

时空数据库中的时空拓扑查询^{*})

The Query of Topological Relationships in Spatio-Temporal Databases

邹永娟 秦小麟

(南京航空航天大学计算机科学与工程系 南京210016)

Abstract In the field of spatio-temporal databases, the algorithms of spatio-temporal query operations are a key problem, where the most important part is on the topological relationships between two objects over time. On the basis of the topological theory of spatial objects, the paper gives abstract definitions of various basic spatio-temporal topological predicates by integrating the factor of time. For the purpose of reflecting sufficiently the dynamic nature of spatio-temporal objects, complex topological predicates are explored. Finally, it gives some applications of Spatio-Temporal Query Language, STQL.

Keywords Spatio-temporal objects, Topological relationships, Spatio-temporal databases

1 引言

现代社会大量的科学应用,如航空系统的飞行器导航、森林火灾探测、乃至于人们生活中应用越来越广泛的移动通讯手机等等,这些都导致有必要研究一个能兼顾时空两属性的数据库管理系统。研制一个能有效表示和处理时空对象的DBMS,其中一项重要工作就是对时空对象进行空间分析操作。典型的时空对象的分析操作估计有近百个,大致划分为七类^[1]:a)时空对象间拓扑关系的操作,如求一个移动点是否在一个移动区域中;b)时空对象与空间对象间拓扑关系的操作,如求一个移动点是否在一个固定区域中;c)返回值为时空对象的操作,如求移动点的轨迹;d)返回值为空间对象的操作,如求一个移动点在某一个特定时刻的空间位置;e)返回值为单个数值的操作,如求一个移动点的速度;f)返回值为多个数值的操作,如求两个移动点间的距离;g)在时空对象集合上的操作,如在一些移动区域上进行 FUSION 操作。

我们正在研究支持移动空间对象的子系统 ROSTE,在该系统中融入各类时空谓词形成时空查询语言 STQL (Spatio-Temporal Query Language),本文主要讨论时空对象的拓扑关系操作。

2 时空操作分析

虽然时空数据库处理的信息主要是时空对象(移动空间对象),但本质上仍然是空间数据,大部分对时空对象的空间操作分析算法均可建立在空间对象(point, line, region)的操作算法之上,其中时空对象的拓扑关系分析可基于对空间对象的拓扑关系分析。Max J. Egenhofer 和 John R. Herring 的 9-intersection 模型提供了经典的二维空间中所有任意两种空间类型之间的拓扑谓词^[2]。例如两个区域间共存在8个拓扑谓词: disjoint, meet, overlap, coveredBy, covers, inside, contains, equal。

把时间因素看作第三维,时空拓扑关系便可被看作存在于三维空间之中,图1分别例示了一个移动点和一个移动区域。

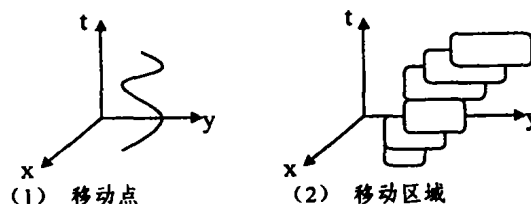


图1 时空对象的三维表现

本文主要对时空点(移动点)和时空区域(移动区域)进行分析。至于时空线(移动线),由于其可看作移动点的轨迹,并且其移动特性类似于移动点,因此不再做讨论。

2.1 数据模型

根据时空对象的平滑移动或变化,我们的数据模型将基于时间连续性,即 $\text{time} = \mathbb{R}$ 。对基本数据类型 α , 引用时间函数 τ , 有: $\tau(\alpha) = \text{time} \rightarrow \alpha$, $\tau(\alpha)$ 就表示从时间 time 到类型 α 的所有时态函数。当 α 是空间数据类型(如 point)时, $\tau(\text{point})$ 可求出移动点在不同时间的空间位置。另外, α 也可以为数值型或布尔型,这在定义时空对象的操作时是非常重要的,例如在求移动点和移动区域的随时间变化的距离(即返回多个数值的操作)时,就要有如下操作:

