

一个多媒体数据同步模型的建立和实现^{*}

黄波 何志均

(浙江大学CAD&CG国家重点实验室 杭州 310027)

(浙江大学人工智能研究所 杭州 310027)

摘要 文章对多媒体数据流间存在的时序问题和时序关系进行介绍,对现有研究情况进行分析.结合一个多媒体数据模型,基于PETRI网,提出了一个同步模型,以表达媒体数据之间存在的同步关系.根据这两个模型对多媒体数据建立了相应的数据结构,并采用该数据结构介绍了对多媒体数据存取播放的实现算法.

关键词 多媒体,时序问题,时序关系,PETRI网,同步模型,数据存取.

中图法分类号 TP391

多媒体信息可分成静态信息和动态信息两类,静态信息如文本、图形和图象;动态数据如动画、声频、动态视频数据和MIDI音乐,它们均是与时间相关的数据流.一般的多媒体应用系统^[1,2]中的一个节点实例本身可能包含多个媒体数据.由于动态数据的引入,不可避免地要考虑各个媒体数据在时间上的相互依赖关系,如,对一段录像进行记录时,不仅需要记录其中一系列的视音频数据,同时还需要考虑两者如图1所示的时序关系,即对应的音频数据和视频数据间的同步关系;另一方面,根据系统的具体需求,也需要对多媒体信息库中的媒体数据定义相互间的时序关系,以便依次呈现在用户界面上.例如,在一个餐饮服务系统中,有一项介绍菜肴的服务,对每一道菜的介绍内容包括有关该道菜的图片以及对该菜的文字介绍和语音讲解.在时序上要求先同时显示图片和文字,再进行播音,形成图2所示的时序关系.

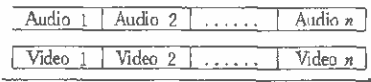


图1 一段录像

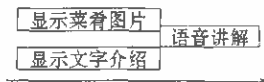


图2 菜肴介绍中的同步问题

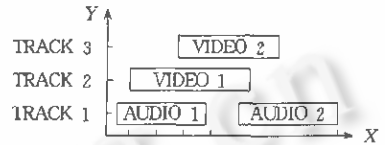


图3

因此,对于多媒体系统中的时间关系,不仅应当从用户界面的角度,即从多媒体演播的角度考虑,对媒体数据之间进行时间关系的定义,另一方面也需要对节点内多媒体数据之间本身存在的时序关系进行管理维护,以便在用户存取节点时,可以按媒体数据间原有的时间依赖关系呈现在用户的面前.当前对多媒体数据时序同步的研究和实现主要着重从用户界面方面进行多媒体数据的演播,一般采用类似图3的表达方法.其中X轴表示逻辑时间,Y轴表示对不同类别的媒体数据分配的数据流道.表达时,将各个媒体数据依次安排在该坐标空间中,数据之间的时序关系通过它们在时间轴上的起始时间偏移量来反映.在实现上,利用现有的操作系统,对每一个数据流道分配一个线程,采用多个线程的方式实现.这种表达方法比较直观,易于理解,但表达结果较为粗糙,存在下面的问题.

- ① 这种表达的结果使得媒体之间的时序关系较为固定,当媒体数据发生变化时,不利于对它们之间的时间同步关系的维护;
- ② 对数据演播时,数据通道过早分配,在实际演播时可能出现偏差;
- ③ 对媒体数据呈现时的持续时间的表达,是根据它本身的持续时间,还是通过时间轴进行分配,如何进行标志区别;
- ④ 当存在多个媒体数据流时,整个时序结构显得复杂化,即不利于数据间在时序上保持一致性,也不利于实现.

* 本文研究得到浙江省自然科学基金资助.作者黄波,1970年生,博士,主要研究领域为超媒体,软件工程,人工智能.何志均,1924年生,教授,博士生导师,主要研究领域为人工智能,专家系统,CIMS,超媒体.

