

超媒体数据库的一致化数据模型^{*}

吴中海 叶澄清 潘云鹤

(浙江大学计算机系 杭州 310027)

摘 要 This paper discusses the features of hypermedia database, analyses its modeling requirements, and offers an unified data model. This model provides several important mechanics, such as object composition, view specification, browsing support, media synchronization and so on.

关键词 Hypermedia database, Object-oriented, Composite node, Type expression.

1. 引 言

1965 年 Ted Nelson^[1]最早提出了超媒体的概念,他把非顺序的写作称为超文本。超媒体则是超文本的多媒体扩展,超媒体数据库(HMDB)是超媒体技术与数据库技术有机结合的产物,其研究对象主要包括超媒体数据模型和查询语言。

超媒体数据库模型的研究是目前数据库、多媒体、Petri 网等领域的一个研究热点,有很多文章从不同侧面阐述了超媒体数据库的建模思想。Koo^[2]在其模型中强调基于图形语法的结构说明;Stotta 和 Furtuta^[3]提出了基于 Petri 网的并发浏览机制;Klas 等^[4]给出了一个基于媒体类的面向对象模型;Horn 和 Stefani^[5]提出了多媒体同步语义的表示方法,并发表了 ESTEREL 查询语言;Golshani 和 Dimtrova^[6]等完成的 EVA 系统提供了语义丰富的多媒体运算。

本文采用面向对象的概念设计方法,根据超媒体数据库的特征,在充分吸收上述建模思想的基础上,提出了一个具有面向对象特征的一致化数据模型,该模型可以方便地支持多媒体数据的表示、组织、访问、管理。

2. 一致化建模要求

为便于描述,本文直接引用了面向对象数据库^[7]的一些基本概念,首先介绍超媒体数据库的一些基本特征,接着提出一致化建模的基本要求及其主要思想。

一般来说,超媒体数据库具有以下特征:①结点

(node)和链(link)是超媒体数据库中两个最基本的概念,结点表示多媒体对象,链表示它们之间的关系。②结点是一个对象,它属于某个对象类,对象类可以是其超类结构中的一部分;③链也是一个对象,有其相应的服务(Link Service);④通过链构成结点的信息互联网;⑤通过链浏览网中的各个结点;⑥通过对象标识(oid)建立索引;⑦通过类型约束进行相应的超媒体处理;⑧通过消息传递实现超媒体数据库查询。这些特点,对一致化建模提出了三个方面的要求:

1)静态特性建模。在超媒体数据库中,数据信息分为两类:一类是常规数据信息,是结构化的,如字符、整数、串等;另一类是多媒体数据信息,是非结构化的,如文本、声音、视频等。传统的数据库模型(如关系模型)只能处理结构化数据,对多媒体数据的处理只能通过多媒体扩展接口(如 MCI,MDK 等)。超媒体数据库模型必须能集成并处理各种类型的数据信息,不但应支持常规数据信息的处理,能对信息实体进行结构化描述,而且应提供类型检查机构支持多媒体数据信息的表示及其运算。本文在结点和链的概念基础上,通过引入线(Line)、组合结点(CompositeNode)、超集(Hyperset)等概念对象给出了超媒体数据信息的层次模型。同时本文引入了类型表达式概念,为类型检查机构提供了类型代数框架。

2)动态特性建模。在超媒体数据库中,引用和组合是两个重要的对象运算符,浏览和查询是信息检索和传递的重要手段,模型必须提供这方面的支持。本文在静态模型的基础上给出了超媒体数据库的 Panning 模型和 Zooming 模型。

^{*} 本文得到国家 863 基金资助。吴中海 博士生;叶澄清 教授;潘云鹤 教授、校长、博士生导师,主要研究领域:人工智能、CAD&CG、多媒体系统。

另外,模型还必须解决版本控制问题。与传统数据库一样,超媒体数据信息在数据库中也有其生命周期,因此对多媒体文档的历史信息建模也是一致化建模的重要任务。

3)约束建模。多媒体对象的运算有其各方面的约束条件,对于静态对象,包括类型约束和空间约束,对于动态对象还应包括时间约束。本文通过视图说明提供对象及其运算的空间约束模型。视图是数据库设计中的一个重要概念,体现了数据库多路访问的特征。本文在视图概念的基础上给出了更高层次的数据抽象——超集,超集的概念为对象的组合和聚集运算提供了结构约束。本文最后还给出了同步运算的时间约束表示方法。关于类型约束则在类型代数中阐述。

总之,超媒体数据库的建模应遵循以下五个原则:①一致性,包括类型一致性和结构一致性;②可扩充性,支持用户自定义的数据类型及其抽象;③可伸缩性,模型允许根据用户的要求进行结点的增删;④开放性,模型能方便地转换和映射到其他模型;⑤实用性,提供视图说明、浏览支持、媒体同步、版本管理等机制。