$\text{Distance}; \tau(\text{point}) \times \tau(\text{region}) \rightarrow \tau(\text{real})$

如果我们定义时态对象为随时间变化的对象,其反映为一组数据,则原则上可将所有非时态操作提升作用到时态对象上,其返回值也为时态对象。其形式表示为,对于任意非时态函数 $f; \alpha_1 \times \dots \times \alpha_n \rightarrow \beta$, 其相对应的时态函数为^[3]: $\uparrow f; \tau(\alpha_1) \times \dots \times \tau(\alpha_n) \rightarrow \tau(\beta)$, 其中 $\uparrow f(s_1, \dots, s_n) = \{(t, f(s_1(t), \dots, s_n(t))) \mid t \in \text{time}\}$ 。

为便于区分,我们用首位字母大写来代表时态函数、时态数据类型和时态对象。例如 $\text{Distance} = \uparrow \text{distance}$, 其中 distance 表示点与区域的距离,有 $\text{distance}; \text{point} \times \text{region} \rightarrow \text{real}$, 则 $\text{Distance}; \text{Point} \times \text{Region} \rightarrow \text{Real}$ 。同样空间拓扑谓词也有其时态形式,例如对于空间谓词 inside 有 $\text{inside}; \text{point} \times \text{region} \rightarrow \text{bool}$, 其时态形式为 $\uparrow \text{inside}; \text{Point} \times \text{Region} \rightarrow \text{Bool}$, 则 $\text{Inside} = \uparrow \text{inside}$, 当任意时刻移动点均在移动区域中时,

^{*})国家自然科学基金资助项目(69973022)。邹永娟 硕士生,主要研究方向为时空数据库。秦小麟 教授,博士生导师,当前研究兴趣为空间分析数据库、时空数据库、GIS等。

返回值为真,否则为假。其中 $\text{Point}=\tau(\text{point})$, $\text{Region}=\tau(\text{region})$, $\text{Real}=\tau(\text{real})$ 。

2.2 时空拓扑谓词

2.2.1 谓词规范 我们将不含空间对象(或时空对象)的谓词称为原始谓词,如基于空间对象的 meet 、 intersect 、 overlap 等和基于时空对象的 Meet 、 Intersect 、 Overlap 等。时空对象的拓扑分析离不开对时空对象的分析,因此在原始谓词的基础上,加上前缀 p/P 、 r/R 分别表示点和区域的空间对象/时空对象形式。时空对象的拓扑关系分析按时间分可包含两类:i)在某一时刻的拓扑关系,可用下标 t 表示;ii)在某一时间域内的拓扑关系,这里用下标 r 表示给定时间域,当无下标时,表示缺省状况(见2.2.3节)。例如: PR-Meet_t 表示移动点在 t 时刻居于移动区域的边界上, PR-Meet_r 表示移动点在某个时间域内都居于移动区域的边界上。

2.2.2 生命期 时空对象的生命期,对于静态空间对象,可以认为其生命期为 $(-\infty, +\infty)$,而时空对象由于其动态性,常具有一定的起始时间和终止时间。这里只考虑连续时间域,因为不连续的时间域可以分段考虑。考虑在一定时间域中,对象间(O_1 与 O_2)拓扑关系就必须考虑二者的生命期(T_1 与 T_2),可分为五个方面:(i) $(-\infty, +\infty)$; (ii) $T_1 \cup T_2$; (iii) T_1 ; (iv) T_2 ; (v) $T_1 \cap T_2 \neq \emptyset$ 。其中第一种情况规定的最为严格,要求两对象一直有定义($T_1=T_2=(-\infty, +\infty)$),即一直存在;第二种其实规定两对象的生命期($T_1=T_2$)相同,即在任一时空对象的生命期内判断拓扑关系;第三和第四分别要求 $T_1 \subseteq T_2$ 和 $T_1 \supseteq T_2$;而第五则约束性较小,仅要求 $T_1 \cap T_2 \neq \emptyset$ 。