本文通讯联系人:何志均,杭州 310027,浙江大学CAD&CG国家重点实验室

本文1997-04-14收到原稿,1997-06-23收到修改稿

⑤ 对于静态数据和动态数据相结合的情况,如何处理?

此外,国外对多媒体数据流间的同步表达模型进行了一系列的研究^[3-5],有代表性的是 THOMAS^[3]根据 PETRI 网提出的 OCPN 同步模型.该模型基于传统的时序关系来表达多媒体数据间的时序关系,解决了上面的问题②.对问题①③⑤的解释则较为模糊.尤其是,它在表达上主要是基于两个媒体数据流,强调对两个数据流在时间(T_a, T_b 和 T_c ,见第1节)上的数学依赖关系,在实现上采用二叉树的结构对多媒体数据进行存储播放,因此,在对多个数据流的同步关系进行表达管理时会引入大量的中间结构.例如,对 N 个数据流,可能会引入 $N-1$ 个中间表达结构,形成大量的数据冗余,造成存储空间的浪费;同时在多媒体数据进行演播时,会花费大量的时间在数据流调度和进程管理上,从而可能引起数据的抖动.本文的研究工作吸收了现有多媒体同步控制方面的研究成果,在建立统一的多媒体数据模型的基础上,提出了一个多媒体数据的同步模型(记为 TPETRI),侧重于对节点内媒体数据之间的时序关系进行管理维护,并提出相应的算法来实现对多媒体数据的存取和演播.

1 时序关系

在传统的实时系统和并行语言研究中,两个事件之间的时序关系总共可分为 13 种关系.由于这些关系之间存在的互逆性,因此可将这 13 种关系归并为 7 种时序关系,如图 4 所示,其中 T_a, T_b 分别表示事件 A, B 所分配的时间段长度, $T_c (T_c > 0)$ 表示两者在时间上相差的时间段长度.

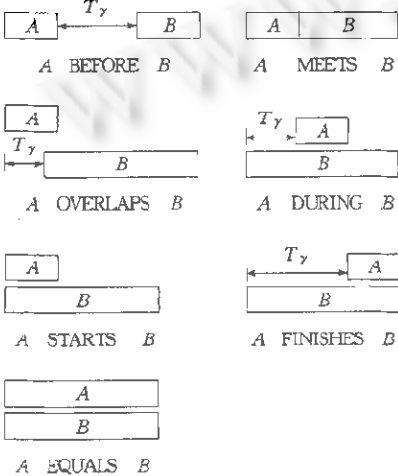


图4 同步关系

与上述两个事件间的时序关系有所不同,多媒体系统中的时序关系往往描述多个媒体数据流在呈现播放时的同步关系,并且媒体数据本身具有时间特性,若非特别对 T_a, T_b 进行规定分配,对于动态的媒体数据,往往是指所涉及媒体数据本身播放时所需要的时间;对于静态数据,则认为它们是瞬时的或由实际演播所需的时间确定.另外,对于媒体数据间的时序关系,往往是以一个数据流作为基准进行描述,因此,两个媒体数据流之间存在着 16 种时序关系,可以合并为 8 种关系.下面是对两个媒体数据流 A 与 B 之间时序关系的描述,其中 T_a, T_b 分别表示数据流 A, B 播放所需要的时间段.

(1) BEFORE(超前关系):表示一种串行关系, A BEFORE B 意味着先播放数据流 A ,然后在 A 播放完 T_a 段时间后开始对数据流 B 进行播放.

(2) MEET(汇合关系): A MEETS B 将 A 表示两者之间存在着顺序关系,数据流 B 紧接着数据流 A 播放.

(3) OVERLAP(覆盖关系): A OVERLAP B 表示两者存在着并行关系,先播放数据流 A ,然后在播放数据流 A T_c 时间段后,同时开始播放数据流 B ,其中 $T_c < T_a < (T_a + T_b)$.

