

面向对象的时空数据模型*

舒红¹ 陈军² 杜道生¹ 周勇前¹

1. (武汉测绘科技大学测绘遥感信息工程国家重点实验室, 武汉市珞喻路 39 号, 430070)

2. (国家基础地理信息中心, 北京市海淀区紫竹院百胜村 1 号, 100044)

摘要 从存储管理和时空语义表现角度给出了时空数据模型的基本设计思想。同时, 从时空对象类型及其数据结构、时空操作方面定义了一个面向对象时空数据模型的逻辑结构。

关键词 时空数据模型; 时空语义; 面向对象; 时空复合; 时态对象代数

分类号 TP 311. 12; P208

地籍变更、海岸线变化、土地城市化、道路改线、环境变化管理等应用领域需要保存并有效地管理历史变化数据, 以方便将来重建历史状态、跟踪变化、预测未来, 要求有一个组织、管理、操作时空数据的高效 DBMS, 最重要的是要建立一个合适的时空数据模型。过去的空间数据库一般不保存历史变化或保留若干典型时间点的全局状态快照序列, 具有较弱的时空语义建模能力, 无法提供时态分析功能, 常常被称为静态空间数据库。时空数据模型是一种有效组织和管理时态地理数据, 属性、空间和时间语义更完整的地理数据模型。目前, 规范化的时空数据模型尚处在探索阶段, 以 Max J. Egenhofer, Donna Pequet, Michael F. Worboy, Geoffrey Edwards, Yvan Bedard, Wolfgang Kanz, 陈军为代表的学者们从概念、理论、结构、实现技术等方面对时空数据模型进行了广泛的研究。面向对象技术是支撑空间复杂对象建模的最有效手段。下面, 我们在分析时空数据建模的基本特点后, 给出时空数据模型设计的基本指导思想, 并设计了一种面向对象的时空数据模型。

1 时空数据模型设计的基本思想

一个合理的时空数据模型必须考虑以下几方面的因素: 节省存储空间、加快存取速度、表现时空语义。时空语义包括地理实体的空间结构、有效时间结构、空间关系、时态关系、地理事件、时空关系^[2]。时空数据模型设计的基本指导思想如下:

(1) 根据应用领域的特点(如宏观变化观测与微观变化观测)和客观现实变化规律(同步变

化与异步变化、频繁变化与缓慢变化), 折中考虑时空数据的空间/属性内聚性和时态内聚性的强度, 选择时间标记的对象。对于属性, 有属性数据项时间标记、实体时间标记、数据库时间标记; 对于空间, 有坐标点时间标记、弧段时间标记、实体时间标记、数据库时间标记^[8]。在应用强调空间/属性内聚性, 关心空间/属性的宏观变化及现实环境空间/属性状态存在大量同步变化现象时, 宜针对较大粒度对象进行有效时间标记(或全局状态时间标记)。反之, 应尽量将有效时间标记在小粒度对象上(局部状态时间标记)。若在需要全局状态时间标记的场合使用了局部状态时间标记, 则大量相同时间标记的存储是一种空间的冗余存储, 且弱化了空间/属性的结构完整性, 带来某时刻全局状态提取的困难。相反, 若在需要局部状态时间标记的场合使用了全局状态时间标记, 则存在大量不变空间/属性数据的冗余存储, 体现不出时间语义, 模糊了参加变化的真正对象。

(2) 同时提供静态(变化不活跃)、动态(变化活跃)数据建模手段(静态、动态数据类型和操作)。当前、历史等不同使用频率的数据分别组织存放, 以方便存取。一般地, 将当前数据存放在本地机磁盘上, 而将历史数据存放在远程服务器大容量光盘上。

(3) 数据结构里显式表达两种地理事件: 地理实体进化事件和地理实体存亡事件。地理事件以事件发生的相关源状态和终止状态表达, 构成地理实体存亡事件的源状态由参加事件的实体标识集合表示。时间的本质为事件发生的序列, 地理事件序列直接表明地理时间语义。常见的状态变化查询即地理事件查询。

