基于GIS系统的空间查询语言

徐承志[1] 许承瑜[1,2] 钱铁云[1]
(武汉大学软件工程国家重点实验室 武汉 430072)[1]
(加州大学尔湾分校电气工程和计算机科学系 加州 92617)[2]

摘 要 目前,主流的空间数据查询语言都是在SFA SQL或SQL/MM Spatial这两大国际标准的基础上进行扩展的。然而,这两大标准对于空间查询和空间分析都是函数式的,所以当查询条件增多时,其复杂的查询表达式既不适合普通用户使用,也不利于提高查询的效率。提出了一种基于GIS系统的空间查询语言SQDL-G,将空间谓词表示为空间运算符,将子查询结构引入查询表达式中,并在ArcGIS平台上建立了该语言的执行引擎。实验表明,该语言表达灵活,结构清晰,易于被用户接受。

关键词 空间数据库,空间查询语言,SQDL-G,ArcGIS
中图法分类号 TP311 **文献标识码** A

New Spatial Query Language Based on GIS

XU Cheng-zhi[1] Phillip SHEU Chen-yu[1,2] QIAN Tie-yun[1]
(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)[1]
(Department of Electrical Engineering and Computer Science, UC Irvine, California 92617, USA)[2]

Abstract At present, the mainstream of spatial query languages are extended from SFA SQL or SQL / MM Spatial, two international standards. All these two standards for spatial query and spatial analysis are function style. When search conditions increase, function style language will become too complex to be used by normal user, and too inefficient to execute. In this paper, a new spatial query language, Semantic Query Description Language for Geography (SQDL-G), based on GIS was proposed. The language converts spatial predicates into spatial operators and introduces sub-query structure. An execute engine of this language was built up based on ArcGIS platform. Experiments showed that this language has an expression of a flexible, clear structure, and easy to user acceptance.

Keywords Spatial database, Spatial query language, SQDL-G, ArcGIS

现实世界中的任何事物都有空间位置的属性,包含这些位置信息的数据库被称为空间数据库,专门用来分析和处理空间数据的系统被称为地理信息系统(Geographic Information System,简称GIS)。除了具有一般数据库的主要特征外,空间数据库还具有综合抽象、非结构化、分类编码和复杂性与多样性等特征。正是由于上述特征,使得空间数据库在信息描述、数据管理、数据操作和服务应用上都与传统数据库存在差异。空间数据不同于关系数据,它们没有严格的二维关系,因此,对于传统的GIS系统大多采用对空间属性和标量属性分别进行管理。对于空间位置和空间关系的分析大多采用面向过程的方式得到分析结果。对于标量属性的查询常采用关系数据库的结构化查询语言SQL。虽然关系查询语言有其固有的缺陷,如不支持复杂数据结构、不能图形化显示、不支持空间查询和分析等,但是鉴于SQL已经非常成熟且被广泛接受,因此在SQL的基础上进行扩展将是分析和处理空间数据的一个趋势[1]。目前有多家公司致力于空间数据库的研究,如ESRI(美国)、MapInfo(美国)、MapGIS(中国)、SuperMap(中国)、武大吉奥(中国)等。

1 当前主流空间数据查询语言国际标准

空间数据库标准在数据库空间扩展过程中发挥着重要的作用。大部分空间数据库厂商在设计查询语言时都遵循国际标准。其中较为流行的两个标准是SFA SQL和SQL/MM Spatial[2]。

开放地理空间信息协会(OGC)和国际标准化组织推出的SFA SQL规范(ISO 19125),定义了函数访问接口,依据地理几何对象模型,提供在不同平台下(OLE/COM, SQL, CORBA)对简单要素(点、线、面)的发布、存储、读取和操作的接口规范说明。SFA的通用体系架构规范描述了简单要素地理几何对象模型,以及地理几何对象的不同表达方式和空间参考系统的表达方式[3,4]。

国际标准化组织/国际电工委员会第一联合技术委员会/

到稿日期:2009-12-10 返修日期:2010-02-21 本文受国家973计划资助项目(2007CB310801),国家自然科学基金资助项目(60873007),高等学校学科创新引智计划(B07037)资助。

徐承志(1976-),男,博士生,讲师,主要研究方向为语义计算、空间数据挖掘等,E-mail:xcz911@gmail.com;许承瑜(1956-),男,教授,主要研究方向为语义计算、大型实时防御知识系统等;钱铁云(1970-),女,副教授,主要研究方向为数据挖掘、信息检索。