(4) DURING(期间关系): A DURING B 表示两个数据流存在并行关系, A 在 B 发生的过程中发生,先播放数据流 A ,经过 T_c 时间段后,同时播放数据流 B ,其中 $T_a < T_b$.

(5) STARTS(开始关系): A STARTS B 将数据流 A 的起始点作为同步点,表示 A 与 B 同时开始播放.

(6) FINISH1(结束关系 1): A FINISH1 B 将数据流 A 的结束点作为同步点,表示在 A 结束时同时结束 B 的播放,其中 $T_b = (T_a + T_c)$.

(7) FINISH2(结束关系 2):与 FINISH1 不同之处在于 $T_c = (T_b + T_a)$.

(8) EQUERAL(同时关系): A EQUERAL B 将数据流 A 的起始点和结束点作为与数据流 B 的同步点,表示 B 与 A 同时开始,同时结束.其中 A EQUERAL B 与 B EQUERAL A 并不相同,后者是以数据流 B 作为基准描述的.

针对上述描述,下面的同步模型将对多个媒体数据流之间的时序关系进行表达.其中,媒体数据可以是动态数据,也可以是静态数据.

2 同步模型

本节首先对我们建立的多媒体数据模型和 PETRI 网进行基本介绍,然后对 PETRI 网进行修改、定义,以形成多媒体数据间的同步模型.

2.1 多媒体数据模型

本数据模型是针对当前多媒体系统数据和软件重用性低、不同系统间不易交流等问题而建立的,独立于某一应用领域的多媒体数据模型。该模型主要通过节点、链和锚3个基本元素进行定义。

定义1. $MMDB=(N,A,M,\lambda,\Gamma)$ 构成一个多媒体信息库,其中

- (1) N 表示节点(Node)集合。节点是多媒体系统中的基本信息单元,它通过 NODEID 进行标识;
- (2) A 是锚(Anchor)的集合,锚在该模型中是作为一种中间结构出现的,表示链的端点。一个锚可以是一个节点,也可以是节点中的某一信息片段,它通过 ANCHORID 进行标识;
- (3) M 是媒体数据集合,媒体数据可以是文本、图象、声音、动画、MIDI、视频等信息。对于同一类的媒体数据,可以采用面向对象的方法进行定义,它的表达包括媒体原始数据、媒体的有关属性和关于该媒体数据的使用方法。
- (4) $\lambda \subseteq (A \times A)$ 在语义上表达了节点之间的各种联系,构成一般多媒体系统中的链(link)。它通过 LINKID 进行标识。
- (5) $\Gamma \subseteq (N \times M)$ 表示节点集合与媒体数据集合间多对多的关系。一个节点中可以包括多个媒体数据,同一个媒体数据可以被多个节点所引用。

由于媒体数据间的时序关系主要体现在节点内或不同节点内媒体数据的时序关系,下面仅对节点进行详细介绍。

节点包含了多媒体系统中基本的信息内容。节点中一般包含着两类信息,一类是与媒体数据有关的信息,另一类是与应用领域有关的信息,并且两类信息密切相关。节点应当不论其大小,数据内容的多少,而从与领域无关及媒体无关的角度出发,进行形式化的定义,同时考虑它的扩充性。我们通过节点实例对节点的结构进行描述。

定义2. $NODE_INSTANCE=(ID,media_objects,features,\varphi,\gamma,methods)$ 是一个节点实例,其中

- (1) ID 是对该节点实例的唯一标识符;
- (2) $media_objects \subseteq M$ 是根据 Γ 定义的该节点实例中所包含的媒体数据;
- (3) $features$ 表示该节点实例的属性列,这些属性根据具体的应用领域而定。在这些属性中,有些属性是与媒体数据无关的,而有些属性则是与媒体数据相关的;
- (4) $\varphi: media_objects \rightarrow feature$ 表示节点实例中的媒体数据与媒体有关的领域属性之间存在的映射关系;
- (5) $\gamma \subseteq (media_objects \times media_objects)$ 表示一个节点实例中包含着的多个媒体数据之间的关系,比如,不同媒体数据之间的时序关系和空间关系就是比较常见的关系;
- (6) $methods$ 表示有关节点的处理方法,根据具体的应用而定。