收稿日期: 1996-05-20 舒红, 男, 27 岁, 助理研究员, 现从事时空数据库系统研究。

* 国家自然科学基金、国家教委霍英东教育基金及博士点基金资助项目, 编号 49471059。

(4) 时空拓扑关系一般指改变地理实体空间拓扑关系的拓扑事件间的时态关系。时空拓扑关系揭示了地理实体在时间和空间上的相关性。为了有效表达时空拓扑关系,需要存储空间拓扑关系的时变序列。

2 时空对象类型及其数据结构

在上面指导思想下,基于 Node-Arc-Polygon 数据结构^[9],我们设计了如下面向对象的时空数据模型:

除图 1 所示核心时空对象类型外,根据指导思想 2,我们还增补了若干时间对象类型和空间

对象类型,它们与时空对象类型一起构成一用于矢量格式时空数据建模的完整时空数据模型逻辑结构。

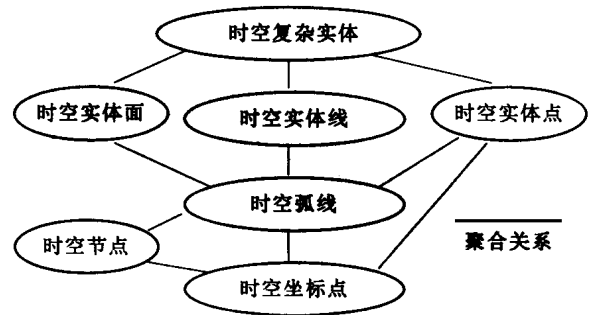


图1 面向对象的时空数据模型
Fig 1 An Object-oriented Spatio-temporal Data Model

基本的数据库类型对象: ...

基本时间对象: 日期 Date、时间 Time、日期时间 Datetime

复合时间对象:

时间点 T. TimePoint := Date | Time | Datetime

时间区间 T. TimePeriod := [Starttime: Date, Endtime: Date] | [Starttime: Time, Endtime: Time] | [Starttime: Datetime, Endtime: Datetime]

时间元素 T. TimeElement := {T. TimePeriod}

复合空间对象:

坐标点 S. CoordPoint := [x: Real | Number, y: Real | Number]

节点 S. Node := [Node_ Id: Number, Coordinate: S. CoordPoint]

弧段 S. Arc := [Arc_ Id: Number, Coordinate1: S. CoordPoint, Coordinate2: S. CoordPoint, ..., Coordinaten: S. CoordPoint, Leftnode_ Id: Number, Rightnode_ Id: Number, Leftpoly_ Id: Number, Rightpoly_ Id: Number]

实体点 S. EntityPoint := [Entity_ Id: Number, Coordinate: S. CoordPoint, U ser_ Id: Number]

实体线 S. EntityLine := [Entity_ Id: Number, Arc_ Id1: Number, Arc_ Id2: Number, ..., Arc_ Idm: Number, U ser_ Id: Number]

实体面 S. EntityArea := [Entity_ Id: Number, Arc_ Id1: Number, Arc_ Id2: Number, ..., Arc_ Idm: Number, U ser_ Id: Number]

复杂实体 S. EntityComplex := [Entity_ Id: Number, Object_ Id1: Number, Object_ Id2: Number, ..., Object_ Idm: Number, U ser_ Id: Number]

复合时空对象:

时空坐标点 ST. CoordPoint := [Coordinate: S. CoordPoint, t: T. TimePoint | T. TimePeriod | T. TimeElement]

时空节点 ST. Node := [Node: S. Node, t: T. TimePoint | T. TimePeriod | T. TimeElement]

时空弧段 ST. Arc := [[[Arc_ Id: Number, Point1: S. CoordPoint, Point2: S. CoordPoint, ..., Pointn: S. CoordPoint], t1: T. TimePoint, { [Leftnode: Number, t2: T. TimePoint] }, { [Rightnode: Number, t3: T. TimePoint] }, { [Leftpoly: Number, t4: T. TimePoint] }, { [Rightpoly: Number, t5: T. TimePoint] }] | [[Arc_ Id: Number, Point1: S. CoordPoint, Point2: S. CoordPoint, ..., Pointn: S. CoordPoint], t1: T. TimePeriod, { [Leftnode: Number, t2: T. TimePeriod] }, { [Rightnode: Number, t3: T. TimePeriod] }, { [Leftpoly: Number, t4: T. TimePeriod] }, { [Rightpoly: Number, t5: T. TimePeriod] }] | [[Arc_ Id: Number, Point1: S. CoordPoint, Point2: S. CoordPoint, ..., Pointn: S. CoordPoint], t1: T. TimeElement, { [Leftnode: Number, t2: T. TimeElement] }, { [Rightnode: Number, t3: T. TimeElement] }, { [Leftpoly: Number, t4: T. TimeElement] }, { [Rightpoly: Number, t5: T. TimeElement] }]]

