多媒体对象的Agent展示集成模型*

巩志国 周龙骧

(中国科学院数学研究所 北京 100080)

摘要 根据多媒体对象的面向对象特点,给出了一种基于"事件-条件-动作"的Agent展示集成模型.该模型集对象的内容、时序、空间关系于一体,通过消息传递与对象展示状态事件控制对象展示集成的行为,保证了对象的封装性,并对用户的交互提供有力的支持,是一种动态集成模型."事件-条件-动作"机制是主动数据库中广泛采用的方法,已有成熟的技术支持.通过对Allen所定义的时序关系的表示,说明了模型的表达能力,并通过实例加以验证.指出了展示集成模型在多媒体数据库中的作用.

关键词 Agent模型,展示集成,面向对象,多媒体对象,时序关系,空间关系.

中图法分类号 TP391

当前在多媒体数据模型的研究中,面向对象模型普遍被认为是一种比较合适的多媒体数据描述方法.[1~6]这主要是因为面向对象理论中的继承性、封装性和可扩展性非常适合于多媒体对象的特点:多媒体对象操作的复杂性对外透明;多媒体对象之间复杂的语义关系被模型化;方法的可重构性使具有相同操作和管理特点的多媒体对象接口统一[1];面向对象机制对于基于集合的查询与导航式浏览的支持等.这些特点导致众多专家学者对它的偏爱.但是,对于多媒体对象的集成(Integration),面向对象的引用(Reference)特性不足以刻画多媒体对象之间的展示关系(Presentation Relation).多媒体数据与传统数据(字符、数值)之间的重要区别在于,其展示特性必须被模型化.复合多媒体对象不仅要体现与子对象之间的内容语义关系(Reference),而且还必须体现复合媒体中各子对象之间的时间调度关系和空间布局关系,否则,势必影响用户对复合媒体对象内容的理解.因此,多媒体对象之间的展示关系必须被充分考虑.

纵观多媒体对象的各种集成方法[3,7~10],其时序关系一般是通过链接方式、图方式或算子方式来刻画的.链方式就是通过在对象之间定义链的方法来表示多媒体对象之间的时序同步.[7,8]这种方法的语义简单,比较自然,但对于同步语义的表示则非常有限,不足以反映出多媒体对象时序同步的动态变化特征.另外,链接定义往往是基于实例的,繁琐且易出错.算子方式是用符号表示由Allen[11]所定义的13种时序关系.进而,多媒体对象之间的时序同步可以用这些算符与对象的有效表达式来表示.[3]直接通过单独定义的时序运算符来操作多媒体对象,显然破坏了对象机制的封装特性.图方式是指用加以扩展的有向图方法表示多媒体对象之间的同步关系.由于时间Petri网对并行、并发及串行的表示非常有力,从而被人们普遍采用.[9,10]这种方法的特点是直观,对于时间同步具有很好的表达能力,但这种方法一般要求事先确定出媒体对象的展示时间.这一点过于苛刻,因为用户的交互及系统负荷的变化都可能动态影响展示时间.所以,用户容许有一定的抖动(Jitter)和延时(Latency).另外,这种模型是一种表示模型(Represent Model),实现时必须变换成可操作模型,其转换过程也是相当复杂的.

Agent是人工智能领域里发展起来的一种新型计算模型[12],现在尚无统一的定义.但有一点可以肯定,即Agent具有功能的连续性及自主性(Autonomy).也就是说,Agent能够连续不断地感知外界发生的以及自身状

* 本文研究得到国家863高科技项目基金资助.作者巩志国,1963年生,博士生,主要研究领域为分布式数据库系统,多媒体数据库系统.周龙骧,1938年生,研究员,博士生导师,主要研究领域为分布式数据库系统,多媒体数据库系统.

本文通讯联系人:巩志国,北京 100080,中国科学院数学研究所

本文1997-11-20收到原稿,1998-01-19收到修改稿

态的变化,并自主产生相应的动作.对 Agent 更高的要求可让其具有认知功能(Cognitive-like Function),以达到高度智能化的效果.由于 Agent 的上述特点,Agent 被广泛应用于分布计算环境,用于协同计算以完成某项任务.我们采用 Agent 模型方式控制并协调多媒体对象展示行为的集成,以克服当前多媒体对象时序同步所遇到的问题.

本文首先给出 Agent 模型的基本运作机制,分析了多媒体对象展示集成中的事件、条件和动作类型,设计了一种控制多媒体对象展示的控制元(Cell)结构,并对其表达能力进行了验证.定义了屏幕布局中的几个基本概念,进一步给出了 Agent 模型结构 BNF 描述,最后分析了该模型在多媒体数据库系统中的重要作用.