采用上面的五元组,可以概括一个节点包含的信息,形成较为通用的节点定义,同时也考虑到了节点所涉及的媒体信息与领域特征。在实际的应用中,可以参考该模型进行具体的定义。

2.2 PETRI 网

首先对 PETRI 网进行一个简单的介绍,一个基本的 PETRI 定义如下。

定义3. 三元组 $N=(P,T,A)$ 称为一个网,如果

- (1) $P \cup T \neq \emptyset$ 且 $P \cap T = \emptyset$;
- (2) $A \subseteq (P \times T) \cup (T \times P)$;
- (3) $dom(A) \cup ran(A) = P \cup T$ 。

其中 T, P 和 A 分别表示了变迁集合、位置集合和有向弧集合,并且

$$dom(A) = \{x \in P \cup T \mid \exists y \in P \cup T, (x, y) \in A\},$$

$$ran(A) = \{x \in P \cup T \mid \exists y \in P \cup T, (y, x) \in A\}.$$

根据上述定义,一个基本 PETRI 网可以采用有向图进行表示。图中变迁 T 采用竖线表示,位置 P 采用椭圆表示,图5就是一个例示的 PETRI 网。对基本的 PETRI 网进一步加以定义,可用于并行程序设计和进程描述,详见文献[6]。

2.3 同步模型 TPETRI

以往对 PETRI 网的研究主要用于并发程序设计以及进程描述。在处理时序问题时,对 PETRI 网所作的扩充主要表现在认为各个变迁的激活具有时间属性。借鉴这些研究成果,考虑到实际情况,即提出本模型是为了便于表达多媒体数据之间的同步关系,因此,我们首先假设各个变迁是瞬时激活的,而认为各个位置是具有时间属性的,并吸收了现有多媒体同步控制方面的研究成果,对上面标准的

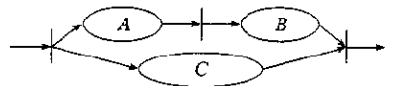


图5 基本PETRI网例子

PETRI网进行扩展定义,建立了统一的同步模型 TPETRI网.其定义如下.

定义 4. 七元组 $N=(P, T, A, Y, \tau, \omega, \Psi, \eta)$ 称为一个 TPETRI 网, 如果它满足下述条件:

- (1) (P, T, A) 是一个基本的 PETRI 网;
- (2) $Y = \{r_1, r_2, \dots, r_n\}$ 表示资源的集合;
- (3) $\forall p \in P, |^*p| = |p^*| = 1, ^*p, p^*$ 表示位置 p 的前端变迁和后端变迁;
- (4) $\tau: P \rightarrow Y$, 该映射使每一个位置均与一个资源相对应;
- (5) $\omega: P \rightarrow R_+$, 对 P 中的资源分配具体的时间段或者采用缺省时间段, 使 P 具有时间特征, 表示对 P 中资源播放的持续时间, 其中 R_+ 表示正实数集;
- (6) Ψ : 是对 P 的标记, 它是一个单值映射: $P \rightarrow \{1, 0\}$;
- (7) $\eta: T \rightarrow P$ 表示对一个变迁赋予的同步标记, 在本文中, 该同步标记由该变迁的一个输入位置表示.

在 TPETRI 中, 对一个位置分配标记, 表示对该位置对应的资源加锁, 以防止互斥访问. 当处理结束时, 再对该资源解锁. 因此, 该标记在下文中又称为锁定标记. 同步标记主要是出于对时序关系中参考数据流的考虑, 它的值作为基准的数据流所对应的位置. 在采用图形对 TPETRI 描述时, T 采用竖棒表示. 实体的竖棒表示该变迁具备同步标记, 并在右上方注明该同步标记的值.

同步模型 TPETRI 具有以下激活原则.

