基于对象关系数据库的时空数据存储与访问技术*

袁一泓，高勇

（北京大学遥感与地理信息系统研究所，北京，100871，yyhmary@163.com）

摘　要　时空数据库的研究是近年来地理信息科学与数据库技术领域研究和应用的热点，其中时空数据模型及其存储和访问技术是时空数据库的关键所在。本文在基于时间片的连续快照模型基础上，采用对象关系数据库 PostgreSQL 作为实现平台，定义了时空数据抽象数据类型及其相关操作，将其嵌入到对象关系数据库中，扩展其时空数据的存储和查询能力，并利用时空索引技术进行查询优化。应用实验验证了上述时空数据存储与访问方法的有效性和实用性。

关键词　时空数据模型，对象关系数据库，时空数据存储与访问

Researches on Storage and Access Technologies of Spatio-Temporal Data based on ORDBMS

YUAN Yihong, GAO Yong

(Institute of RS and GIS, Peking University, Beijing, China, 100871, Email:yyhmary@163.com)

Abstract: Researches on Spatio-Temporal database gain more and more attentions in domain of geographical information science and database technologies, where Storage and Access Technologies of Spatio-Temporal models play a significant part. Based on time-sliced date model, this paper defines relative data objects and operations, which are implanted into ORDBMS to improve the Storage and Access capacity. Spatio-Temporal indexing techniques are also used to optimize the query processing. All these techniques cited above are implemented in the world's most popular open-source ORDBMS-PostgreSQL. The experimental results come out to indicate that technologies in this paper are efficient and expedient in Spatio-Temporal data research fields.

Key words: Spatial-Temporal data model, ORDBMS, Storage and Access of Spatial-Temporal data

1 引言

现实世界具有明显的动态特征，空间和时间是现实世界最基本、最重要的属性[1]。通常情况下，大多数实体不会长时期具有恒定不变的空间特征，其空间位置与属性具有随时间变化的特征。因此，地理时空数据的组织与管理逐渐成为近年来地理信息领域的研究热点之一[2]。建立合理有效的时空数据模型，通过 GIS 的现有技术描述、存储和分析空间对象及其属性的动态特性，可以记录历史的地理实际状况，有助于解释各种地理变化的原因及过程，进而服务于空间决策层。

时空数据模型是时空数据库的核心部分，它保证着时空数据库的完整性，也决定着对实

*空间信息集成与 3S 工程应用北京市重点实验室（北京大学）科研基金资助

体动态信息的具体操作的实现。一个时态GIS或时空数据库系统如果没有良好的时空数据模型，就很难有效的支持对事物随时间变化的信息的查询和分析[3]。

当前基于传统关系型数据模型的时空数据建模中，根据对时间属性不同的处理方式，主要存在两中方法：1）时间作为属性附加项：如第一范式模型、时空复合模型等；2）时间作为新的维度：如时空立方体模型、基态修正模型等。此外，基于对象关系数据库系统（ORDBMS）存储管理空间数据日益成为研究的主要趋势，由于 ORDBMS 具有强大的拓展功能，支持复杂类型的拓展定义，并可在此基础上实现各种复杂算子的拓展，易于对时空数据进行类型定义和实现与之相应的各种操作 [4]。因此，基于 ORDBMS，以定义抽象数据类型（ADT）的方式完成对时空数据的建模是目前研究的热点方向。

但是，目前该领域的研究中可利用的技术实现仍然较少。因此，本文选择开源的对象关系数据库 PostgreSQL，利用其类型和操作的可拓展性，基于连续快照模型定义其时空数据抽象数据类型，扩展其时空数据存储与访问功能，支持时空关系查询和分析应用。

2 时空数据模型定义

2.1 时空数据模型

本文定义的时空数据模型结构如图1所示。时空数据对象模型在OpenGIS定义的几何类型基础上扩充了时态类型，形成了统一的时空数据模型。时空数据类型DyGeometry由空间数据类型Geometry和时态数据类Temporal组合定义的[4]，其中空间类型依据OpenGIS规范的Geometry类型定义，具体分为Point、Line、Polygon等子类型[5]。