时空实体点 ST. EntityPoint := [[[EntityPoint: S. EntityPoint, t: T. TimePoint]], {Resource. Entity. ids: Number}, U ser_ Id: Number] | [[[EntityPoint: S. EntityPoint, t: T. TimePeriod]], {Resource. Entity. ids: Number}, U ser_ Id: Number] | [[[EntityPoint: S. EntityPoint, t: T. TimeElement]], {Resource. Entity. ids: Number}, U ser_ Id: Number]

时空实体线 ST. EntityLine := [[[EntityLine: S. EntityLine, t: T. TimePoint]], {Resource. Entity. ids: Number}, U ser_ Id: Number] | [[[EntityLine: S. EntityLine, t: T. TimePeriod]], {Resource. Entity. ids: Number}, U ser_ Id: Number] | [[[EntityLine: S. EntityLine, t: T. TimeElement]], {Resource. Entity. ids: Number}, U ser_ Id: Number]

时空实体面 ST. EntityArea := [[[EntityArea: S. EntityArea, t: T. TimePoint]], {Resource. Entity. ids: Number}, U ser_ Id: Num-

```
ber] | [{"EntityArea": S.EntityArea, t: T.TimePeriod}], {Resource-Entity-ids: Number}, User-Id:
Number] | [{"EntityArea": S.EntityArea, t: T.TimeElement}], {Resource-Entity-ids: Number},
User-Id: Number]
```

```
时空复杂实体 ST-EntityComplex ::= [{"EntityComplex": S.EntityComplex, t: T.TimePoint}], {Resource-Entity-ids: Number}, User-Id: Number] | [{"EntityComplex": S.EntityComplex, t: T.TimePeriod}], {Resource-Entity-ids: Number}, User-Id: Number] | [{"EntityComplex": S.EntityComplex, t: T.TimeElement}], {Resource-Entity-ids: Number}, User-Id: Number]
```

上面表示中, 时间元素为若干有序不相交的时间区间的集合。引进时间元素的目的在于提供表达对象的有效时间结构。时间元素具有在“加+ (或并)、积* (或交)、减- (或差-)”运算下保持封闭的良好性质, 而时间区间却不具有如此性质。时空弧段与若干时间点的空间快照状态人为叠加在某一个时刻形成的碎分弧段 (又称时空复合弧段)、时空节点类似^[8]。“[]”为元组对象构造子, “{ }”为集合对象构造子, “ ”为序列对象构造子, “[]”为时态对象构造子, 各类型对象数据结构以BNF形式给出, “|”为可选符。所给出的对象基本类型可用于构造复杂的二维矢量空间的时空数据模型。

3 时空操作类型

如表 1, 按数据成分类型分类, 时空数据模型里的操作可分为属性操作(A A)、空间操作(S S)、时间操作(T T)、三者互操作(A S、A T、S A、S T、T A、T S)。

表 1 按数据内容的操作分类

Table 1 Classification of Operators by Content of Data

	属性	空间	时间
属性	A A	A S	A T
空间	S A	S S	S T
时间	T A	T S	T T

如表 2, 按数据性质分类, 时空数据模型里的操作可分为值操作(属性值操作、空间值操作、时间值操作)、关系操作(属性关系操作、空间关系操作、时间关系操作)、变化跟踪操作(属性变化跟踪 A A、空间变化跟踪 S S 操作)。

表 2 按数据性质的操作分类

Table 2 Classification of Operators by Nature of Data

	属性	空间
属性	A A	A S
空间	S A	S S

3.1 属性操作 A A

基本属性操作由对象代数或基于对象代数的对象查询语言给出, 复杂应用操作由用户编制程序实现。

3.2 空间操作 S S

a 空间对象成员数据获取/设置操作

坐标点 S-Coord Point: getx(), gety(), setx(), sety()