数据管理和交换分技术委员会(ISO/IEC JTC1/SC32)发布了开发 SQL 多媒体和应用程序包(SQL/MM)标准,其中第三部分 Part 3:Spatial(ISO/IEC 13249-3)定义了空间数据类型及操作,解决了如何存储、获取和处理这些空间数据[5]。该标准除了定义几何对象模型外,还定义了角度和方向类型对象模型。SQL/MM Spatial 标准基于 SQL99 标准,其描述的扩展环境与 SFA SQL 中支持的 UDT 扩展环境一致。

总体来说,两个标准起源和发展时间相当长,SQL/MM 侧重于 Geometry 扩展空间类型的实现,在这方面涵盖的面、涉及的内容要比 SFA SQL 更宽泛、丰富。而 SFA SQL 兼顾了预定义数据类型实现模型,在文本标注类型上比 SQL/MM 略胜一筹。从长远来看,由于行业的需求逐渐统一,两者有相互融合的趋势。

2 以 Oracle Spatial 为代表的空间数据查询语言

目前已有许多数据库厂商提供对 SQL/MM 和 SFA SQL 标准的支持,如 Oracle Spatial,DB2 Spatial Extender,Informix Spatial DataBlade,SQL Server 2008 Spatial Katmai 和 MySQL Spatial Extension 等。其中,Oracle Spatial 是杰出的代表,它同时支持 SQL/MM 和 SFA SQL 两个标准。Oracle Spatial 的查询功能主要通过空间算子(Spatial Operator)和空间函数(Geometry Function)来实现[6],通过下面简单介绍,可以了解它们的主要功能和运算能力。

2.1 空间算子

空间算子是返回布尔类型变量的函数,只能运用在 WHERE 子句中。Oracle Spatial 包含 5 个空间算子,其中最重要的是 SDO_FILTER 和 SDO_RELATE。

SDO_FILTER(geometry1,geometry2,params)

通过空间索引,根据给定的几何要素检索出具有空间相互关系的空间对象。这里的空间关系是指几何不分离,即 Non-disjoint。如果两个对象有关系就返回"TRUE",否则返回"FALSE"。例如,从 Polygons 表中选出一组对象并显示其 gid 值,这些对象与 query_polys 表中 gid 值为 1 的对象存在联系。

```
SELECT A. gid
FROM Polygons A,query_polys B
WHERE B. gid = 1 AND
    SDO_FILTER(A. geom,B. geom,querytype = 'WINDOW') =
'TRUE';
```

SDO_RELATE(geometry1,geometry2,params)

通过空间索引,根据给定的几何要素(如一个多边形)检索出与其有特殊空间关系的几何要素,并且空间关系可用符号"+"实现 OR 运算。例如,从 Polygons 表中选出一组对象并显示其 gid 值,这些对象与 query_polys 表中 gid 为 1 的对象有"inside"或"coveredby"关系。

```
SELECT A. gid
FROM polygons A,query_polys B
WHERE B. gid = 1 AND
      SDO_RELATE(A. Geometry,B. Geometry,
      'mask=inside+coveredby querytype=WINDOW')='TRUE';
```

除了上述两个算子,还有与距离相关的 3 个算子,它们分别是 SDO_NN_DISTANCE,SDO_WITHIN_DISTANCE 和 SDO_NN。

2.2 空间函数

空间函数能够返回多种数据类型,可以用在 SELECT 子句和 WHERE 子句中。Oracle Spatial 中的空间函数多达 19 种,下面介绍有代表性的几个函数。

SDO_GEOM. RELATE(geom1, dim1, mask, geom2, dim2)或 SDO_GEOM. RELATE(geom1,mask,geom2,tol)

测试两个对象间的某种空间关系是否成立,并且空间关系可用符号"+"实现 OR 运算。例如,测试 New Jersey 州内的每个郡与该州是否具有'covers'的拓扑关系。

```
SELECT c. county,sdo_geom. relate(s. geom,'covers',c. geom,0. 05)
FROM states s,counties c
WHERE s. state='New Jersey' AND s. state=c. state;
```

SDO_GEOM. SDO_AREA(geom,dim [,unit])或 SDO_GEOM. SDO_AREA(geom,tol [,unit])

返回一个二维多边形的面积,例如计算出 New Jersey 州每个郡的面积。

```
SELECT c. state,sdo_geom. sdo_area(c. geom,0. 05)
FROM counties c
WHERE c. state='New Jersey';
```