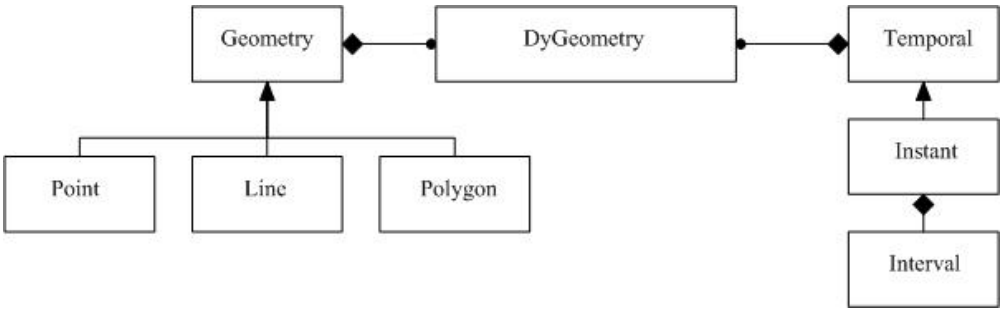

图 1 基于对象关系数据库的时空数据对象模型

2.2 时空数据类型定义

（1）空间类型

在本文定义的时空类型中，空间类型部分直接延用了 OGC 的定义标准。基本的空间类型包括点（Point）、线（Line）、多边形（Polygon），OGC 规范给出了其基于集合论和拓扑论的定义。

（2）时间类型

为了定义时空数据类型，还需要先对时间类型加以定义。

将时间全集 T 表示为一维实数欧氏空间 IR^1（一维实数轴），设$(T,<)$为一个全序集合，则在 T 上的一个时刻 Instant 定义为

$$Instantl(T)=\{t|t\in R\cup\{\phi\}\}$$

T 上的时段 Interval 定义为：

$$Interval(T)=\{(s,e,lc,rc)|s,e\in T, lc,rc\in bool, s\leq e, (s=e)\Rightarrow(lc=rc=true)\}$$

即一个时段由它的起止点 *s* 和 *e* 以及标示该时段是否为闭区间两个标志表达。一般情况下，时段表示为闭区间，即 *lc=rc=true*，则 *Interval*=[*t,t'*]。

（3）时空类型

基于连续快照模型，时空类型 *DyGeometry* 是从时间集合 *T* 到空间对象 *O* 的映射，即：

$$DyGeometry: T \rightarrow O$$

那么一个时空对象就可以表示为由时段和空间几何对象的值对构成的集合，即：

设 *O* 为时空类型的空间几何对象可能的取值集合，则时空类型 *DyGeometry* 定义为：

$$DyGeometry=\{(i,o)|i \in T, o \in O\}。$$

而时空类型在给定时刻 *t*（$t \in T$）的时间片是一个时刻和空间几何对象的二元值对(i_t,o_t)，其中 $i_s \le t \le i_e$，而 o_t 就是此对象在 *t* 该时刻的快照。整个二元值对可作为一个整体的快照类型定义或存储。

上述对于 DyGeometry 的离散形式定义对与离散时空变化和连续时空变化都同样适用。对于连续变化，可以定义 *i* 为两个相邻观测时刻所构成的时段。设一系列观测时刻的全集表示为 $MT=\{t_1, t_2,\ldots,t_n\}$，满足 $t_p<t_q$ 且 $0 \le p<q \le n$，则令 $i \in \{[t_p, t_{p+1}]\}$，满足 $t_p \in MT$ 且 $0 \le p<n$。在这种情况下，要求对该移动对象的观测时间间隔满足取样要求，即假定规定的取样时刻的观测能够反映移动对象的时空变化规律。

（4）时空类型操作

为实现对上述定义的时空类型的操作，定义时空操作函数。其中基本时空操作函数如表1所示，包括基本编辑操作、时空属性操作（计算对象大小、获取快照值）等。

表1 时空操作函数

clone(dygeom : DyGeometry) : DyGeometry	对时空对象进行复制
size (dygeom : DyGeometry): DyGeometry	计算时空对象的大小
lifecycle(dygeom : DyGeometry): Interval	时空对象的生命周期
snapshot(dygeom : DyGeometry, inst : Instant): Geometry	取得时空对象在特点时刻的快照
exists(dygeom : DyGeometry, inst : Instant): bool	时空对象在特定时刻是否存在