① 对于一个变迁, 如果它包含同步标记, 当对该标记所对应的输入位置的处理过程结束时, 则撤销所有输入位置中的锁定标记, 并立即激活该变迁;

② 对于一个变迁, 如果它不包含同步标记, 则当该变迁所有输入位置的处理结束时, 撤销这些输入位置的锁定标记, 该变迁立即被激活;

③ 一个变迁一旦被激活, 则对该变迁所有的输出位置分配锁定标记, 并同时开始对它们的处理过程.

采用该同步模型, 我们将用于表达不同媒体数据之间的时序关系, 首先有下面的定理.

定理. 对于任何两个媒体数据之间的时序关系, 都可以采用上面的同步模型 TPETRI 网进行表达.

证明: 由于任何两个媒体数据流之间存在着“时序关系”一节所述的 8 种关系. 首先我们引入一个不包含任何资源的空位置 P_r . 它的处理过程是进行空循环, 持续时间为 T_r . 这样, 对这 8 种时序关系, 都可以采用我们提出的同步模型 TPETRI 来表达, 图 6 所示是对上面 8 种关系的表达结果. 在图中, P_a 对应的资源为 A , 持续时间为 T_a ; P_b 对应的资源为 B , 持续时间为 T_b .

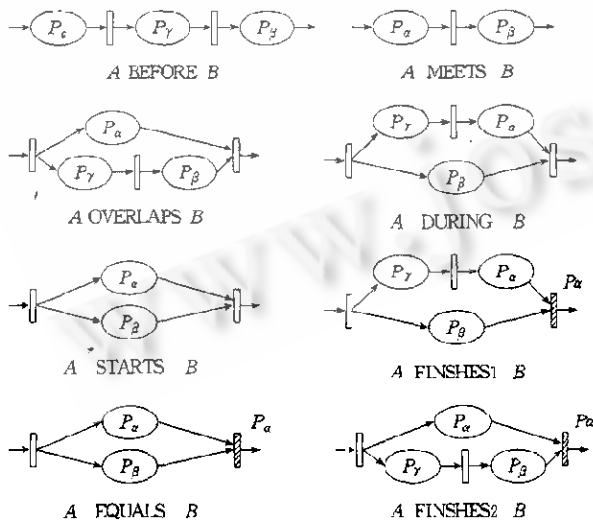


图6 时序关系的TPETRI表达

由于 TPETRI 中存在着下述原则: 当一个变迁一旦被激活, 则立即同时开始对该变迁所有的输出位置的处理过程, 因此, 在模型表达中对变迁中的同步标记采用该变迁的输入位置进行表示, 以考虑存在于数据流结束点的同步问题, 而不需要考虑存在于一个数据流起始点处的同步情况. 通过引入同步标记, 可以保证, 采用 TPETRI, 对不同的时序关系将得到不同的表达结果. 在 OCPN 模型中所表达的时序关系仍然是基于两个事件之间的时序关系, 期间、覆盖与结束 3 个关系的表达容易混淆, 如, 对期间与结束两个关系的表达是一样的. 而在本模型中则不同, 并且对于同一种时序关系, 存在且唯一存在一个表达结果, 反之亦然.

根据上面的定理, 我们可以得到以下推论.

推论. 对于任意多个媒体数据之间的时序关系, 同样可以采用上面的模型进行表达.

证明: 因为我们可以将任何两个媒体数据之间建立的 TPETRI 表达作为一个大的同步表达 TPETRI 网中的位置对待, 而这个位置的时间特征可以通过它的两个媒体子数据 A, B 唯一确定, 表 1 是根据图 6 所表达各种时序关系导出的该关系总的时间段. 因此, 我们可以首先考虑

两个媒体数据间时序关系的表达,然后通过不断引入一个其他的媒体数据来完成整个时序关系的表达.