SDO_GEOM. SDO_BUFFER(geom, dim, dist [, params])或 SDO_GEOM. SDO_BUFFER(geom,dist,tol [,params])

返回目标对象的缓冲区,例如查找在'I 170'号高速公路的 0. 5 公里缓冲区内的城市。

```
SELECT c. city
FROM cities c,highways h
WHERE h. id='I 170' AND sdo_relate(c. location,sdo_geom. sdo_
    buffer(h. geom, 0. 5, 0. 05),'mask = anyinteract querytype =
    WINDOW')='TRUE';
```

除了上述空间函数外,还有很多关于空间度量和空间分析的函数,如计算长度、计算拓扑操作等。

3 函数型空间算子存在的问题

通过前面的例子可知,基于 SAF SQL 和 SQL/MM 的 Oracle Spatial 中包含丰富的空间查询操作。但是,无论是空间算子还是空间函数都是函数型的,当它们以查询条件的形式出现在 WHERE 子句中时,非常类似于传统 SQL 中的多表连接操作,如表 1 所列。

表 1 连接操作比较

传统 SQL	Oracle Spatial(简化示意)
SELECT a. id	SELECT a. id
FROM A a,B b	FROM A a,B b
WHERE a. geom = b. geom	WHERE relate(a. geom,b. geom,'equal')

在处理简单空间查询时,连接操作是没有问题的,但是当查询条件变多时,这种结构就暴露出一些缺陷了。下面用一个包含两个查询条件的例子来说明这种缺陷。

例 1 查找 A 中某个对象的 id,条件是 A 中的这个对象在 B 中某个对象的半径为 0. 5 的缓冲区中,并且 B 中的这个对象与 C 中的 id 为 10 的对象在空间上连接。

解法 1 采用连接操作的方式实现

```
SELECT a. id
FROM A a,B b,C c
WHERE sdo_relate(a. location,
```

```
sdo_geom. sdo_buffer(b. geom,0. 5,0. 05),
'mask = inside querytype = WINDOW')= 'TRUE' AND
sdo_relate (b. geom,c. geom,'mask = anyinteract
querytyp = WINDOW')='TRUE' AND c. id=10;
```

解法 1 的缺点是，通过连接操作来计算多个对象间的拓扑关系是低效率的。连接操作是将多个对象先进行笛卡尔积运算，再进行条件筛选。然而笛卡尔积运算会产生大量的中间数据，当条件增多时，中间数据会呈指数级增长，执行效率会大大降低。空间查询更看重执行效率，因为它的运算量要远远大于关系查询。为了提高效率，执行系统必须付出额外的代价对查询表达式进行优化，比如尽早执行选择操作、避免出现笛卡尔积操作等，然后才能实际执行[7]。

解法 2 采用函数参数嵌套的方式实现

```
SELECT a. id
FROM A a
WHERE sdo_relate (a. location,
sdo_geom. sdo_buffer( (SELECT b. geom FROM B b
WHERE sdo_relate ( b. geom,(SELECT c. geom FROM C c
WHERE c. id=10),'mask = anyinteract querytyp = WINDOW') =
'TRUE'),0. 5,0. 05),'mask = inside querytype = WINDOW') =
'TRUE';
```

为了便于理解，将解法 2 分解写成如下的形式：

```
SELECT a. id
FROM A a
WHERE sdo_relate (a. location,
                  sdo_geom. sdo_buffer( (代码 A),0. 5,0. 05),
                  'mask = inside querytype = WINDOW') =
                  'TRUE';
```

代码 A：

```
SELECT b. geom
FROM B b
WHERE sdo_relate (b. geom, (代码 B), 'mask = anyinteract query-
typ = WINDOW')='TRUE'
```

代码 B：

```
SELECT c. geom
FROM C c
WHERE c. id=10
```

解法 2 的缺点是，查询表达式结构复杂，用户书写困难且难以阅读，特别是当查询条件达到 3 个以上时，参数的嵌套层数太深，用户无法接受。

4 SQDL-G

4.1 空间运算符与子查询

针对 Oracle Spatial 面对复杂条件查询时的困境，解决的办法就是引入新的查询结构，即子查询结构。我们知道在传统的 SQL 中，子查询能够完成大部分连接操作的功能，同时子查询通常要比连接操作的执行效率高。子查询的结构自然使得选择运算会最先执行，这是查询优化策略中最重要的一条，它常常可以使得执行时间降低几个数量级，因为选择运算会使计算中的中间结果大大减少。要引入子查询结构就必须改变函数型空间算子的表现形式，将其转换为类似于 =，>，<，IN，BETWEEN 这样的运算符。同时 SELECT 表达式的结果不仅可以返回空间对象的标量属性，同时也可以返回空间对象的空间属性，只有如此，子查询的结果才能被父查询利用起来作为父查询的空间查询条件。表 2 显示了连接操作和子查询操作在空间查询中的不同表现形式。