因此,时间域分别抽象表示为 T_∞ 、 T_U 、 T_{T_1} 、 T_{T_2} 、 $T_\cap$,用 T_v 通用表示该五类。

2.2.3 时空基本拓扑谓词 如果用 O 代表时空对象, o 代表空间对象(非时态对象), X/x 表示原始谓词的时态/非时态形式,则得到时空对象的基本拓扑谓词的抽象表示,如 $Oo-X_t$ 表示在某一时刻时空对象(O)和空间对象(o)的某一拓扑关系(X), $Oo-X_r$ 则表示给定时间域内时空对象(O)和空间对象(o)的某一拓扑关系(X)。下表列出了点/区域之间的所有原始谓词。

	拓扑关系(原始谓词 x)
point-point	equal, disjoint
point-region	Inside, disjoint, meet
Region-region	disjoint, meet, overlap, coveredBy, covers, inside, contains, equal

(1)基于时刻的查询。对于在给定时刻 t 对空间对象/时空对象间的时空拓扑谓词有如下三种抽象形式:

- (a) $oO-X_t(o, O, t) \Leftrightarrow t \in T_{T_2} \wedge oo-x(o, O(t))$ 。
- (b) $Oo-X_t(O, o, t) \Leftrightarrow t \in T_{T_1} \wedge oo-x(O(t), o)$ 。
- (c) $OO-X_t(O_1, O_2, t) \Leftrightarrow t \in T_\cap \wedge oo-x(O_1(t), O_2(t))$ 。

(2)基于时间域的查询。对于给定时间域 T' ,空间对象/时空对象间的时空拓扑谓词有如下三种抽象形式:

- (a) $oO-X_r(o, O, T') \Leftrightarrow (T' \subseteq T_{T_2}) \wedge (\forall t \in T' \ oO-X_t(o, O, t))$ 。
- (b) $Oo-X_r(O, o, T') \Leftrightarrow (T' \subseteq T_{T_1}) \wedge (\forall t \in T' \ Oo-X_t(O, o, t))$ 。
- (c) $OO-X_r(O_1, O_2, T') \Leftrightarrow (T' \subseteq T_\cap) \wedge (\forall t \in T' \ OO-X_t(O_1, O_2, t))$ 。

(3)缺省查询。对“Equal”原始谓词,它要求两个时空对象的生命期相同,且在此生命期中的任意时刻其空间位置相等(equal)。对“Inside”原始谓词,当对移动点和移动区域进行此拓扑判断时,则只需要满足“在移动点的生命期中,该点一直在移动区域之中”的条件,从而不需要移动点和移动区域的生命期相同。

对原始谓词 X ,有 $X(S_1, \dots, S_n) := \{(t, x(S_1(t), \dots, S_n(t))) \mid t \in \text{time}\}$ 。若用 $\forall \cap x$ 表示 $\forall t \in T_\cap x(S_1(t), S_2(t))$,则可定义各种原始谓词的缺省时间域时的形式:

$\text{Disjoint} \Leftrightarrow \forall \cap \text{disjoint}$ $\text{Meet} \Leftrightarrow \forall \cap \text{meet}$ $\text{Overlap} \Leftrightarrow \forall \cap \text{overlap}$ $\text{Equal} \Leftrightarrow \forall \cap \text{equal}$ $\text{Covers} \Leftrightarrow \forall r_2 \text{covers}$ $\text{Contains} \Leftrightarrow \forall r_2 \text{contains}$ $\text{CoveredBy} \Leftrightarrow \forall r_1 \text{coveredBy}$ $\text{Inside} \Leftrightarrow \forall r_1 \text{inside}$

上面定义的8种原始谓词是针对(移动)区域间的拓扑关系而言的,其中 Disjoint 、 Meet 和 Inside 也可用于移动点和移动区域,最终的谓词形式只要在这些原始谓词前面加上表明对象的前缀(p/P 、 r/R)即可。