另一类重要的时空操作是时态关系判别函数。时态关系描述了两时空对象在时间维度上的相容性及先后关系，是进行时空拓扑关系查询的基础。时空对象A和B的时态关系可以表示为由其开始时刻和终止时刻确定的一个四元组所决定，设t_{sA}、t_{sB}分别为时态对象A、B的开始时刻，t_{eA}、t_{eB}分别为时态对象A、B的终止时刻，则其时态关系函数的定义如表2所示。

表2　时态关系函数

Before(A : DyGeometry, B : DyGeometry) : bool	判断时空对象A的结束时刻是否在B的开始时刻之前（$t_{eA}<t_{sB}$）
After(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}>t_{eB}$
During(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}>t_{sB} \wedge t_{eA}>t_{eB}$
Contains(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}<t_{sB} \wedge t_{eA}>t_{eB}$
Overlaps(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}<t_{sB} \wedge t_{eA}<t_{eB}$
Overlapped-by(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}>t_{sB} \wedge t_{eA}>t_{eB}$
Meets(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{eA}=t_{sB}$

Met-by(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}=t_{eB}$
Equals(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}=t_{sB}\wedge t_{eA}=t_{eB}$
Starts(A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}=t_{sB}\wedge t_{eA}<t_{eB}$
Started-by (A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}=t_{sB}\wedge t_{eA}>t_{eB}$
Finishes (A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}>t_{sB}\wedge t_{eA}=t_{eB}$
Finished-by (A : DyGeometry, B : DyGeometry) : bool	判断是否$t_{sA}<t_{sB}\wedge t_{eA}=t_{eB}$
Intersection (A : DyGeometry, B : DyGeometry) : Interval	两个时态对象的生命周期的重合时段

类似的，可以定义时态对象与时刻、时段之间时态关系判别函数。

3 时空数据类型的数据库实现

在上述时空数据模型的基础上，本文采用开源的 PostgreSQL 对象关系数据库对上述时空类型的存储和访问过程进行了实现。通常在 ORDBMS 中扩展模型时应满足：（1）定义数据类型；（2）定义与数据类型相关的对象操作函数；（3）对于新拓展类型及相关操作都可以通过 SQL 进行访问。

在本文的应用中，依据 PostgreSQL 的拓展规则，具体完成以下内容：

（1）在 C 语言的类型系统上定义新的时空类型 DyGeometry；

（2）实现类型的输入输出函数和文本转换函数，以完成 DyGeometry 类型和文本类型的相互转化，便于将信息以文本方式展现给用户；

（3）实现时空对象的操作函数，以进行时空查询和分析；

（4）将上述内容编译为 PostgreSQL 自身的动态链接库，在数据库中进行注册。

3.1 时空数据类型创建

本文基于WKBGeometry定义DyGeometry在PostgreSQL中的数据结构。WKBGeometry是OpenGIS协会指定的图形数据存储方法。它把几何实体的图形信息编码成连续的二进制数据流，能以二进制、Raw或图像类型字段存储在SQL数据库中，并能通过SQL语句或ODBC客户程序读取[6]，极大地方便了客户端与数据库端的交互过程。

基于这一定义模式，本文首先对OGC中定义的空间类型Geometry类的二进制形式WKBGeometry及相关操作在PostgreSQL中进行了拓展，并实现了相关的操作函数及空间拓扑分析等空间数据库的必需功能。在实现空间类型拓展的基础上，进行时空数据及时空操作的扩展。

时空对象DyGeometry的WKB表达形式如下：

```
typedef struct tagWKBSlice
{
    Time        t;
    Geometry    geom;
} WKBSlice;
typedef struct tagWKBDyGeometry
{
    uint32      numSlices;
    WKBSlice    slices[numSlices];
```

}WKBDyGeometry;

其中，WKBDyGeometry表示为时间片WKBSlice的数组，而每个时间片是一个由时刻t和空间几何对象geom构成的二元组。WKBDyGeometry的内存模式如图2所示。