表 2 连接操作与子查询操作的比较

连接操作(简化示意)	子查询(简化示意)
SELECT a. id FROM A a,B b WHERE relate (a. geom,b. geom,'intersect') AND b. id = 10;	SELECT a. id FROM A a WHERE a intersect (SELECT b FROM B b WHERE b. id=10);

通过对两种查询结构的比较还可以发现，在 Oracle Spatial 的空间算子中，查询结果都是从第一个参数所代表的表或集合中选出的，第二个参数只是查询条件的一个载体。而在子查询结构中，被筛选的空间对象的表或集合是 WHERE 子句中空间运算符的左操作数，查询条件的载体自然成为空间运算符的右操作数，查询结果通过 SELECT 子句返回。子查询结构可以通过图 1 形象地表示出来。

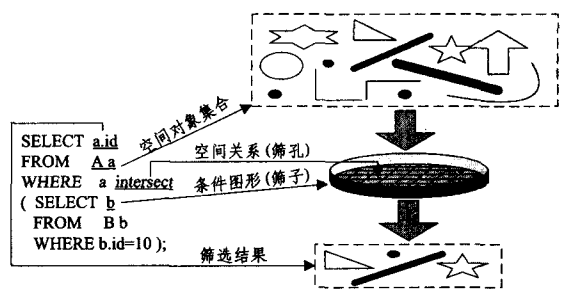

图 1 空间子查询示意图

4.2 SQDL-G 的实现平台

为了满足复杂的空间查询的需要，本文提出一种新的基于 SQL 扩展的空间查询语言 SQDL-G(Semantic Query Description Language for Geography)。首先，SQDL-G 继承了 SQL 的主要结构，易于被用户接受，其次它采用了子查询的表现形式，结构清晰，执行效率高，且更符合自然语言的特征。

SQDL-G 是在 GIS 层上实现的，这样做的好处如下。因为处在数据库的上层，所以 SQDL-G 不会与以 Oracle Spatial 为代表的 SFA SQL 和 SQL/MM 标准发生冲突。Oracle Spatial 的服务对象是数据库层，针对底层数据有一套完整的功能实现，包括数据创建、数据更新、数据查询和数据管理。SQDL-G 则是为 GIS 平台上二次开发的软件服务的，它的主要功能是对空间对象及空间关系进行查询和分析。由于建立在 GIS 平台上，空间数据库的兼容性问题和坐标系选择问题都由 GIS 平台给屏蔽了。以 ArcGIS 为例，其中的 ArcSDE 组件是一个空间数据服务器软件，它为客户端提供了在 DBMS 中存储、管理和使用空间数据的通道。ArcSDE 全面支持 Oracle Spatial，SQL Server，DB2 和 Informix 等空间数据库格式[8,9]。所以建立在 ArcGIS 平台上的 SQDL-G 可以对 ArcGIS 系统所支持的所有格式的空间数据进行操作。图 2 显示了 SQDL-G 和 Oracle Spatial 为不同层次服务。同时，由于 SQDL-G 建立在 GIS 平台上，因此其查询结果可以很方便地以图形化的形式显示出来，给用户一个更加直观的印象。

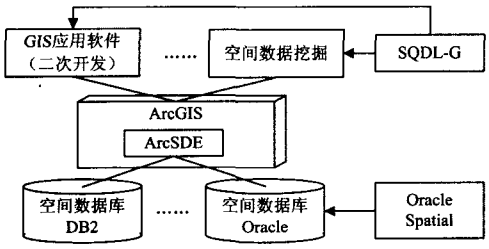

图 2　SQDL-G 所在服务层示意图

设计 SQDL-G 的主要目的是为普通用户提供查询和分析的服务，并不涉及数据创建、数据导入和数据管理，因此它没有复杂的语法和繁冗的参数设置，并且尽量符合自然语言的使用习惯。同时，SQDL-G 可以为基于 GIS 平台的数据挖掘任务提供信息采集和分类的服务，因此它的查询功能也可以是大尺度的、粗犷的。