2.2.4 时空复杂拓扑谓词 上述对时空对象的拓扑关系操作并未充分考虑时空对象的动态性。设想查询敌机有无进入过我方领空,这个动态过程(Pr-Enter)应包括三个拓扑谓词: Pr-Disjoint_r , Pr-Meet_r / Pr-Meet_t , Pr-inside_r 。其中 Meet 有两种情况:根据敌机在我方领空边界有无停留分为时刻和时间域查询。

设函数 $ts(T)$ 和 $te(T)$ 分别得到时间域 T 的下限和上限, $[]$ 表示可选项, P 和 r 分别表示动态点和静态区域,则有:

$\text{Pr-Enter}(P, r, [T]) \Leftrightarrow ((\exists T'_1 \text{Pr-Disjoint}_r(P, r, T'_1)) \wedge (\exists t_2 \text{Pr-Meet}_t(P, r, t_2)) \wedge (\exists T'_3 \text{Pr-Inside}_r(P, r, T'_3)) \wedge (te(T'_1) = t_2 = ts(T'_3)) [\wedge (T'_1 \subseteq T \wedge T'_3 \subseteq T)]) \vee ((\exists T'_1 \text{Pr-Disjoint}_t(P, r, T'_1)) \wedge (\exists T'_2 \text{Pr-Meet}_r(P, r, T'_2)) \wedge (\exists T'_3 \text{Pr-Inside}_r(P, r, T'_3)) \wedge (te(T'_1) = ts(T'_2)) \wedge (te(T'_2) = ts(T'_3)) [\wedge (T'_1 \subseteq T \wedge T'_2 \subseteq T \wedge T'_3 \subseteq T)]))$

上式可简化为:

$\text{Pr-Enter} \Leftrightarrow \text{Pr-Disjoint}_r \rightarrow \text{Pr-Meet}_r / \text{Pr-Meet}_t \rightarrow \text{Pr-Inside}_r$

其中: $\rightarrow$ 表示拓扑关系发生变化的连续的先后顺序,要注意的是 $\rightarrow X_t \rightarrow X_r$ 等价于 $\rightarrow X_r$ 。这里用上标 $-$ 表示某谓词的逆,设 $X \Leftrightarrow X_1 \rightarrow X_2$, 则 $X^- \Leftrightarrow X_2 \rightarrow X_1$ 。

如下是移动点与区域之间的复杂拓扑谓词:

$\text{Pr-Touch} \Leftrightarrow \text{Pr-Disjoint}_r \rightarrow \text{Pr-Meet}_t \rightarrow \text{Pr-Disjoint}_r$ 。
 $\text{Pr-Snap} \Leftrightarrow \text{Pr-Disjoint}_r \rightarrow \text{Pr-Meet}_t / \text{Pr-Meet}_r$ 。
 $\text{Pr-Release} \Leftrightarrow \text{Pr-Meet}_t \rightarrow \text{Pr-Disjoint}_r$ 。
 $\text{Pr-Bypass} \Leftrightarrow \text{Pr-Disjoint}_r \rightarrow \text{Pr-Meet}_r / \text{Pr-Meet}_t \rightarrow \text{Pr-Disjoint}_r$ 。
 $\text{Pr-Excuse} \Leftrightarrow \text{Pr-Meet}_t \rightarrow \text{Pr-Disjoint}_t \rightarrow \text{Pr-Meet}_r$ 。
 $\text{Pr-Enter} \Leftrightarrow \text{Pr-Disjoint}_r \rightarrow \text{Pr-Meet}_r / \text{Pr-Meet}_t \rightarrow \text{Pr-Inside}_r$ 。
 $\text{Pr-Leave} \Leftrightarrow \text{Pr-Enter}^-$ 。
 $\text{Pr-Cross} \Leftrightarrow \text{Pr-Enter}^- \text{Pr-Leave}$ 。

将前缀换为 pR 、 PR 即可得到点与移动区域、移动点与移动区域之间的复杂拓扑谓词。移动区域间的复杂拓扑谓词可表示为:

$\text{RR-Enter} \Leftrightarrow \text{RR-Disjoint}_r \rightarrow \text{RR-Meet}_r / \text{RR-Meet}_t \rightarrow \text{RR-Overlap}_r \rightarrow \text{RR-Inside}_r$ 。
 $\text{RR-Leave} \Leftrightarrow \text{RR-Enter}^-$ 。
 $\text{RR-Cross} \Leftrightarrow \text{RR-Enter} \rightarrow \text{RR-Leave}$ 。
 $\text{RR-Melt} \Leftrightarrow \text{RR-Disjoint}_r \rightarrow \text{RR-Meet}_r / \text{RR-Meet}_t \rightarrow \text{RR-Overlap}_r \rightarrow \text{RR-equal}_r$ 。
 $\text{RR-Passin} \Leftrightarrow \text{RR-Disjoint}_r \rightarrow \text{RR-Meet}_r / \text{RR-Meet}_t \rightarrow \text{RR-Overlap}_r$ 。
 $\text{RR-Passout} \Leftrightarrow \text{RR-Passin}^-$ 。
 $\text{RR-Pass} \Leftrightarrow \text{RR-Passin} \rightarrow \text{RR-Passout}$ 。

将前缀换为 rR 、 Rr 即可得到区域与移动区域、移动区域与区域之间的复杂拓扑谓词。

根据不同的应用会有多种复杂拓扑关系,以上所列出的只是其中一小部分,但已体现时空对象的动态性了。

3 时空查询语言

目前我们正在研究支持移动空间对象的子系统 ROSTE (RObust Spatial-Temporal Extension) 包括移动空间对象的数据类型、移动空间对象的存储结构、移动空间对象间的操作、移动空间对象与空间对象的操作等。并且正在把该子系统与可扩充 DBMS-AMOS^[7]集成,生成时空 DBMS——STDBS。

STDBS 的查询语言为 STQL, 可以进行一些非常复杂的时空分析操作。设有模式如下:

```
planes(airline: charstring, id: charstring, Flight: Point)
cities(name: charstring, id: charstring, position: region)
weather(kind: charstring, Position: Region)
```

例1 找出所有两架飞机航线间的距离小于1公里的飞机。

```
SELECT p. airline, p. id, q. airline, q. id
FROM planes p, planes q
WHERE min(Distance(p. Flight, q. Flight)) < 1
```

例2 查询7:00到7:20时间内,国航300飞机所在的城市。

```
SELECT cities. name FROM planes, cities
WHERE Pr_Inside(planes. Flight, cities. position, 7:00, 7:20) AND
planes. airline = "Airchina300"
```

例3 查询曾穿过暴风雪或浓雾区的飞机数,并分组表示。

```
SELECT COUNT(*) FROM planes, weather
WHERE weather. kind = "snowstorm" OR weather. kind = "thickfog"
GROUP BY PR_Cross(planes. Flight, weather. Position)
```

结论及未来有关工作 本文讨论了时空数据库中的各种时空查询,主要对时空拓扑谓词进行了定义。为了满足具体应用的要求,文章还重点探讨了时空对象的复杂拓扑谓词,最后分类例示了具体的 STQL 查询。本文给出的拓扑谓词可用作时空 DBMS 中时空对象拓扑关系操作。我们已设计和实现了基于 Realms^[8]的时空对象表示、存储与管理,并且实现了大部分基于平面扫描算法的各种时空拓扑操作算法,算法的效

