

超媒体著作系统 OOHAS 中的对象设计

韩波 (中南财经大学信息系 430064)

张建昌 (中科院软件所 100080)

韩光鹏 (武汉大学计算机科学系 430072)

摘要:OOHAS是结合面向对象技术设计并实现的一个超媒体著作系统。本文详细介绍了系统中所定义的各种不同层次的对象操作,并分析了由这些对象构造超媒体文档的能力和特点。

关键词:超媒体 面向对象 复合结点

超媒体是一种将多媒体信息组织成非线性网状结构的信息管理技术。在已有的超媒体系统中,编著者先将收集的各类信息按内部的逻辑联系划分成具有不同表现主题的结点,然后用链将它们连接成为网状结构——超网,以支持信息的“联想式”浏览。近年来,随着多媒体表现技术的不断成熟和社会需求的不断增长,超网中所包含的多媒体信息种类越来越多,且关系复杂、处理繁琐。因此,已有的许多超媒体系统已不适于开发大型超媒体文档的要求。

面向对象技术以易于人们所理想和接受的思维方式,将客观世界或被求解的问题化分为一些“对象”以及对象的组合和联系,利用“对象”这个概念将数据结构及作用其上的操作封装在一起,支持作为可独立存在的建模实体,通过消息传递来实现对象之间的联系,若干对象通过消息传递可以有机的组成一个系统。因此,面向对象技术所具有的封装性、继承性、可扩充性等特征能为大型超媒体文档管理提供重要的支持作用。

在我们设计并实现的超媒体著作系统 OOHAS(Object-oriented hypermedia Authoring System)中,按照面向对象的观点和方法,将超媒体信息划分成为不同的各类对象,从而使超媒体文档的创建、编著从简单的无语义的链接过程发展到面向对象的造型组合过程。

本文第一节详细介绍 OOHAS 系统中所定义的各种对象,第二节分析这些对象的操作及其语义约束,第三节讨论由这些对象构造超媒体文档的能力。

1. 对象类型

在 OOHAS 系统中,超媒体文档由复合结点、原子结点、控制结点、媒体结点、信息对象、时间同步对象、锚定

对象等组成,下面分别加以说明:

(1) 信息对象。包含媒体数据和相应的解码和表现信息。可以通过国际上现在通常采用的“建议”或标准来对这种对象进行编、解码。信息对象的编、解码信息位于对象头部的“Hook”中,它通常包括媒体的类型标识(“Wav”,“Bmp”)、媒体固有的表现时间或空间范围及其他一些参数设置(如表现位置、表现空间、阴影方式、光栅操作类型、调色板、正常播放速度)等。

OOHAS 中支持的信息对象类型有:文本、图形、图象、音乐、语音、动画、视频、外部程序等。

(2) 时间同步对象。它确定了一个信息对象在开始演播时需满足的条件,从而使原子结点中各媒体的演播达到时间同步的效果。

(3) 锚定对象。锚定是指媒体结点存在的逻辑空间内引起信息变迁的区域。它由锚定类型、标识符、锚定区、变迁目的结点组成。标识符是相对于某一个媒体结点安排的,在系统中,由原子结点、媒体结点和锚定标识符才能唯一地确定一个锚定。超媒体系统中的非线性链接正是通过这个对象实现的。

(4) 媒体结点。它是由信息对象、时间同步对象和锚定对象组成。这种组成使得对媒体结点的管理更方便,并有可能让多个相似的媒体结点共享相同的信息对象。

(5) 控制结点。现有的超媒体系统中实体通常代表信息载体的概念,即不含任何信息的实体就不能称之为结点。这种限制无法满足处理复杂情况时文档需求。本系统中,我们采用控制结点来克服这个弱点。它规定有两方面作用:①存在性约束:用于决定某种类型结点的存

在性,以保持浏览的完整性和语义完整性。②语义连接:控制结点可以作为两个结点之间的中间语义连接。

(6) 原子结点。它是超媒体文档的基本组成单元,对应于其他系统中页、窗口或屏的概念。其组成为:原子结点::=控制结点||媒体结点|

(7) 复合结点

. 复合结构类。复合结构类是用于锚定超媒体文档中不同结点之间的抽象数据类型。复合结构类中定义的结点结构包括顺序、树型、任意连接和‘孤立’结点(参见文献5)。所谓‘孤立’结点是指不与复合结构中的任何其他结点发生关联的结点,它是复合结点中的特例,对于这种结点读者是无法浏览的,但编辑者可以随时通过有关操作使之与复合结点内的其他结点相连。

. 复合结点类。复合结点类是复合结构类的子类。它将复合结构类中结点间定义的结构联系赋予一定的类型。

. 复合结点。复合结点是复合结点的实例,即为满足复合结点类中定义的结构和类型连接的复合结点对象。超媒体文档可看作是一个根据预定义的复合结构类库或复合结构类库所生成的实例——复合结点构成的集合。图1给出了一个复合结点的示例。

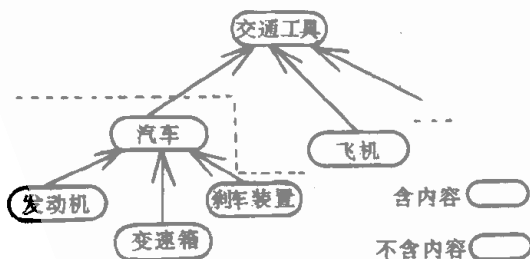


图1 复合结点“交通工具”示例

在OOHAS中,复合结点具有下列性质:

. 可嵌套性:即构成复合结点依旧可以是复合结点。如图1所示,‘交通工具’是一个复合结点,构成它的‘汽车’也可以是一个复合结点。

. 复合结点中允许控制结点存在。我们还是以图1中的‘飞机’结点是控制结点,也就是说‘飞机’结点在这里起到了结构和语义两个方面的作用,但它并不是用户的兴趣所在,但是如果将这个结点去掉,则复合结点‘交通工具’将不具备完整的涵义。

. 封装性:如图1所示‘交通工具’是由‘汽车’和‘飞机’等聚合而成的复合结点,它对应于一个封装的类。对于用户而言,汽车的内部结构在‘交通工具’结点这一层是不可见的,用户只能通过交通工具类提供的接口来控制对汽车对象的操作,如浏览、修改等,而‘汽车’也是一个复合结点,它由发动机、变速箱、刹车装置三个子结点构成,至于发动机是否又作为一个复合结点则视表现主题的需要而定。

. 共享性:结点X可以属于一个或多个复合结点,这并不意味着结点X会被物理地存储多次,而只是用一个指针指向结点X。对于用户来说这一切是不可见的。

2. 对象操作

在OOHAS系统中,对象的操作主要分为四种:浏览、创建、删除和修改。以下将分别介绍它们的运行规则和语义约束。

(1) 浏览(BROWSE)。浏览是超媒体系统中所必须的操作,对于大型的超媒体信息系统则更是如此。由于大量的信息及信息间的非线性联系,读者往往需要知道当前结点在整个系统中的位置及与其他结点的联结关系等。因此,浏览功能除显示信息内容外,还必须提供导航机制。

从系统的观点来看,读者对复合结点的一次遍历过程就是围绕某一主题的超媒体信息浏览过程。

(2) 创建(CREAT)。创建一个新原子结点,即向系统插入一个原始结点,它既可以是媒体结点也可以是控制结点。插入时必须首先确定它所属的复合结构类,同时提示与之相关联的父结点或兄结点,便于用户决定是否还要进行下一步创建工作。

(3) 删除(DELETE)。由于OOHAS系统中定义了复合结点,所以删除操作将不仅是结点的简单删除,它还将影响到许多与之相关的结点,以至整个超网结构的变化。因此,在删除的同时还应该定义一套完整性约束,来确保删除操作能保持系统在语义上的完整性,系统中定义了以下几种删除规则:

① 删除复合结点内的原子结点,从理论上说,所有与之关联的结点都应该被删除,但是一个结点很可能为多个复合结点所共享,从实现的角度来说,搜索所有的相关结点会影响操作效果。因此,我们将被删除结点转变为控制结点,这样既保持了完整性,又节省了时间。

② 如果删除复合结点,除了将它本身变为控制结点外,还将删除它的所有子结点的内容,并把所有子结点也

变为控制结点。

③ 如果删除复合结点内的控制结点,那么删除它的所有子结点。

(4)修改(CHANGE)。修改操作也可能导致网络结构的改变,因此同样应给予一定的语义约束。OOHAS系统中支持如下两种修改操作:

① 修改媒体结点:若修改结点的内容,那么系统状态不变,因为修改后结点与旧结点保持同一个对象标识。

② 修改控制结点:由于结构控制结点不含内容,对它实行的唯一修改就是属性修改,而不改变其他关系。

3. 对象的表示能力

OOHAS系统中提供了一整套对象:复合结点、原子结点、控制结点、媒体结点等,编著者负责将这些对象进行合适的组合以正确地各种超媒体文档设计。在这种意义下,OOHAS系统是一个创建、处理、显示嵌入超网结构中的各类对象的工具。

系统用复合结构类、复合结点类的实例——复合结点来描述具有同一表现主题的结点及它们之间的关系。这样利于编著者采用结构化的方法和造型的观点将各种主题的多媒体信息组织在一起,而不是仅仅局限于底层链接的观点;也使得读者能根据复合结点的逻辑语义来确定自己的当前,从而大大减少了“迷路”机会。

同时,时间同步对象的引入可使编著者定义原子结点内各媒体对象间较为复杂的时序关系,从而能满足多

种动态媒体和静态媒体间的协调表现要求。

因此,我们定义的各种不同层次的对象具有较强的表示能力,可以较好地实现大型超媒体文档的开发和维护。

4. 结束语

综上所述,OOHAS系统中的各种对象为超媒体文档的组织和管理提供了良好的支持。实现应用表明,它们简化了超媒体的网络结构,增强了导航时结点的描述能力,并在一定程度上克服了现有超媒体系统中一般只考虑文档的逻辑结构和布局结构,而未充分利用文档语义内容的缺陷。另外,它们也使得系统良好的扩充性,并有效地减少了信息的冗余存储。

参考文献

- [1] Wang, H K and Wu, Jean-Line C 'Object model for hypermedia applications' Computer Communications, 1995, 18(7): 475 - 485
- [2] Conklin E J 'Hypertext: an introduction and survey' IEEE Computer, 1987, 20(9): 17 - 41
- [3] Isakowitz, T., Stohr, E A and Balasubramanian P. 'RMM: A Methodology for Structured Hypermedia Design' Communications of the ACM, 1995, 38(8): 34 - 44

(来稿时间:1997年10月)