3. 面向对象的超媒体数据库概念模型

我们把超媒体数据库模型分为两部分:静态部分和动态部分。静态部分主要描述多媒体文档模式的层次结构及其类型框架,动态部分则包括多媒体信息对象的检索、运算和管理。面向对象的思想为静态建模和动态建模提供了简便有效的方法。

3.1 基本概念对象

从理论上讲,通过结点和链可以构成超媒体信息对象的任意网状关系网,但实际上由于网状模型结构过于复杂,检索速度缓慢,很难满足超媒体数据库的要求。为此,我们在结点和链两个基本概念对象的基础上进一步引入了另外三个概念对象:线对象(Line)、组合结点对象(CompositeNode)和超集对象(Hyperset)。

线对象是链的扩展,描述了对象结点之间的关系。它分为两种:信息线(InfoLine)和链接线(LinkLine),信息线对象包含了物理对象结点的对象标识。链接线对象即链,它包含了概念对象结点的对象标识。

为说明组合结点对象的概念,先对结点对象形式化定义如下:

$Node = \{ (^ Line | ^ Node | ^ CompositeNode) \}$

其中:“{ }”表示一个集合,“^ X”表示类型为 X 的实

例的指针或 oid,“|”表示选择操作。组合结点对象(CompositeNode)包含两部分:一张其组成单元的有序表,一个上下文相关的描述信息的集合。组合结点对象可以被归纳定义如下:

$CompositeNode = (mainNode; [^ Node], minorNode; ^ CompositeNode)$

“[]”表示一有序表,“^ ”表示一个元组。mainNode 是一张有序的结点对象表,构成控制视图;minorNode 用于存放一些辅助信息,这些信息在 mainNode 的上下文中得以解释。因此可以把辅助信息看作主导信息的一部分,一旦主导信息被删除,辅助信息也相应被删除。

在我们的模型中,结点对象和组合结点对象被认为是同一层次上的数据抽象,均是信息表示的概念单位。组合结点对象包含了结点对象所具有的全部特征,较好地体现了用户对信息的不同兴趣。另外组合结点对象还体现了信息间的同步语义,例如为一幅画面配解释词,显然画面是主导信息,解释词是辅助信息,一旦画面消失,解释词也相应消失。

超集对象是更高层次上的数据抽象,可以形式化定义如下:

$Hyperset = [(^ Node | ^ CompositeNode | ^ Hyperset)]$

利用上述概念对象很容易建立超媒体数据库的层次结构模型,如图 1 所示。

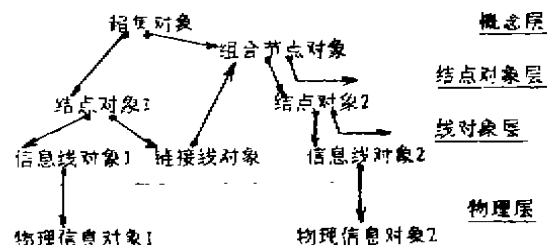


图 1

3.2 类型代数框架

超媒体数据库静态建模的另一个重要任务是为类型检查机构建立代数框架。与传统的数据库类似,超媒体数据库中的事务也可看成是由一些对象集合和运行于集合上的函数组成,换言之,数据库事务可看成是基于符号表的多类代数运算。符号表由四种符号组成:类型名、变量名、函数名和运算符。运算符和基本类型名由模型定义,变量名、函数名和复杂类型名由用户定义。

定义 1 简单类型表达式 STE.1)若 α 为一类型

名,则 α 为一简单类型表达式;2)如果 $\alpha_0, \dots, \alpha_n$ 均为STE,则 $\alpha_0 \cup \alpha_1, \alpha_0 \times \alpha_1 \times \dots \times \alpha_n, P(\alpha_0)$ 也是一个STE,其中“ $\cup$ ”表示并集运算,“ $\times$ ”表示迪卡尔积,“ P ”表示幂集运算(下同)。

定义2 函数类型表达式FTE(参元数为 n)。 $\forall i, (0 \leq i < n)$,如果 α_i, β 均是STE,那么 $\alpha_0, \dots, \alpha_i, \dots, \alpha_{n-1} \rightarrow \beta$ 是一个FTE。其中“ $\rightarrow$ ”表示蕴含。