表 1

<i>A before B</i>	<i>A meets B</i>	<i>A overlaps B</i>	<i>A during B</i>	<i>A starts B</i>	<i>A finishes1 B</i>	<i>A finishes2 B</i>	<i>A equals B</i>
$T_a + T_\beta + T_\gamma$	$T_a + T_\beta$	$T_\beta + T_\gamma$	T_β	$\max(T_a, T_\beta)$	T_β	T_a	T_a

采用该同步模型,我们可以对文章开始时的两个例子即一段录像和菜肴解说建立图 7 和图 8 的表达.在录像片断的同步表达中, $P_{audio\ n}$ 和 $P_{video\ n}$ 对应的数据分别是 audio n 和 video n ,在菜肴介绍的表达中, P_b, P_t, P_a 对应的数据分别是菜肴图片、文字解说和声音讲解.

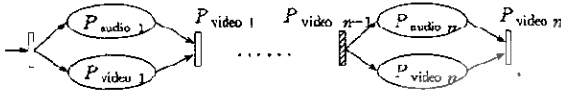


图7 录像片断的同步表达

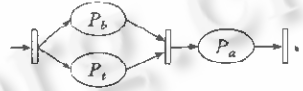


图8 菜肴介绍的同步表达

3 实现策略

根据上面提出的多媒体数据模型和同步模型,我们对多媒体系统中的同步关系的表达及数据存取播放进行了具体的实现.下面介绍根据这些结构进行多媒体数据存取播放的算法实现.

3.1 数据结构

下面是我们基于多媒体数据模型和同步模型建立的基本数据结构和索引结构.

```

CLASS node{
  STRING id;
  STRING nodetype;
  NODEREP * rep; /* point to the internal structure of a node */
  MEDIAOBJLIST * medialist; /* media object list */
  TRANSITION * transbegin; /* the synchronization representation that this node include */
  .....other methods.....
}
CLASS mediaobject{
  STRING id;
  STRING mediatype; /* media type such as "video" */
  MEDIAOBJREP * rep; /* point to the internal structure of a media object */
  NODELIST * nodefirst; /* node list */
  .....other methods.....
}
STRUCT transition{
  STRING id;
  PLACELIST * in; /* the input places of this transition */
  PLACELIST * out; /* the output places of this transition */
  PLACE * syntag; /* the synchronization tag of this transition which refer to a inplace */
}
STRUCT placelist{
  STRING id;
  PLACE * placeptr; /* refer to a place */
  TRANSITION * transptr; /* the transition that this place arrive after the media object plays */
  PLACELIST * next;
}
CLASS place{
  STRING id;
  MEDIAOBJECT * mediaobjptr; /* the used media object resouce */
  DURING interval; /* the duration that the media object of this place plays, it is represented by a begin time and end time */
  int lock; /* if the media object is used */
  VOID play();
}

```

```

VOID stop(),
}
STRUCT mediaobjlist{
STRING id; /* the id of a media object which is refered in this node */
MEDIAOBJECT * referedmediaobject; /* point to the refered media object */
MEDIAOBJLIST * next;
}
STRUCT nodelist{
STRING id; /* the id of refered node */
NODE * referednode; /* point to the refered node */
NODELIST * next;
}

```

在上面的定义中,节点结构的定义独立于具体的应用环境,在结构中涉及节点的标识符、节点的类型、通过 `noderep` 指向该节点的具体内容、节点的处理方法,此外还包括两个指针,分别指向该节点中所引用的媒体数据对象链的第1个对象和由节点中媒体数据之间时序关系形成的 PETRI 网表达的第1个变迁。

媒体对象结构独立于媒体数据的具体类型结构和存储格式,在结构中包括该媒体对象的标识符、媒体类型、通过 `mediaobjectrep` 指向该对象的具体内容结构、通过 `nodelist` 指向引用该媒体对象的所有节点链以及媒体数据的处理方法。

