & attributes, urban planning)。由于本系统采用已存在的 CAD 数字化图形,缺少属性及确切分层信息,故本系统重点在于给已有图形划分 Category 并加属性,因此在表示模式的基础上进一步提取属性编辑功能,形成非空间表示模式,即 ASS (edit, attributes, MGE attributes, urban planning)。为了便于集成,在已有非空间表示模式的基础上,采用部件式开发方式改造原模块,形成集成模式,即统一的五元组表示方式: IS (edit, attributes, MGE attributes, edit, urban planning)。所改造部件中表的入口是根据不同层定义不同值,即据该部件的 Category 和 Feature 给出不同的入口值,通过 Feature 关联图形和对应的属性,进行操作。至此,该功能部件可参与相关系统集成。笔者直接在基础 GIS 中进行应用,从而大大提高了开发速度。同

时为了维护方便,通过过滤,提供更简便的用户模式,即 US (edit, attributes), 方便用户对系统的维护。

4 结 论

“部件开发”和“部件集成”是地理信息系统开发方向,而部件功能库的提出为这种分工合作式开发提供了基础。只有通过 GIS 功能部件库的有效管理,这些功能部件才能被充分利用,达到资源极大共享。

参 考 文 献

- 1 徐正权. 软件复用方法与技术. 武汉: 华中理工大学出版社, 1998
- 2 龚健雅. 当代 GIS 的若干理论和技术. 武汉: 武汉测绘科技大学出版社, 1999
- 3 汤勤. 部件对象模型及其相关技术. 见: 龚健雅. 当代 GIS 的若干理论和技术. 武汉: 武汉测绘科技大学出版社, 1999
- 4 孟令奎. 地理信息系统功能集成研究. 武汉测绘科技大学博士后工作报告. 武汉, 1996
- 5 Tracz W. Reusability Comes of Age. IEEE Software, 1987, 4(4): 6~8
- 6 Prieto D R, Freeman P. Classifying Software for Reusability. IEEE Software, 1987, 4(1): 6~16
- 7 David J A. Towards Integrated Geographical Information Processing. Geographical Information Science, 1998, 12 (4): 353~371
- 8 Wojtek K. Component-Based Software Engineering. IEEE Software, 1998, 9/10. 34~36

作者简介: 徐志红, 博士生。现从事 GIS 功能集成、时态 GIS 及知识挖掘的研究。参与温州市地籍管理信息系统、诸暨市资源管理信息系统查询子系统、三峡库区管理信息系统、中山市基础地理信息系统等总体设计。

E-mail: xzh-xzh@163.net

GIS Functional Component Library and Function Integration

XU Zhihong¹ BIAN Fulong² MENG Lingkui²

(1) Zhejiang Provincial Institute of Land Surveying & Planning, 40 Qingyun Street, Hangzhou, China 310003

(2) School of Remote Sensing and Information Engineering, Wuhan University, 129 Luoyu Road, Wuhan, China 430079

Abstract: In recent years, with the emergence of commercial distributed-computing infrastructures such as DCOM and CORBA, and the increasing need for interoperability of independently developed software “chunks”, components and component-based software engineering (CBSE) have gained substantial interest in the software community.

CBSE brings architectural thinking to the center of the software development process. It clearly separates infrastructure from system logic and helps to cope with system complexity. It helps organize large-scale development and, most importantly, makes system building less expensive. Compared with other application development, GIS has its own characteristics. As we know, only huge and complete GIS software can be released as GIS platform, at the same time, most of GIS application development are based on certain GIS platform, while only a few functions of GIS platform are used in the application development, which leads to great waste in resources and funds. Until now, many application development are united as “workshop”, on the one hand, the programmers developed the function/module from the bottom according to one user’s request; on the other hand, many programmers developed the same function/module according to the specific requests, thus led to repeated development of the same function. And this “workshop” development mode needs more time and more funds. The use of CBSE will avoid such disadvantages. As component technology has begun to impact the process of software development, it turns the traditional developing mode of “workshop” to the new mode of “component production” and “component integration”.

Now there are many GIS functional components: Mapobjects (ESRI corp.), MapX (Mapinfo corp.), Geomedia components (INTERGRAPH corp.), etc. With the development of component technology, there will be more and more components. How to manage these components and how to make good use of these components are one topic of this paper. The paper brings forward the concept of GIS functional component library, which is used to manage the components and make it convenience for people to select the proper component. Considering GIS particular characteristic—spatial information, GIS functional component library is discussed from its organizational structure, access, retrieval, evaluation, etc. Distinguished from other libraries, the description contains more information of GIS component, including spatial data form, applying object, domain and aiding software, etc. Spatial data form means projection, coordinate and data format, etc. Applying object distinguishes attribute information from spatial information. Domain emphasizes the component reuse. According to GIS functions, we classified the component into eight categories: input, edit, display, query, analysis, output, management and others, then they can be further divided according to spatial information or aspatial information. Thus, we obtain clustering of the similar function. After that, we take five-element-group description, which includes function, applying object, support media, domain, applying region, and facet weight concept figure to distinguish similar function and select proper function.

Based on the GIS functional component library, with consideration to GIS spatial and aspatial characteristic, five-level integration construction was proposed. It includes user schema, integration schema, spatial and aspatial schema, express schema and local functional schema. Finally, an instance was provided as the example of “component production” and “component integration”.

Key words: GIS functional component library; functional component; component integration

About the author: XU Zhihong, Ph D candidate. Her research directions are GIS function integration temporal GIS and data mining. She is engaged in the construction of Zhejiang land information system. She has taken part in Wenzhou cadastral management information system, query subsystem of Zhuji land resource management information system, general design of the Three Gorge management information system, Zhongshan fundamental geographic information system, etc.

E-mail: xzh-xzh@163.net