定义3 类型标记是一个符号表上的函数,它给每一个函数名指派一函数类型表达式(FTE),给每一个变量名指派一类型表达式。

运算类型表达式OTE的定义与FTE相仿。基于上述定义,我们给出模型的类型代数框架。

定义4 给定一简单类型表达式 α ,和一类型标记 Σ 上的代数集 A ,记 A_α 为类型 α 的对象集(或称 α 类),则有:1)如果 α 是一类型名,则 A_α 是在代数 A 上的 α 类;2)如果 α 为 $\alpha_0 \cup \alpha_1$,则 $A_\alpha = A_{\alpha_0} \cup A_{\alpha_1}$;3)如果 α 为 $\alpha_0 \times \dots \times \alpha_i$ 则 $A_\alpha = A_{\alpha_0} \times \dots \times A_{\alpha_i}$;4)如果 α 为 $P(\alpha_0)$ 则 $A_\alpha = P(A_{\alpha_0})$ 。

定义5 给定一类型标记集 Σ ,记 E_α 为类型 α 的所有类型表达式集,则 E_α 可归纳定义如下:1)所有类型为 α 的变量名属于 E_α ;2)如果 η 是一函数名,其函数类型表达式为: $\alpha_0, \dots, \alpha_i, \dots, \alpha_{n-1} \rightarrow \beta (0 \leq i < n)$,且 $\forall i, e_i$ 属于 E_{α_i} ,那么 $\eta(e_0, \dots, e_i, \dots, e_{n-1})$ 属于 E_β ;3)如果 ζ 是一算符,其运算类型表达式为: $\alpha_0, \dots, \alpha_i, \dots, \alpha_{n-1} \rightarrow \beta (0 \leq i < n)$,那么给定 n 个表达式 $e_0, \dots, e_i, \dots, e_{n-1}$,且 e_i 属于 E_{α_i} ,则 $\zeta(e_0, \dots, e_i, \dots, e_{n-1})$ 属于 E_β 。

因此,类型代数框架可以认为是类型标记与类型表达式的总和,即 $\Sigma + E$ 。

3.3 视图说明

视图对象是指包含对象空间中一些类对象的对象结构。对象结构中的每个类对象被称作上下文,它携带了一个类对象的典型信息。一个线性的由顶向下的上下文结构称为透视,描述了操作和属性的线性继承关系。一组上下文可以表示一个或多个透视,它们被聚集到视图对象中。为了将信息和操作的访问权限传递给授权用户组,视图对象被连接到用户类对象。视图对象的创建遵守以下四条规则:

1)如果类 C_1 是类 C_2 的直接子类, C_2 是 C_3 的直接子类,那么 C_1 可以被概括为 C_3 的子类。

2)视图说明是指对视图对象按其结构的顺序进行赋值。

3)视图说明的继承范围由视图对象所引用的上下文决定。

4)视图说明受数据对象空间中的IS-A的结构

约束。类结构之间的IS-A关系必须通过模板结构(template)来表示。

模板结构是一个特殊的对象结构,它根据类对象的IS-A关系,聚集了对象空间类结构中的全体类。模板结构是创建视图对象的模板,可以由系统自动创建或更新。

视图说明可以通过消息传递来实现。如有一消息,make_view: (view_name) with: (list of class names)发送到对象ViewObject,将根据模板结构产生一个具体的视图对象实例。(list of class names)是一张线性表,它用来描述视图对象的透视结构。

例1 设有一视图对象V_Object其结构如图2所示:

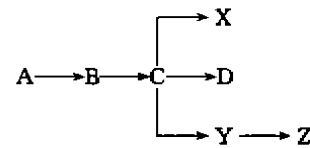


图2

那么V_Object可以由表[[A,B,C,D],[C,X],[C,Y,Z]]来描述。视图对象的结构方便地支持了对象的属性继承。例如,V_Object的可见用户允许访问上下文A,B,C,D,X,Y,Z的信息。并且一旦属于类Y的一个对象被访问,那么该对象将继承其超类A,B,C的全体属性。

3.4 浏览支持

超媒体数据库中的数据访问有两种方法:查询和浏览。简单的查询包括条件查询和标识查询,当用户在检索数据时,可能还没有确切的查询表达,这时就要采用浏览的方法。浏览包括横向浏览(panning)和纵向浏览(zooming)。横向浏览是指同层信息结构中的浏览,纵向浏览是指由顶向下的浏览操作。在我们的模型中,超媒体数据库的实体是按层次方式组织的对象集,层次结构中的根结点表示最一般的信息,层次越深,结点所包含的信息越具体。

1)Panning模型。我们提出对等对象、对等类的概念来描述panning模型。对等类(PeerClass)是一张处于同层信息结构中的对象标识表。对等对象(peer object)则是对等类的实例。

例2 在一本英汉词典的文档结构中,其对等对象一般包括:

```

[Description, Preface, Reference, List, Appendix]
[List-a, List-b, ..., List-z]
[Item-a1, ..., Item-am]
.....
[Item-z1, ..., Item-zn]
  
