

移动对象数据模型^{*})

Data Modeling of Moving Objects

吴 劲 卢显良

(电子科技大学计算机学院 成都610054)

Abstract Spatio-temporal databases deal with geometries changing over time. It is difficult to provide a universal spatio-temporal data model. But, aiming at special application, such as moving objects, it is possible to form a special spatio-temporal data model. We propose to represent such time-dependent geometries as attribute data types with suitable operations, that is, to provide an abstract data type extension to a DBMS data model and query language. This paper presents a design of such a system of abstract data types.

Keywords Moving objects, Spatio-temporal database, Modeling, Data types

1 引言

空间数据库把“空间”的概念引入了数据库系统中,使之能够处理2D和3D方面的问题,最常用的就是GIS应用,这一领域的研究已经进行了二十几年,并形成体系,拥有许多的研究团体和一定数量的与空间数据库系统相关的期刊、杂志。而时间数据库则是利用了时间的概念来创建一种各高抽象级的数据库应用,近二十年来,该领域的研究也非常活跃,特别是近几年,时间数据库的重要性,已得到了科技界的认同。把这两个独立的研究领域结合起来,就形成了一门新兴学科“时空数据库”的基础。

时空数据库不仅要处理复杂的空间问题,还要考虑随着时间的变化而发生的空间变迁,这一特性决定了很难形成一个通用的高性能的时空数据模型。但可以针对特定应用,根据这类应用的特点,形成专用的时空数据模型,并根据其特性扩充查询语言。时空数据库中非常有应用前景的一类应用系统就是移动对象数据库,该系统可广泛应用于交通、旅游及军事等系统中。

本文从介绍移动对象数据库系统的特点入手,用一种带有相应操作的特定数据类型来表示基于时间的空间几何问题,在此基础上,就可以提供一种抽象的数据类型去扩展现有数据库管理系统(DBMS)。

2 移动对象数据库系统的特点

我们可以把时空运动对象分成两大类:离散移动对象和连续移动对象。对于离散移动对象,例如森林,要在数据库中,跟踪其物理位置和范围的变化,相对比较容易,我们可以通过更新数据库中的数据来达到目的。若希望查询空间位置和范围的离散变化,还可以在数据库中加入时间列,使时间列对应一段时间间隔,特定的空间位置和范围就与特定时间段相关联,这样,就可以反映出空间信息随着时间的变化而发生的变迁。

但对于连续移动对象(简称移动对象),相对于离散移动对象而言,要在传统的DBMS中表示它们就非常困难。因为在传统的DBMS中,数据库中存储的数据是常量,除非它被

明确地更新。而移动对象在空间的位置随时间的变化而变换,如果用传统的DBMS来管理移动对象数据类型,必然需要非常频繁地更新数据库,这不仅严重影响数据库系统的性能,还会加重无线带宽的开销。

因此,采用传统方式管理移动对象数据是行不通的,但若是换一种思路,数据库中存储的数据可以不是常量,是变量,甚至可以是函数,情况就不同了。例如,如果知道移动对象的速度和路由,就可以计算出未来某时间移动对象的位置。具体来讲,就是在数据库中存贮的移动对象位置不是常量,它的值取决于移动对象的运动速度和路由,而速度和路由的变化率远远低于位置的变化率,这样在大多数情况之下,就不需频繁更新数据库,可以大大改善系统性能。

为了支持这一特性,必须对传统的DBMS进行扩展,首先应定义适合表示移动对象数据的数据类型和操作,在此基础上可以开发出针对移动对象的数据切片(data blade),把它嵌入到现有的DBMS中就可以支持移动对象应用了。

3 移动对象数据模型

我们可以把移动对象数据模型分成五大类:基本型(BASE)、空间型(SPATIAL)、基本时间型(TIME)、时间型(TEMPORAL)和域型(RANGE)。基本型包括:整型(int)、实型(real)、字符串型(string)和布尔型(bool);空间型包括:点型(point)、点集型(points)、线型(line)和区域型(region);基本时间型包括:时刻(instant);时间型包括:移动型(moving)、及时型(intime);域型包括:域型(range)。其中时间型由基本型和空间型组合而成,例如:moving(point)、moving(region);域型由基本型和基本时间型组合而成,例如:range(int)、range(real)。这些数据类型的关系如表1所示。

表1 移动对象数据模型

类型	模型
<i>int, real, string, bool</i>	→BASE
<i>Point, points, line, region</i>	→SPATIAL
<i>Instant</i>	→TIME
<i>moving, intime</i>	BASE ∪ SPATIAL →TEMPORAL
<i>Range</i>	BASE ∪ TIME →RANGE