4.3　SQDL-G 的空间运算符

前面已经提到，为了引入空间子查询，需要将空间算子和部分空间函数转换为空间运算符的形式。因此在设计 SQDL-G 时借鉴了传统 SQL 中的运算符分类，将所有的空间谓词看作空间运算符，并分类为空间算术运算符、空间关系运算符和空间逻辑运算符。空间算术运算符的功能是将一个或两个空间图形进行拓扑运算，运算后的结果依然是一个空间图形。空间关系运算符的功能是判断两个空间图形间的拓扑关系是否成立，运算结果是真/假逻辑值。空间逻辑运算符的功能等同于传统的逻辑运算符的功能，实现逻辑值间的运算。空间运算符的符号和含义如表 3 所列。

表 3　空间运算符分类表

空间算术运算符		空间关系运算符		空间逻辑运算符	
∪	Union	*	AnyInteract	∧	And
∩	Intersect	⊙	Inside	∨	Or
−	Difference	∅	Cross	¬	Not
⊕	Xor	≡	Equal		
◎	Buffer	><	Touch		
○	Boundary	●	Contain		

空间算术运算符的左、右操作数分别是 SQDL-G 表达式（子查询），其计算结果作为空间关系运算符的右操作数。同时，空间关系运算符的左操作数就是父查询中的被筛选对象集合。为了让子查询的运算结果成为父查询的空间筛选条件，子查询的 SELECT 子句不仅可以返回对象的标量属性，还可以返回对象的空间图形。当然，SQDL-G 表达式也能对标量条件（非空间条件）进行筛选。标量条件将与空间条件并列出现在 WHERE 子句当中。

除了上述运算符外，SQDL-G 还定义了一些空间函数和集合函数来丰富语义，如 area()，length()，sum()，avg()等。

4.4　SQDL-G 的实例及其形式化

为了完整展现 SQDL-G 的特性，下面将演示一个多条件的空间查询表达式。

例 2　在某个二次开发的 GIS 系统中，存储了中国的主要地理数据，分别由 5 个图层组成，它们依次是省会（capital）、铁路网络（railroad）、公路网络（highway）、水域（water）、省（province）。现在需要查询位于江苏省和安徽省境内的湖泊，并显示其名字。

```
SELECT w. name
FROM water w
WHERE w. name like '%湖', w anyInteract (SELECT p
                                         FROM province p
                                         WHERE p. name =
                                         '江苏省' )
                        or w anyInteract (SELECT p
                                          FROM province p
                                          WHERE p. name=
                                          '安徽省' )
```

SQDL-G 可以被类似关系代数[10,11]一样的抽象语言形式化，我们称这种抽象语言为空间代数。在空间代数中，符号 Π 是投影运算符，表示在查询结果中显示空间对象的某个标量属性，符号 σ 是选择运算符，表示从众多空间对象中筛选符合条件的对象，其它空间代数符号如表 3 所列。把例 2 按照空间代数的规则做形式化处理，结果为：

$$\prod_{w.\,name}\left(\sigma_{(w.\,name\ like\ '\%湖')\wedge((w*\sigma_{p.\,name='江苏省'}(province\ p))\vee(w*\sigma_{p.\,name='安徽省'}(province\ p)))}(water\ w)\right)$$

空间代数能将 SQDL-G 表达式抽象为代数的形式，这为 SQDL-G 的计算、简化和优化提供了数学基础，这也是本课题下一步要研究的内容。

4.5　SQDL-G 表达式的等价转换

具有丰富运算符的 SQDL-G 表达式，在一定条件下可以等价转换为其它的形式，不同表达形式具有不同的执行策略和执行效率。以例 2 为例，其中的子查询结构是两个 SELECT 子句的空间逻辑表达式，并且这两个 SELECT 子句作用于同一个空间关系运算 anyInteract，可以将 anyInteract 以公因式提取出来。同时空间逻辑运算符 or 等价转换为空间算术运算符 union。经过这样的处理，子查询中的空间逻辑表达式就转换成为空间关系表达式，详见例 3。

例 3

```
SELECT w. name
FROM water w
WHERE w. name like '%湖', w anyInteract ( ( SELECT p
                                            FROM province p
                                            WHERE p. name =
                                            '江苏省' )
                                          union
                                          ( SELECT p
                                            FROM province p
                                            WHERE p. name =
                                            '安徽省' ) )
```

例 3 还可以作进一步简化。观察子查询结构中的两个 SELECT 子句，发现它们的查询对象都是 province 图层，所以可将 province 作为公因式提取出来，并将两个标量条件合并到一个 SELECT 子句中。其中空间关系运算符 union 转换为关系运算符 or，详见例 4。