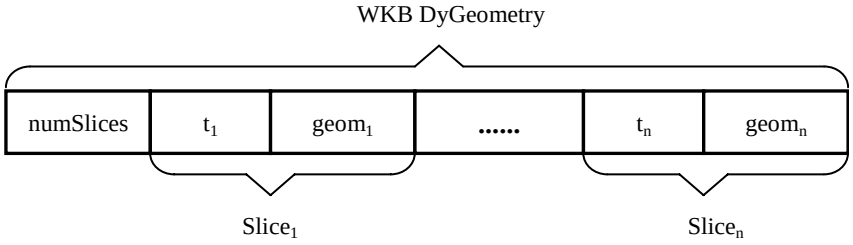

图2　WKBDyGeometry的存储模式

作为二进制表示法，WKBDyGeometry可以很好地满足数据库接口的存储和访问需求，但二进制流数据不能直接以可读形式呈现给用户，也不能直接在SQL语言中使用，因此需要对时空对象进行文本方式的定义。

OGC规范中定义了空间对象的文本形式WKT，为空间信息的文本表达提供了标准化的模式。本文在此基础上定义时空数据的文本类型，即WKTDyGeometry。

WKTDyGeometry的EBNF（Extended Backus Naur Form）表达形式如下所示：

<WKT_DyGeometry>::= DyGeometry(<时间片>{;<时间片>}*)

<时间片>::=<时间对象><空间对象>

<时间对象>::=Time(<年>-<月>-<日> <时>:<分>:<秒>)

<空间对象>::=<Point>|<Line>|<Polygon>|NULL

<Polygon>::=Polygon({<Linestring>})

<Line>::= Linestring(<Linestring>)

<Linestring>::= <坐标点对>{,<坐标点对> }*

<Point>::= Point(<坐标点对>)

<坐标点对>::= <x> <y>

<x> := double precision literal

<y> := double precision literal

其中，空间对象为NULL时标识该对象消亡，时间对象的具体格式为：Time(YYYY-MM-DD HH-mm-SS.SSS)。下为一个包含两个时间片的WKTDyGeometry的表示实例：

DyGeometry(Time(2005-04-20 10:10:10.000) Point(10 10); Time(2005-04-20 15:10:10.000) Point(20 10))

此外，根据PostgreSQL的自定义类型拓展规则，定义相应的I/O 函数（输入输出函数）。这些函数决定该类型如何在字串里出现（让用户输入和输出给用户）以及类型如何在存储器里组织[11]。 并使用sql语句在数据库中对类型进行注册，PostgreSQL的类型注册是通过CEATE TYPE函数实现的，创建时需要对类型的输入输出函数及类型定义的源文件进行指定。

3.2操作函数创建

利用PostgreSQL的扩展功能，将时空操作函数注册到PostgreSQL中，使这些函数可以在SQL语言中直接使用，扩展其时空操作能力。例如snapshot函数的注册方法如下：

CREATE OR REPLACE FUNCTION snapshot (DyGeometry, Instant)

RETURNS Geometry

AS ‘libDyGeometry’, ‘get_snapshot’

完成类型与函数的注册后，便可进行时空数据库创建、时空关系查询等针对时空数据的具体数据库应用：

建立数据表：

建立表buses，记录若干长途汽车的运行情况，其中字段pos为时空数据类型，即：

```
CREATE TABLE buses
{
    id          integer
    pos         DyGeometry
}
```

对数据表的基本操作包括插入、删除、修改、查询等，即：

插入：

```
INSERT INTO buses (id, pos)
VALUES ( ‘001’, ‘DyGeometry(Time(2005-04-20 10:10:10.000) Point(10 10))’)
```

修改：

```
UPDATE buses
SET pos = ‘DyGeometry(Time(2000-01-01 00:00:00.000) Point(0 0))’
WHERE id = ‘001’
```

时态关系查询：

如查询在某一具体时刻之后出发的长途汽车：

```
SELECT id FROM buses
WHERE After(pos, Time(2000-01-01 00:00:00.000))
```

时空拓扑关系查询：

设已建立表 region，有字段：id: integer、county:Geometry，记录了行政区划数据。查询在给定时刻在特定县域内运行的长途汽车：

```
SELECT buses.id FROM buses, region
WHERE Contains(county, snapshot(buses.pos, Time(2000-01-01 00:00:00.000)))
        AND region.id=002
```