节点 S-Node: getcoordpoint(), setcoordpoint()

弧段 S-Arc: getstartnode(), getendnode(), getvertice(), setstartnode(), setendnode(), setvertice()

多边形 S-Polygon: getarcs(), setarcs()

实体点 S-Entity Point: getcoordpoint(), setcoordpoint()

实体线 S-Entity Line: getfirstnode(), getlastnode(), getarcs(), setfirstnode(), setlastnode(), setarcs()

实体面 S-Entity Area: getpolygon(), getcentroid(), setpolygon(), setcentroid()

复杂实体 S-Entity Complex: getentities(), Setentities()

b 空间对象计算函数

弧段 S-Arc: getlength()

多边形 S-Polygon: getlength(), getarea()

实体线 S-Entity Line: getlength()

实体面 S-Entity Area: getlength(), getarea()

c 空间数据查询操作

范围查询: In-window s(), In-Circle(), In-Polygon(), At-Location(), Near-to()

d 空间对象空间关系操作

空间对象空间关系操作主要分为判断两对象间成立否 (xx(Obj1, Obj2)) 或查询满足给定空间关系的对象 (xx(Obj1, ?))。空间关系操作的主要类型如下。

空间拓扑关系操作: disjoint, meet, overlap, cross, in

空间方向关系操作: east, west, south, north over, below (front, after, left, right, up, down)

空间距离关系操作: near, far

e 空间对象生成操作

缓冲: buffer()

叠加: intersect(), union(), difference()

f 空间对象变化操作

坐标点 S. Coord Point: moveto()

节点 S. Node: delete(), add()

弧段 S. Arc: split(), merge()

多边形 S. Polygon: split(), merge()

实体点 S. Entity Point: moveto(), appear(), disappear()

实体线 S. Entity Line: split(), merge(), appear(), disappear()

实体面 S. Entity Area: split(), merge(), appear(), disappear()

复杂实体 S. Entity Complex: separate(), composite(), appear(), disappear()

形状、大小、方向、结构是地理实体的空间几何特征;形状、大小、方向、结构的变化为地理实体表现出的几何特征变化;替换、传播、复制、生产、繁殖是在不同应用领域的各种具体地理事件的涵义。上面所列出的空间操作主要是有关空间对象数据结构变化的操作。

3.3 时间操作 T-T

a 时间对象成员数据获取/设置操作

日期 Date: getcurdate(), getyear(), getmonth(), getday(), dtoc(), securdate(), setyear(), setmonth(), setday(), ctod()

时间 Time: getcurtime(), gethour(), getminute(), getsecond(), ctot(), setcurtime(), sethour(), setminute(), setsecond(), ttoc()

时期时间 Datetime: getcurdatetime(), getdate(), getyear(), getmonth(), getday(), getcurtime(), gethour(), getminute(), getsecond(), setcurdatetime(), setdate(), setyear(), setmonth(), setday(), setcurtime(), sethour(), setminute(), setsecond(), dttoc(), ctodt()

时间点 T-Time Point: getdate(), gettime(), getdatetime(), setdate(), settime(), setdatetime(), temporalordering()

时间区间 T-Time Period: getstarttimepoint(), getendtimepoint(), setstarttimepoint(), setendtimepoint(), extendintervalto timepoints(), Shrinktimepointsto interval(), temporalordering()

时间元素 T-Time Element: getstarttimeinterval(), getendtimeinterval(), getstarttimepoint(), getendtimepoint(), setstarttimepoint(), setendtimepoint()

b 时间对象计算函数

时间区间 T-Time Period: duration()

时间元素 T-Time Element: duration()

c 时态数据查询操作

(见 T-A、T-S)

d 时间对象生成操作

同类型时间对象运算符: 加+ (或并)、积* (或交)、减- (或差-)

e 时态对象时态关系操作

时态拓扑关系操作: disjoint(), meet(), overlap(), cross in()

时态方向关系操作: last(), next(), before(), after()

时态距离关系操作: earlier(), later()

时态数据 (时态属性、时态空间) TA-TA、TS-TS

时态属性对象成员数据获取/设置操作: getvalobject(), getvalidtime(), setvalobject(), setvalidtime()。