率是令人满意的,作者将另文讨论时空分析操作算法的设计和实现。

参考文献

- 1 秦小麟. 移动空间数据类型和操作的初步研究. 计算机科学, 2000, 27(1)
- 2 Egenhofer M J, Herring J R. Categorizing Binary Topological Relations Between Regions, Lines, and Points in Geographic Databases. [Technical Report]. Department of Surveying Engineering, University of Maine, 1990
- 3 Erwig M, Güting R H, Schneider M, Vazirgianis M. Abstract and Discrete Modeling of Spatio-Temporal Data Types. In: Proc. of the 6th ACM Intl. Workshop on Geographical Information Systems (ACM-GIS), 1998
- 4 Erwig M, Schneider M. Spatio-Temporal Predicates. [Informatik-Report]. FernUniversity Hagen, Dec. 1999
- 5 Güting R H, Böhlen M H, et al. A Foundation for Representing and Querying Moving Objects. [Informatik-Report 238]. FernUniversität Hagen, 1998
- 6 Erwig M, Schneider M. Developments in Spatio-Temporal Query Languages. IEEE Int. Workshop on Spatio-Temporal Data Models and Languages, 1999
- 7 Karlsson J S, et al. AMOS User's Guide. Linköping University, Sweden, 1996
- 8 Güting R H, Schneider M. Realms: A Foundation for Spatial Data Types in Database Systems. In: Proc. 3rd Intl. Symposium on Large Spatial Databases, Singapore, June 1993. 14~35
- 9 Al-Taha K, Snodgrass R T, Soo M D. Bibliography on Spatio-Temporal Databases. A CM SIGMOD Record, 1994, 22(1): 59~67
- 10 Böhlen M H, Jensen C. Seamless Integration of Time into SQL. [Technical Report R-96-49]. Dept. of Computer Science, Aalborg University, Dec. 1996
- 11 Erwig M, et al. Temporal and Spatio-Temporal Data Models and Their Expressive Power. FernUniversity Hagen. [Informatik-Report]. Dec. 1997
- 12 Claramunt C, Thedriault M. Managing Time in GIS: An Event-Oriented Approach, Recent Advances in Temporal Databases. Springer-Verlag, 1995. 23~42
- 13 Güting R H. An Introduction to Spatial Database Systems. VLDB Journal, 1994, 3(4): 357~399

(上接第47页)

3) 为对象行为和属性的细节建模, 精化类的结构, 建立动态设计模型;

4) 数据库设计。按照前面所介绍的对象映射为关系数据库的规则进行数据设计;

5) 设计与问题域无关的用于系统实现的服务类, 补充完善设计对象模型, 例如数据库访问、用户接口等。

4.3 基于UML的MIS分析与设计方法的优点

上面的基于UML的MIS分析与设计方法相对传统的基于E-R图的结构化分析与设计方法具有以下优点:

- 对象与客观世界的实体相似, 便于为客观世界建立模型。

- UML 提供了大量的易于理解的模型图, 方便统一建模。

- 从需求阶段到实现阶段, 使用统一的对象概念, 开发过程中阶段的改变, 不需要方法或范型的转换, 可以做到平滑的过渡。

- 结合 COM+ / CORBA / EJB 等分布式计算技术, 有利于建立大型分布式的 MIS 的应用。

- 能充分利用 OOP 开发工具的潜力, 开发出高质量的代

码, 有利于代码重用及后期的维护。

- 软件的分析设计成果得到了较大程度的复用。

小结 本文提出了一种基于UML的MIS分析与设计方法, 并对其中的一个关键问题——持久对象与关系数据库的映射规则进行了概括性总结。这种方法克服了传统的基于E-R模型的结构化分析与设计方法的缺点, 在MIS分析与设计的具体实践中取得了比较好的结果。

参考文献

- 1 Booch, Jacobson, Rumbaugh. The Unified Modeling Language Reference Manual. Addison-Wesley, 1998. 11
- 2 Priestley M. Practical Object-Oriented Design with UML. 清华大学出版社(影印版), 2000. 12
- 3 刘超, 张莉编著. 可视化面向对象建模技术. 北京航空航天大学出版社, 1999. 7
- 4 Cord P, Yourdon E 著, 邵维忠等译. 面向对象的分析. 北京大学出版社, 1991. 5
- 5 Cord P, Yourdon E 著, 邵维忠等译. 面向对象的设计. 北京大学出版社, 1994. 12
- 6 Blaha M. Using UML to Design Database Applications OMT Associates, Inc.; William Premerlani, General Electric R & D, 1999
- 7 邵维忠, 杨美清. 面向对象的系统分析. 广西科技出版社, 1998. 12