3.3 时空索引

为提高查询效率，需要为时空数据建立索引机制。目前大多数时空数据索引方法都是基于 R 树及其变种（如 R*树），扩展了 R 树的结构及其维护机制支持时空数据查询。而 PostgreSQL 系统支持通用搜索树（GiST：Generalized Search Tree）技术，可以为用户扩展的时空对象数据类型提供新的索引方法。GiST 树使用方便，只需指定索引项内容，并且扩展包括索引项提取函数、索引项合并和分离（对应于树内部结点合并和分裂）、查询在内部结点上的分支处理以及查询在叶子节点上的处理等相关操作即可[7]。本文在 GiST 框架下，按照 R* 树索引机制完成上述函数，并将其注册到 PostgreSQL 中，完成对时空数据索引方法的扩展。初步实验结果表明，该时空索引方法可以明显提高查询效率。

4 应用

利用上述时空数据库的实现手段，我们对于北京市铁路物流运输数据进行了管理和分析。铁路物流数据作为基于 Point 类型的时空对象。应用这一技术，可以对物流数据进行合理化存储，并进行相应的时空拓扑查询和分析，如进行货物的路径追踪（图 3）、路线的流量分析（图 4）等。该实例证明了本文时空数据库实现技术的实用性及有效性。

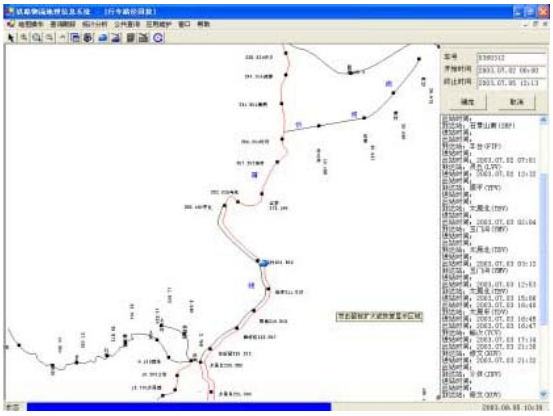

图 3 铁路货物的运输路径

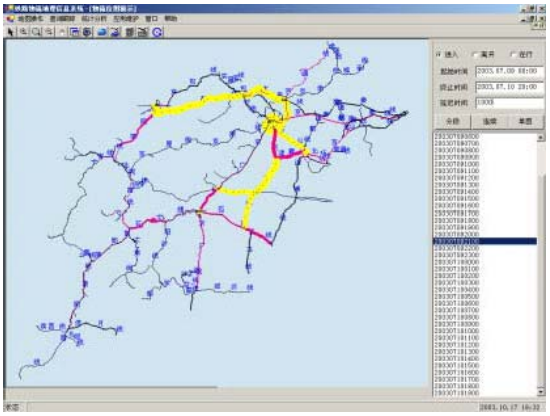

图 4 货物流量分析

5 结论与展望

基于ORDBMS来管理时空数据是目前研究的主要趋势，本文利用开源数据库PostgreSQL，对基于时间片的时空数据模型的存储和访问技术进行了实现，在数据库中拓展了基本的数据类型和相关时空操作，实验证明效果良好。

本文虽然采用了索引技术提升查询效率，但由于模型本身的限制，没有对效率问题进行较深入的研究，后续工作可对时空数据模型进行改进，以探索更为高效的方法。

参考文献：

[1]姜晓轶 周云轩. 从空间到时间--时空数据模型研究. 吉林大学学报（地球科学版），2006, 36(03):480-485

[2]王家耀, 魏海平, 成毅等. 时空 GIS 的研究与进展. 海洋测绘, 2004, 24(05): 2-4

[3] YUAN M. Temporal GIS and Spatio-Temporal Modeling[D]. The University of Oklahoma, Norman, Okla. 1996

[4] 高勇,林星,刘瑜,邬伦,陈斌,马修军. 基于对象关系数据库的时空数据模型研究[J]地理与地理信息科学 , 2006, 22(03): 26-30

[5]Open GIS Consortium Inc. OpenGIS simple features specification for SQL1.1[R]. http://www.opengis.rog/docs/99-049.pdf, 1999

[6] 孙红春 王卫安 基础地理信息图文一体化数据模型 测绘学报, 2001,(01): 4-6

[7] 林星 高勇 张毅 余博 秦适 邬伦 对象关系数据库中的时空索引机制研究 地理与地理信息科学 2006, 22(03): 31-34