^{*})本文由国防973重大基础研究项目资助,编号:973-1-4-2。吴 劲 博士生,主要研究方向为网络及数据库技术。卢显良 教授,博士生导师,主要研究方向为网络技术及应用。

3.1 基本型(BASE)

基本型包括:整型(int)、实型(real)、字符串型(string)和布尔型(bool),若某个域未定义,用符号“ $\perp$ ”来表示。

定义1 数据类型为 α 的集合表示为“ A_α ”,这里 α 可以是类型int、real、string和bool,具体表示如下:

$$A_{\text{int}} \triangleq \mathbb{Z} \cup \{\perp\},$$

$$A_{\text{real}} \triangleq \mathbb{R} \cup \{\perp\},$$

$$A_{\text{string}} \triangleq \mathbb{V} \cup \{\perp\}, \text{ 其中 } \mathbb{V} \text{ 是一个有限字符集,}$$

$$A_{\text{bool}} \triangleq \{\text{FALSE}, \text{TRUE}\} \cup \{\perp\}.$$

有时,某个域不容许有未定义值,不包含未定义值的集合可以表示为 A_α^* : $A_\alpha^* \triangleq A_\alpha \setminus \{\perp\}$ 。

3.2 空间型(SPATIAL)

空间型包括:点型(point)、点集型(points)、线型(line)和区域型(region),如图1所示。point表示欧几里得平面上的一个点;points表示数量有限的点集;line表示平面上的有限连续曲线集;region表示有限的互不相交的区域的集合,每个区域(也称为“脸”)内部可以有洞。

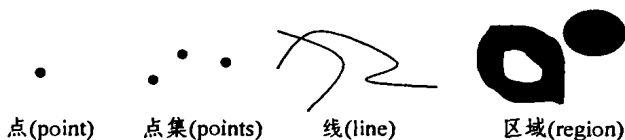


图1 基本空间数据类型

定义2 point和points类型的集合分别定义如下:

$$A_{\text{point}} \triangleq \mathbb{R}^2 \cup \{\perp\}$$

$$A_{\text{points}} \triangleq \{P \subseteq \mathbb{R}^2 \mid P \text{ 是有限的}\}$$

为了定义“线”,我们首先要定义曲线的概念。

定义3 我们可以把曲线映射到区间 $[0,1]$ 上, $f:[0,1] \rightarrow \mathbb{R}^2, \forall a, b \in [0,1]: f(a) = f(b) \Rightarrow a = b \vee \{a, b\} = \{0, 1\}$ 。

令 $\text{rng}(f) = \{p \in \mathbb{R}^2 \mid \exists a \in [0,1]: f(a) = p\}$;若曲线 f, g 相等,可表示为 $\text{rng}(f) = \text{rng}(g)$;点 $f(0)$ 和 $f(1)$ 是 f 的端点,如果 $f(0) = f(1) = p$,我们称 f 是以 p 为端点的环。

定义4 令 $Q \subseteq \mathbb{R}^2$ 且 $p \in Q$,若 p 是孤立于(isolated) Q 的: $\Leftrightarrow \exists \epsilon \in \mathbb{R}, \epsilon > 0: U(p, \epsilon) \cap (Q \setminus \{p\}) = \emptyset$ 。

这里 $U(p, \epsilon)$ 表示以 ϵ 为半径围绕 p 的区域。所有在 Q 内的孤立点的集合表示为“isolated(Q)”。

定义5 令 C 为定义3中定义的所有曲线的集合, $C' \subset C$,若 C' 被称为“简单”曲线集: $\Leftrightarrow \forall c_1, c_2 \in C': \text{isolated}(\text{rgn}(c_1)) \cap \text{rgn}(c_2) = \emptyset$ 是有有限集。

定义6 曲线 f, g 准不相交的条件是: $\forall a, b \in (0,1)$,有 $f(a) \neq f(b)$;若交点为 p ,则 $\exists a, b \in (0,1), f(a) = p = g(b)$ 。

定义7 S 是一曲线集, $C \subseteq S$, S 中的 C 集合体是有限集的条件如下:

(1) $\forall f, g \in C, f \neq g$: 称 f 和 g 准不相交。

(2) $\forall f, g \in C, f \neq g$: f 和 g 相交于 $p \Rightarrow (\exists h \in C, f \neq h \neq g, f$ 和 h 相交于 $p) \vee (f$ 或 g 是以 p 为端点的圈)。

C 集合体的点集表示为“points(C)”,所有 S 中 C 集合体的集合表示为“CC(S)”。

定义8 S 为简单曲线集,line类型的集合定义如下:

$$A_{\text{line}} \triangleq \{Q \subseteq \mathbb{R}^2 \mid \exists C \in \text{CC}(S): \text{points}(C) = Q\}$$