例 4

```
SELECT w. name
FROM water w
WHERE w. name like '%湖', w anyInteract ( SELECT p
                                          FROM province p
                                          WHERE p. name =
                                          '江苏省' or p. name =
```

'安徽省')

注意,空间代数在作等价转换的时候要格外小心,因为它的等价规则容易被人误用。虽然例1向例2转换可以看作式(1):

$$\underset{w*x\vee w*y}{\sigma}(w)\Leftrightarrow\underset{w*(x\cup y)}{\sigma}(w) \tag{1}$$

式(2)表示了不同的情况:

$$\underset{w*x\wedge w*y}{\sigma}(w)\neq\underset{w*(x\cap y)}{\sigma}(w) \tag{2}$$

式(2)左半部的含义是选择空间上既与 x 有关联,又与 y 有关联的 w。式(2)右半部的含义却是选择空间上与 x 和 y 相交部分有关联的 w。显然,式(2)右边只是左边的一个子集。但是,观察式(3)却是另外一种情形:

$$\underset{w\odot x\wedge w\odot y}{\sigma}(w)\Leftrightarrow\underset{w\odot(x\cap y)}{\sigma}(w) \tag{3}$$

式(3)左半部的含义是选择空间上既在 x 内部,又在 y 内部的 w。式(2)右半部的含义是选择空间上在 x 和 y 相交部分的内部的 w。通过分析可知,式(3)的左边和右边具有相同的含义。

通过上面的式(1)—式(3)可知,在对 SQDL-G 表达式做转换和简化的工作时,还需要考虑空间算术运算符、空间关系运算符和空间逻辑运算符,这样才能避免出现错误。

4.6 SQDL-G 的解析与执行

SQDL-G 的语法主体框架如下:

〈SQDL-G 表达式〉::= SELECT 查询对象
FROM 图层
WHERE 〈标量条件〉,〈空间条件〉

〈空间条件〉::= 〈空间条件表达式〉[空间逻辑运算符 〈空间条件表达式〉]

〈空间条件表达式〉::= 查询对象 空间关系运算符 〈SQDL-G 表达式〉

在解析 SQDL-G 表达式之前首先构建抽象语法树 AST (Abstract Syntax Tree, AST),然后从下至上、从右至左的逆后序(RLN)依次扫描语法树,并形成执行序列。下面以例5为例。

例5 查找所有在武昌区范围内的学校。

首先,将例5中的 SQDL-G 表达式转换为图3中的语法树,然后按照逆后序的次序扫描语法树并得到执行序列。例3的执行序列是,首先执行子查询中 SELECT 子句,找到武昌区对应的图形,然后将此图形作为父查询的空间条件,执行父查询的 SELECT 子句,即查找在该图形内的学校。

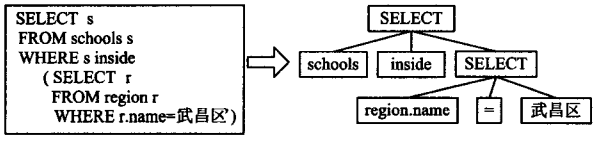

图3 SQDL-G 转换为语法树

本论文采用 ArcGIS9.2 作为 GIS 平台,VB 作为 GIS 二次开发的语言,以 ArcEngine 工具包来实现显示地图和执行查询的工作。由于 SQDL-G 表达式的解析工作是由 JavaCC 工具包实现的,而 VB 不能直接调用 Java 程序,因此本文的解决方案是,先由 JavaCC 将 SQDL-G 表达式解析为可识别的中间文件,再由 VB 编写的执行引擎读取中间文件,并产生执行序列,最后通过文本和绘图两种方式同时显示出来。图4显示了整个解析与执行过程的框架,图5是执行引擎在运行例2时的界面截图。