固定时间 T, 由属性查询空间数据 A-S
属性条件 对象标识 空间对象 (同传统操作)。

固定时间 T, 由空间查询属性数据 S-A
空间条件 对象标识 属性对象 (同传统操作)。

固定空间 S/属性 A, 由时间查询属性/空间数据 T-A/T-S

时间点查询: 快照属性/空间数据;

时间段查询: 时间段正好在单个对象的有效期内, 相当快照属性/空间数据; 时间段包括几个对象的有效期, 则为快照系列;

时间关系查询: 查询满足给定时态关系对象的值。

固定空间 S/属性 A, 由属性/空间查询时间数据 A-T/S-T

对象值的有效时间查询, 不同对象的有效时间关系查询。

属性/空间变化, 由一属性/空间状态跟踪另一属性/空间状态 A-A/S-S

地理实体前一属性/空间状态 (值或相对关系) 查询, 地理实体后一属性/空间状态 (值或相对关系) 查询, 地理实体存亡事件中, 产生该地理实体的源地理实体查询, 该地理实体参与生成的结果地理实体查询。

如第二节所述, 各种二维矢量结构的空间对象类型数据结构均可借助时态对象代数的4个构

造子实现,同时,以上时空对象各种操作除空间计算、空间对象生成(缓冲、叠加)、空间关系、时态关系需依赖几何计算(以API实现)外,其余各操作均可通过基本时态对象代数操作组合完成。

4 结束语

本文指出了时空数据模型设计的基本指导思想,同时给出了若干常用时空数据类型及其数据结构的时态对象代数实现,列出了各种常用的时空对象操作。这些构成一个面向对象时空数据模型的基本框架。在实现一个通用的面向对象时空数据库管理系统时,数据类型、操作类型和完整性约束还需更系统和更深入的研究。

参 考 文 献

- 1 Langran G. Time in Geographic Information Systems Taylor & Francis Ltd, 1992
- 2 舒红,陈军. 地理数据模型中的时间语义. 见: 中国GIS协会第二届年会论文集, 1996
- 3 Peuquet D J, Duan N. An Event-Based Spatio-Temporal Data Model (ESTDM) for Temporal Analysis of Geographical Data. Int. J. Geographical Information Systems, 1995, 9 (1): 7~ 24
- 4 Worboys M F, Hearnshaw H M, Maguire D J. Object-oriented Data Modelling for Spatial Databases. Int. J. Geographical Information Systems, 1990, 4 (4): 369~ 383
- 5 Worboys M F. A Generic Model for Planar Geographical Object. Int. J. Geographical Information Systems, 1992, 6 (5): 353~ 372
- 6 Edwards G, Gagnon P, Bedard Y. Spatial-Temporal Topology and Causal Mechanisms in Time-Integrated GIS: From Conceptual Model to Implementation Strategies, 1992: 842~ 857
- 7 Cheng T, Wolfgang K, Robert A. Coupling GIS and Environmental Modelling: the Implications for Spatio-Temporal Data Modelling. ISPRS, Vienna, 1996 (B3): 849~ 856
- 8 杜道生,舒红. 基于同步数据项组和碎分拓扑弧段时间标记的时态地理数据模型. 武汉测绘科技大学学报, 1997, 22 (2): 96~ 101
- 9 Fegeas R G, Cascio J L, Lazar R A. An Overview of FIPS 173, The Spatial Data Transfer Standard. Cartography and Geographic Information System, 1992, 19 (5): 278~ 293

An Object-oriented Spatio-temporal Data Model

Shu Hong¹ Chen Jun² Du Daosheng¹ Zhou Yongqian¹

1. (National Laboratory for Information Engineering in Surveying, Mapping and Remote Sensing, W T U S M, 39 Luoyu Road, Wuhan, China, 430070)

2. (National Geometric Information Center, 1 Baisheng Village, Zizhuyuan, Beijing China, 100044)

Abstract In this paper, some basic ideas of designing a spatio-temporal data model have been presented from two aspects of storage management and description of spatio-temporal semantics. Meanwhile, spatio-temporal object types and its data structure, spatio-temporal operators about an object-oriented spatio-temporal data model has also been presented.

Key words spatio-temporal data model; spatio-temporal semantics; object-oriented; spatio-temporal composite; temporal object algebra