定义9 两个闭集 Q 和 S 若是准不相交的: $\Leftrightarrow Q \cap S$ 是有限的。

定义10 对于曲线集 S , S 中的 R 集合体是一个非空的

有限正则闭集则:(1) R 中任两个元素都是准不相交的;(2) $\forall r \in R, \exists c \in \text{CC}(S): \partial r = \text{points}(c)$ 。

这里“ ∂r ”表示 r 的边界, R 集合体的每个元素都称为“脸(face)”,所有脸中点的并集表示为:points(R),所有 S 中的 R 集合体的集合表示为“RC(S)”。

定义11 S 为简单曲线集,region类型的集合定义如下:

$$A_{\text{region}} \triangleq \{Q \subseteq \mathbb{R}^2 \mid R \in \text{RC}(S): Q = \text{points}(C)\}$$

另外我们定义: $A_{\alpha^*} \triangleq A_\alpha \setminus \{\emptyset\}$

3.3 基本时间型(TIME)

时刻(instant)表示时间轴上的特定点,我们认为时间是线性且连续的,可以用实数来表示。

定义12 instant类型的集合定义如下:

$$A_{\text{instant}} \triangleq \mathbb{R} \cup \{\perp\},$$

3.4 时间型(TEMPORAL)

从基本型和空间型,我们可以导出相应的时间型。moving类型使类型 α 与时间建立一种联系。intime类型使类型 α 与类型instant建立一种联系。

定义13 moving(α)类型的集合定义如下:

$A_{\text{moving}(\alpha)} \triangleq \{f \mid f: A_{\text{instant}} \rightarrow A_\alpha^* \text{ 是一个局部函数} \wedge \Gamma(f) \text{ 是有限的}\}$

这里, f 是集合moving(α)中的值,它是随时间变化的函数,“ $\Gamma(f)$ 是有限的”表示 f 是由有限数量的一些连续部分组成。

对于所有的“moving”类型,我们加入前缀“m”,简写为:mpoint, mpoints, mline, mregion, mint, mreal, mstring 和 mbool。

定义14 intime(α)类型的集合定义如下:

$$A_{\text{intime}(\alpha)} \triangleq A_{\text{instant}} \times A_\alpha$$

3.5 域型(RANGE)

对于所有时间型的操作都是和域相对应,例如,moving(real)的值在一维域上,域型主要用于表示实数或整数的区间集合。

定义15 α -interval 定义如下:

集合 $X \subseteq A_\alpha^*, \forall x, y \in X, \forall z \in A_\alpha^*, \text{有 } x < y < z \Rightarrow z \in X$ 。

若两个 α -interval是相邻的,则它们必不相交且它们的并集也是 α -interval。

α -range是不相交且不相邻的区间(interval)的有限集。对于 α -range R ,points(R)表示所有区间的并集。

定义16 range(α)类型的集合定义如下:

$$A_{\text{range}(\alpha)} \triangleq \{X \subseteq A_\alpha^* \mid \exists \alpha\text{-range } R: X = \text{points}(R)\}$$

我们往往会对时间域非常感兴趣,所以可引入一特殊类型:periods = range(instant)。

4 移动数据模型的应用

下面我们将简单介绍一下基于移动数据类型的基本操作,以及如何利用这些类型和操作进行移动数据查询。

4.1 基本操作

移动数据类型的操作主要应遵循三个主要原则:(一)操作应尽可能地通用;(二)应保持非时间型和时间型操作的一致性;(三)要与时空数据的特点相结合。

根据这三个主要原则,移动数据的操作可分成两大类:一类是针对非时间型的操作,如表2所示;另一类是针对时间型的操作,如表3所示。

表2 非时间型的操作

类型	操作
谓词	isempty, =, ≠, intersects, inside, <, ≤, ≥, >, before, touches, attached, overlaps, on-border, in-interior
集合操作	intersection, union, minus, crossings, touch-points, common-border
聚合	min, max, avg, center, single
数词	No-components, size, perimeter, duration, length, area
距离和方向	Distance, direction
基本操作	and, or, not

表3 时间型的操作

类型	操作
域映射	deftime, rangevalues, locations, trajectory, route, traversed, inst, val
域操作	atinstant, atperiods, initial, final, present, at, atmin, atmax, passes
时间	When
变化率	derivative, speed, turn, velocity

4.2 移动对象数据查询

本小节我们将举一些例子来说明如何利用移动数据模型来进行移动数据查询。

1) 非时间型操作的应用

例1 我们用本文第3节定义的移动数据类型来表示城市、省份、河流和道路:

```
city (name: string, pop: int, center: point);
province (name: string, area: region);
river (name: string, route: line);
highway (name: string, route: line);
```