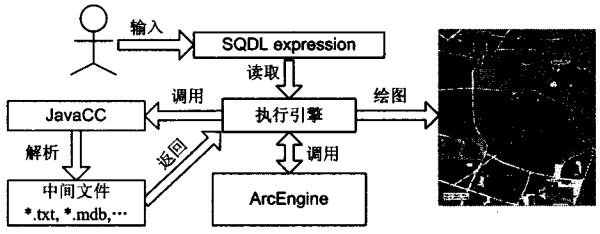

图4 SQDL-G 的解析与执行框架图

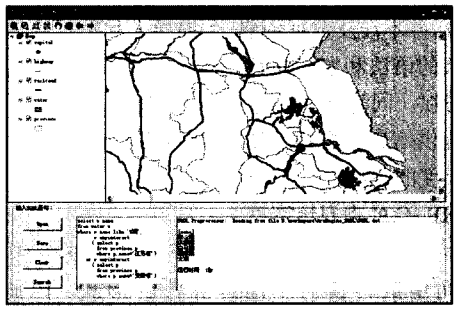

图5 执行例2的界面截图

图5中的地图数据来源于测绘科学数据共享服务网[12]。

结束语 随着 GIS 系统的发展并逐渐融入日常生活中,普通用户越来越不满足于现阶段以关键字的方式来实现地理信息的查询和分析。人们不仅希望查询语言能够像自然语言一样方便使用,还希望查询语言能够像自然语言一样具有灵活的表达能力。对于 SQDL-G 的研究恰好满足了当前用户的需求。当然 SQDL-G 还有许多有待完善的地方,它还需要加强针对空间对象的聚集分析的功能,即将 Order by 子句和更多的空间聚集函数引入到空间查询中。到目前为止,SQDL-G 中的查询条件都来自于地图中的空间对象,下一步应该允许用户使用鼠标或其它工具自定义筛选条件,比如查找在用户鼠标拖出的矩形框中的城市。同时,针对栅格数据和三维数据的分析和处理必将成为今后空间数据研究的热点,所以 SQDL-G 的进一步研究也不可避免地要涉及到它们。

参 考 文 献

[1] 吴信才. 空间数据库[M]. 北京:科学出版社,2009:159

[2] 张纯,陈国荣,程昌秀. 两种主流空间数据库国际标准与应用分析[J]. 地球信息科学学报,2009,11(4):526-533

[3] 龚健雅,高文秀. 地理信息共享与互操作技术及标准 [J]. 地理信息世界,2000,3(3):18-17

[4] Open Geospatial Consortium Inc. OpenGIS Implementation Specification for Geographic information-Simple Feature access-Part 1: Common Architecture, 2006 [EB/OL]. http://www.opengeospatial.org/standards/sfa,2009-12-10

[5] International Organization for Standardization. ISO/IEC 13249-1:Information Technology-Database Language-SQL Multimedia and Application Packages-Part 3: Spatial, 2006 [EB/OL]. http://www.iso.org/iso/catalogue_detail.htm? csnumber = 53698,2009-12-10

[6] Oracle Inc. Oracle Spatial User's Guide and Reference Release 9.0.1[EB/OL]. http://download.oracle.com/docs/html/A88805_01/toc.htm,2009-12-10

(下转第213页)

表 1　问句在主题域中的分布情况

No	Domain	51～100	101～150
1	International Economic	11	5
2	International Finance	1	0
3	U. S. Ecomonic	2	4
4	U. S. Politics	0	3
5	Science and technology	11	5
6	Environment	4	0
7	International Relations	5	5
8	Law and government	4	6
9	Finance	2	2
10	international politics	2	9
11	Military	4	3
12	Political	2	0
13	Medical and biological	0	8

表 2 是多主题域用户模型扩展问句方法的实验结果。$\alpha=0$ 是基线，没有扩展问句；$\alpha=1$ 是只包含反馈集的扩展模型，而不包含原问句模型。问句 101～150 在 disk-1 中共有 7002 个相关文档。α 在 0.1 至 0.9 之间的结果均好于一般的语言模型检索结果，α 在 0.8 处查全率最高，在 0.6 处平均查准率最高，在 0.4 处，前 20 个文档的查准率最高。

表 2　多主题域用户模型扩展问句实验结果

α	Recall/7002	MAP	p@20
0	4036	0.2133	0.3930
0.1	4083	0.2167	0.3970
0.2	4126	0.2202	0.3970
0.3	4153	0.2229	0.4050
0.4	4175	0.2245	**0.4170**
0.5	4198	0.2253	0.4050
0.6	4207	**0.2261**	0.4030
0.7	4223	0.2246	0.4070
0.8	**4245**	0.2210	0.4020
0.9	4225	0.2127	0.3850
1	3855	0.1771	0.3330

表 3 是 $\alpha=0.6$ 时 UF 与 LM 的比较，UF 的各项指标均比 LM 有较好的改进。

表 3　多主题域用户模型扩展问句模型与一般语言模型的比较

Model	Rec/7002	MAP	p@20
LM	4036	0.2133	0.3930
UF	4207	0.2261	0.4030
+%	4.2	6.0	2.5