同步模型 TPETRI 的表达通过变迁、位置链、位置3个数据结构实现,一个变迁通过一个唯一的标识符进行标识,其中记录了该变迁中所有的输入位置链、输出位置链。如果该变迁存在着同步标记,则通过采用 `syntag` 指向该变迁的一个输入位置表示;否则, `syntag` 指向空值。如果该变迁是一个同步表达的最初变迁,则 `in` 指向空值;如果该变迁是最后一个变迁,则 `out` 指向空值。同样,如上节所述,我们引入一个空位置,该位置不引用任何媒体数据,仅是空闲一段时间而已。

位置链表示一个变迁所指向的所有输入或输出位置所形成的链表。

位置是对时序数据类的表达,它存在着一个唯一的标识符,指向具体的媒体数据、播放的时间段、锁标记。对于一个位置,它可以指向一个静态数据,如文本、图片,也可以指向动态的数据,如声音、视频数据流,媒体数据播放的时间段一般采用缺省值,即将该数据全部呈现所需的时间,也可以由用户自己定义;锁标记主要是出于对数据的共享存取考虑。此外,对位置还存在两个基本方法:播放方法 `play` 和停止方法 `stop`。播放方法是当对该位置中媒体数据播放时调用,首先对该数据加锁,然后调用该媒体数据所属类的播放程序,并在播放完毕时对数据解锁;停止方法是指,首先调用该数据所属媒体类中的停止方法,然后对所引用的媒体数据进行解锁。

通过变迁、位置链和位置,我们可以建立上节中所述的对时序关系表达的 PETRI 网。此外,我们还引入了节点链表、媒体对象链表作为节点、媒体对象之间联系的中间结构。采用这些数据结构,我们可以对多媒体系统中的信息进行存储管理,并进行存取和播放。下一节将详细介绍存取播放的算法实现。

3.2 存取播放

采用上面的数据结构,并结合应用系统中节点、链和锚的具体内容结构,我们可以定义相应的数据库结构来实现对系统中多媒体信息的存储;利用节点链表、媒体对象链表和节点、链、锚具体的数据内容结构,我们可以实现多媒体系统中的数据查询。以查询对象是节点为例,我们可以根据符合一定条件的媒体数据,查找包含该数据的节点。算法过程如下:

- step1: 根据用户提供的媒体数据的特征,对相应的媒体数据库进行查询,查找出符合条件的媒体数据;
- step2: 对每一个符合条件的媒体数据,根据它的 id,从媒体对象表找出相应的媒体对象项;
- step3: 根据该项中的 `nodefirst`,沿着 `nodelist` 即可查询出包括该数据的所有节点。

对于节点中的多媒体数据的播放,我们可以根据存储的同步模型记录的节点中媒体数据的时序关系,直接实现多媒体数据的播放。下面是具体的播放实现算法。

- step1: 对于一个节点,根据它的 `transbegin` 找到该节点所包括的 TPETRI 网中的第1个变迁 t ,然后转向 step2;
- step2: 如果变迁 t 的输出位置链 `out` 不为空值,转向 step3,否则,结束;
- step3: 沿 `placelist` 对该变迁所指出的每个输出位置 p 建立线程,在线程中调用它的 `play` 方法,对该位置中的媒体数据按该数据的缺省时间或 `interval` 指定的时间进行播放。各个线程通过通信实现同步;
- step4: 当一个位置 p 中的媒体数据播放完时,根据 `transptr` 对该位置 p 指向的下一个变迁 t' 进行 step5 中的检查;
- step5: 如果该变迁 t' 存在着同步标记,并且当前结束播放的位置 p 等于该同步标记,则对该变迁 t' 所指向的其他输入位置。调用

它们的 stop 方法,以停止对它们中的媒体数据的播放以及进行解锁.令 $t=t'$,转入 step2;如果该变迁 t' 不存在同步标记,则进行 step6 中的检查;

step6:通过逐个检查,如果该变迁 t' 所指向的所有输入位置中的媒体数据均已解锁即播放完毕.令 $t=t'$,转入 step2.