例2 表示查询“国道 AAA 和长江的交点”:

```
LET 交点 = ELEMENT(
  SELECT single (crossings (R.route, H.route))
  FROM river R, highway H
  WHERE R.name = “长江” and H.name = “AAA” and
    R.route intersects H.route)
```

例2中的 crossings 操作返回的是“点集 points”类型, single 操作把 points 类型转化为 point 类型。

例3 表示查询“四川和贵州的边界长度”:

```
LET 四川 =
  ELEMENT (SELECT area FORM province WHERE name =
    “四川”);
LET 贵州 =
  ELEMENT (SELECT area FORM province WHERE name =
    “贵州”);
length (common-border (四川, 贵州))
```

例4 表示查询“重庆以北400公里内的城市”:

```
LET 重庆 = ELEMENT (SELECT center FORM city WHERE
  name = “重庆”);
SELECT name FORM city
WHERE distance (center, 重庆) < 400 and
  direction (重庆, center) >= 45 and
  direction (重庆, center) <= 135
```

例4中的 direction 函数值的单位是“度”(0≤angle≤360)。

2) 时间型操作的应用

例5 我们用本文第3节定义的移动数据类型来表示飞机和地点:

```
flight (airline: string, from: string, to: string, route: mpoint);
site (name: string, pos: point)
```

例6 表示查询“SZ8741航班在贵州省内飞行的长度”:

```
LET route8741 = ELEMENT (
  SELECT route FROM flight WHERE airline = “SZ8741”);
length (intersection (贵州, trajectory (route8741)))
```

例7 表示查询“什么时候 SZ8741航班经过贵阳机场, 距离贵阳机场的距离是多少?”:

```
LET 贵阳机场 =
  ELEMENT (SELECT pos FROM site WHERE name = “贵阳
  机场”);
LET pass = closest (route8741, 贵阳机场);
inst (pass); distance (贵阳机场, val (pass))
```

closest 操作的作用是: mpoint × point → intime (point), 即返回飞机靠近贵阳机场的时间和位置。

例8 表示查询“在什么时候、什么地点, SZ8741航班进入贵州境内?”

```
LET entry = initial (at (route8741, 贵州));
inst (entry); val (entry)
```

例9 表示查询“在地图上标出 SZ8741航班飞行速度超过800km/h 的部分”:

```
trajectory (atperiods (route8741, deftime (at (speed (route8741) >
  800, TRUE))))
```

从以上的几个例子我们可以看出, 我们可以把移动对象数据类型和操作插入关系型、面向对象型或其它 DBMS 数据模型中, 称为数据切片, 以此来扩充 DBMS, 使之能够支持移动对象应用。

结束语 移动对象技术具有非常强的实用性, 对它的研究, 不仅可以推动时空数据库相关技术的进步, 而且随着无线网络应用的不断深入, 该技术还可广泛应用于交通、旅游及军事等系统中, 具有巨大的市场价值。

参考文献

- Erwig M, Güting R H, Schneider M, Vazirgiannis M. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. GeoInfo. 1999. 269~296
- Forlizzi L, Güting R H, Nardelli E, Schneider M. A data model and data structures for moving objects databases. In: Proc. of the ACM SIGMOD Conf. on Management of Data (Dallas, TX, May), 1999. 319~330
- Güting R H, et al. A foundation for representing and querying moving objects. Tech. Rep. Informatik 238, FernUniversität, Hagen. 1998. <http://www.fernuni-hagen.de/inf/pi4/papers/Foundation.ps.gz>
- Wijzen J. Design of temporal relational databases based on dynamic and temporal functional dependencies. In: Proc. of the Intl. Workshop on Temporal Data-bases, Zurich, Switzerland, Sept. 1995
- Loeckx J, Ehrich H-D, Wolf M. Specification of Abstract Data Types. Wiley/Teubner computing series. John Wiley and Sons, Inc., New York, NY, 1997
- Sistla A P, Wolfson O, Chamberlain S, Dao S. Modeling and querying moving objects. In: Proc. of the 1997 Intl. Conf. on Data Engineering, 1997. 422~432
- Vazirgiannis M, Theodoridis Y, Sellis T. Spatio-temporal composition and indexing for large multimedia applications. Multimedia Syst. 1998. 284~298
- Wolfson O, et al. Moving Objects Databases: Issues and Solutions. In: Proc. of the tenth Intl. Conf. on Scientific and Statistical Database Management (SSDBM98). Capri (Italy), July 1998. 111~122