表 4 是 UF 与 PF 相结合的方法（记为 UF+PF）的实验结果，此处 λ，β 和 γ 分别取 0.2，0.3 和 0.5。与 LM 相比，各项指标均有较大的改进，与 PF 相比略有提高。

表 4　UF 与 UF+PF 的实验结果与比较

Model	Rec/7002	MAP	p@20
LM	4036	0.2133	0.3930
PF	4320	0.2388	0.4290
UF+PF	4364	0.2410	0.4390
+%to LM	8.1	13.0	11.7
+%to PF	1.0	0.9	2.3

结束语　个性化信息检索是复杂多样的，很难用一种通用的方法满足各种要求。本文提出的方法吸取了伪相关反馈方法和相关反馈方法的优点，考虑了用户希望检索结果涵盖其多个感兴趣主题域的情况。通过实验证明了本方法的有效性。实验表明 UF 的性能优于 LM，PF 的性能优于 UF，UF+PF 的性能优于 PF。用户模型具有很重要的作用，它的质量会直接影响到检索性能。许多细微的东西在我们的方法中还没有考虑，从选取好的用户模型内容、反馈集和扩展词等几方面入手，提高个性化检索性能还有一定的空间。

参 考 文 献

[1] Ponte J, Croft W B. A language modeling approach to information retrieval[C]// Proceedings of ACM SIGIR. 1998: 275-281

[2] Rocchio J. Relevance feedback in information retrieval[Z]. the SMART retrieval system. 1971: 313-323

[3] Salton G, Buckley C. Improving retrieval performance by relevance feedback[J]. American Society for Information Science, 1990, 41: 288-297

[4] Zhai C X, Lafferty J. Model-based Feedback in the Language Modeling Approach to Information Retrieval[C]// Proceedings of ACM CIKM. 2001: 403-410

[5] Cao Guihong, Nie Jianyun. Integrating Word Relationships into Language Models[C]// Proceedings of ACM SIGIR. 2005: 298-305

[6] Nie Jian-yun, Cao Gui-hong, Bai Jing. Inferential Language Models for Information Retrieval[J]. Transactions on Asian Language Information Processing (TALIP), 2006, 5(4): 296-322

[7] Bai Jing, Song Dawei, Bruza Peter, et al. Query Expansion Using Term Relationships in Language Models fo Information Rereieval[C]// Proceedings of ACM CIKM. 2005: 688-695

[8] Bai Jing, Nie Jian-yun, Bouchard H, et al. Using Query Context in Information Retrieval [C] // Proceedings of ACM SIGIR. 2007: 15-22

[9] Bai Jing, Nie Jian-yun. Adapting information retrieval to query contexts[J]. Information Processing and Management, 2008, 44: 1901-1922

[10] Lafferty J, Zhai Cheng-xiang. Document Language Models, Query Models, and Risk Minimization for information Retrieval[C]// Proceedings of ACM SIGIR. 2001: 111-119

[11] Shen Xue-hua, Zhai Cheng-xiang. Active Feedback in Ad Hoc Information Retrieval[C]// Proceedings of ACM SIGIR. 2005: 59-66

[12] Buckley C, Salton G, Allan J, et al. Automatic query expansion using SMART[C]// Proceedings of ACM SIGIR. 1988: 321-331

[13] Dumais S, Cutrell E, Caliz J, et al. Stuff I've seen: A system for personal information retrieval and re-use[C]// Proceedings of ACM SIGIR. 2003: 72-79

[14] Liu F, Yu C, Meng W. Personalized web search by mapping user queries to categories[C]// Proceedings of ACM CIKM. 2002: 558-565

（上接第 210 页）

[7] Jarke M, Koch J. Query Optimization in Database Systems [J]. ACM Computing Surveys, 1984, 16: 2

[8] ESRI Inc. ArcGIS Server—企业级服务器 [EB/OL]. http://www.esrichina-bj.cn/templates/T_yestem_News/index.aspx?nodeid=155, 2009-12-10

[9] 韩鹏，王泉. 地理信息系统开发[M]. 武汉：武汉大学出版社，2008: 340

[10] Hall P A V, Hitchcock P, Todd S J P. An Algebra of Relations for Machine Computation [C]// Conference Record of the Second ACM Symposium on Principles of Programming Languages. 1975

[11] 王珊，萨师煊. 数据库系统概论（第四版）[M]. 北京：高等教育出版社，2006: 52

[12] 测绘科学数据共享服务网. 1 比 400 万中国地图 [EB/OL]. http://sms.webmap.cn/, 2009-12-10