采用上面的算法,我们可以按照节点中媒体数据间原有的时序关系实现多媒体数据的播放.

4 结 论

采用同步模型 TPETRI,可以对一个节点中多个媒体数据流之间的时序关系进行表达,避免了较早地考虑数据通道的分配问题.将表达与实现区分开来.媒体数据可以是静态数据也可以是动态数据.对模型中各个位置的时间分配,可采用所涉及的具体媒体数据本身的时间特征,也可以由用户进行分配.此外,我们还采用该同步模型,建立良好的用户界面.用户可以根据自己的需要,利用多媒体信息库中的数据,建立相应的时序关系,以便于多媒体信息的浏览.

由于同步模型 TPETRI 对每一个同步关系存在着唯一的表达结果,并且对于不同的时序关系表达也不相同,另一方面,从“实现策略”一节可以看出,我们对 TPETRI 直接采用有向图进行表达,因此,通过图遍历,可以实现对媒体间时序关系的维护,并且在表达存储时避免了大量中间结构的出现,减少了数据冗余.

综上所述,同步模型 TPETRI 对文章开始时提出的问题较好地进行了解决.结合多媒体数据模型,我们将其应用于多媒体创建浏览系统 SUPSYS^[7]的重新实现与开发中,不仅实现了多媒体数据的存储管理,而且还对节点中存在的媒体数据之间的时序关系进行存储维护,并在数据访问时直接按照它们原有的时序关系进行播放,方便了数据的存取.但在实际的应用过程中,我们发现,采用该模型和算法进行数据间同步控制时,对于好的硬件平台,数据呈现效果较好,但对于较差的硬件平台和同步数据流数目较大时,数据流的实际呈现与表达的时序关系不能准确匹配.究其原因,对同步关系不仅需要表达层上建立模型,还需要在实现层上进一步地考虑更细一级的数据同步规划问题,这正是我们当前主要的工作之所在.

参 考 文 献

- 1 Conklin J. Hypertext: an introduction and survey. Computer, 1988,20(9):17~41
- 2 Hall Pav. Hypertext systems and application. Information and Software Technology, 1990,32(7):447~490
- 3 Thomas D C Little *et al.* Synchronization and storage models for multimedia objects. IEEE Journal on Selected Areas in Communications, 1990,8(3):413~427
- 4 Gemmel D J, Christodoulakis S. Principles of delay-sensitive multimedia data storage and retrieval. ACM Transactions on Information Systems, 1993,10(1):51~90
- 5 Prabhakaran B, Raghavan S V. Synchronization models for multimedia presentation with user participation. In: Proceedings of ACM Multimedia'93. New York: Association of ACM for Computing Machinery, 1993. 157~166
- 6 陆汝铃. 计算机语言的形式语义. 北京:科学出版社,1992. 71~94
(Lu Ru-qian. The Formal Semantics of Computer Languages. Beijing: Science Press, 1992. 71~94)
- 7 黄波,何志均. SUPSYS: 一个超文本开发使用系统. 见:第1届全国多媒体研讨会论文集. 杭州:浙江大学出版社,1993
(Huang Bo, He Zhi-jun. SUPSYS: a hypertext development and application system. In: Proceedings of the 1st National Workshop on Multi-media. Hangzhou: Zhejiang University Press, 1993. 212~217)

The Building and Implementation of a Synchronization Model for Multimedia Data

HUANG Bo HE Zhi-jun

(State Key Laboratory of CAD & CG Zhejiang University Hangzhou 310027)

(Artificial Intelligence Institute Zhejiang University Hangzhou 310027)

Abstract In this paper, temporal problems and temporal relationships among multimedia dataflow are introduced, then related research is analyzed. Integrating with a multimedia data model, a synchronization model based on PETRI net is presented to represent temporal relationships among multimedia data. According to these two models, corresponding data structure is built and an algorithm for multimedia data retrieval is proposed.

Key words Multimedia, temporal problem, temporal relationships, PETRI net, synchronization model, data retrieval.