```

"List_a"表示以字母'a'开头的词汇表;"Item_a1"表示字母'a'开头的第一个词汇条目。词典的文档结构还可进一步细分,例如条目下的对等层可以包含其释义的文本描述、图形示范,还可以包括其同义、反义词的索引。

Panning 模型提供对等浏览命令 peer_browse。假设用户当前处于某一结点对象 P,peer_browse 命令将使用户访问与 P 处于同层信息结构中的下一个对象 Next(P);如果 P 是表中的最后一个对象,那么用户将访问表中的第一个结点对象。这样浏览可以一直进行下去,直至用户终止。

2)Zooming 模型。纵向浏览有两种方法,一种是信息导航,另一种是链导航。这两种操作均可通过结点对象的消息行为来建模。设有一消息模式 N zoom_at; (L),其中,N 是接收消息的结点对象,不妨说是一组合结点对象,L 是包含在 N 中的一个线对象。如果 L 是一个信息线对象,那么将根据其类型进行相应的信息导航;如果 L 是一个链接线对象,则将其传递给目标对象进行链导航。

查词典是一个典型的例子,在例2中,我们可以把文本描述和图形示范看作是条目对象的信息线,其索引和其他引用(如 see also)可看作是链接线,这样寻找一个单词的释义可采用信息导航,查阅其同义、反义词和其它相关的单词可通过链导航。

3.5 同步运算

我们将媒体同步归为两类:空间同步和时序同步。前者是指多个媒体对象在空间域上的同步,例如在某一时刻 τ ,在屏幕上同时出现几个画面,空间同步的对象只能是可见类型对象。时序同步是指多个媒体在时间域上的同步,例如,在时间域 $t_1 \sim t_2$,放影一剧本,该剧本包括一段视频序列 δ 及其配音 Ω 。时序同步运算有 simultaneous,before,meet,at 等。

组合结点对象可以很好地表示空间同步概念。我们知道 mainNode 是一张有序结点对象表,minorNode 则是上下文相关的辅助信息集合,是无序的。当访问一组合结点对象时,其包含的所有可见对象将同时被显示。例如,设有一组合结点对象 $w = \{oida, x = \{oidb, oide\}\}$,oida,oidb,oide 是线对象标识,x 是一结点对象。一旦 w 被访问到,则三个线对

象所包含的可见信息将被同时显示。

时序同步运算的语义可以用二阶谓词 $H(t, w)$ 归纳定义, $H(t, w)$ 表示对象 w 在时间 t 的状态,给定一时刻 τ 和一时序运算表达式 ρ , $h(\tau, \rho)$ 表示 ρ 在时刻 τ 的状态。

1)如果 ρ 为 w_1 at w_2 则有 $h(\tau, \rho) = H(w_2, w_1)$ $w_2 \geq \tau$ 。

2)如果 ρ 为 w_1 simultaneous w_2 则有 $h(\tau, \rho) = H(\tau, (w_1, w_2))$ 。

3)如果 ρ 为 w_1 before w_2 则有 $h(\tau, \rho) = \exists t_1 (t_1 < \tau) \wedge \forall t (t < t_1) (H(t, w_1) \wedge \text{not } H(t, w_2)) \wedge \forall t (t > t_1) (\text{not } H(t, w_1) \wedge H(t, w_2))$ 。

4)如果 ρ 为 w_1 meet w_2 则有 $h(\tau, \rho) = \exists t_1 (t_1 > \tau) \wedge \forall t (t < t_1) (H(t, w_1) \wedge \text{not } H(t, w_2)) \wedge \forall t (t \geq t_1) (\text{not } H(t, w_1) \wedge H(t, w_2))$ 。

关于媒体同步的详细论述可参考文[5]。

参 考 文 献

- [1] T. Nelson, Replacing the printed word, A complete literary system, in Lavington, IFIP Congress 1980 North-Holland, 1980
- [2] R. Koo, A Model for Electronic Documents, SIGOIS Bulletin, Vol. 10 No. 1, 1989
- [3] P. D. Stotts 等, Petri-Net-Based Hypertext: Document structure with browsing semantics, ACM trans. on information systems, Vol. 7 No. 1, 1989
- [4] W. Klas, Using an object-oriented approach to model multimedia data, Computer communication, Vol. 13 No. 4, 1990
- [5] F. Horn, 等, On programming and supporting multimedia object synchronization, The computer journal, Vol. 36 No. 1, 1993
- [6] F. Golshani 等, Design and specification of EVA: a language for multimedia database systems, Proc. of the Int. Conf. on Database and Expert Systems Application, 1992
- [7] Alfons Kemper 等, Object-oriented database management, Published by Prentice Hall Inc. 1994