1 多媒体对象集成的基本 Agent 机制

Agent 模型的结构尚无标准可言.文献[12]把 Agent 描述为 4 个智能(Mental)构件的组合,这 4 个智能构件分别为 beliefs,capabilities,choices 和 commitments.其中 beliefs 定义了 Agent 所具有的知识;capabilities 描述的是 Agent 所具有的处理能力;choices 为 Agent 对某一动作的执行;commitments 描述了 Agent 对其他 Agent 所承担的责任.另外,为使基于这 4 个智能构件的 Agent 运作,还必须引入各种条件和规则.该方法智能描述程度高,适合于拟人化系统的刻画,但其结构复杂、实现困难,且不适合于对多媒体对象的集成描述.我们根据多媒体对象展示集成的需求,构造出基于事件(Events)、条件(Conditions)和动作(Actions)的 Agent 模型结构.其技术特点是:(1) 多媒体对象已经存在,我们对其集成操作只能通过消息传递(Message Passing);(2) 多媒体对象有主动对象(Audio,Video,Animation)与被动对象(Text,Image)之分.主动对象能够返回展示的结束状态,而被动对象的结束只有通过 Agent 发送的终止消息来实现;(3) Agent 为状态结构与通讯的封装体,可被用户或其他 Agent 直接引用;(4) 支持局部于 Agent 的展示控制,能够根据外部的展示请求或内部对象的展示状态决定自身的展示行为;(5) 若干 Agent 可以合成为更复杂的 Agent,以便完成更复杂的多媒体对象集成;(6) Events-conditions-actions 为主动数据库广泛采用的机制[13],已有成熟的技术.我们这里的 Agent 模型以此为基础,可以保证 Agent 的运行效率,且易于实现.

多媒体对象展示 Agent 集成模型的运行机制如图 1 所示.Agent 的消息接口感知外部消息、时钟报警以及对象的展示状态变化所产生的消息,过滤掉无用的消息,对事件状态进行更新,对条件进行赋值,然后,Agent 需检查条件满足的情况,进一步执行条件与动作的匹配.对于为真的条件,执行相应的动作.这里,动作执行的内容就是传送相应的展示请求到对象或设置时钟延时报警.多媒体对象接收到 Agent 所发来的消息后,会按消息内容展示,并产生相应的展示开始事件.主动对象能够反馈自身的结束状态,而被动对象展示状态的改变只有依靠 Agent 的调度命令.这种多媒体对象展示集成机制的特点在于:

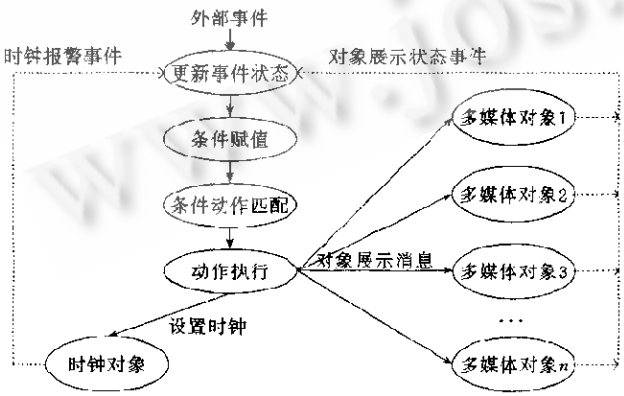

图1 Agent调度机制

(1) 模型基于 events-conditions-actions,这是主动数据库中广泛采用的机制,可使以事件驱动为核心的系统控制结构简单清晰.

(2) Agent 通过消息机制实现对多媒体对象的展示控制,维护了对象的封装性,有利于对象的分布存储.

(3) 对象展示状态的变化(展示开始、展示结束)作为它们的时序同步点,可保证主动多媒体对象(Audio,Video,Animation)的展示主动性,无需事先给出其展示时间,使多媒体对象时序同步描述更加自然.因为多媒体展示时,用户的交互比传统要频繁,这势必影响媒体的播放时间.所以,预先确定其展示时间的方法很不自然.

(4) 引入时钟对象概念.这样,可由 Agent 设置被动对象(Text,Image)的展示时间,或利用时钟对象报警强制结束主动对象的展示.

(5) 利用条件(Conditions)概念可灵活设置多媒体对象同步策略,即先者等待、令延迟者加速及终止延迟者.

(6) 利用外部事件(其他 Agent 或用户产生的事件)与内部对象的展示状态事件控制对象展示的时序集成,可使用户参与媒体的同步展示.所以,这种时序集成模式对 Hypermedia 有充分的支持.

2　事件、条件和动作

事件(Events)是 Agent 控制调度多媒体对象展示的动力,即 Agent 由事件驱动.在主动数据库理论研究与应用中,所指事件一般为基于事务的发生.我们这里所考虑的事件用来服务于多媒体对象同步展示的描述,分为两类:外部事件和内部事件.

外部事件描述了 Agent 模型的外部接口,用户或其他 Agent 只有通过产生这些外部事件才能激活、控制或结束 Agent 的活动.这些外部事件为:*start_com*:激活 Agent;*stop_com*:结束 Agent 的活动状态;$choice_1$:用户自定义事件 1;$choice_2$:用户自定义事件 2;...;$choice_n$:用户自定义事件 n.

事件 *start_com* 导致 Agent 活动的开始,在这里也就是 Agent 控制一组多媒体对象的展示调度的开始点.事件 *stop com* 可使 Agent 的展示调度结束.事件 $choice_1, choice_2, \ldots, choice_n$ 可用于外部(用户或其他 Agent)参与该 Agent 展示调度的控制.

内部事件反映 Agent 的于对象展示活动的状态或时钟状态所产生的事件,Agent 可以根据这些内部事件自主控制对象的展示调度.我们模型中的内部事件为:*object.s*:对象 *object* 展示开始;*object.e*:对象 *object* 展示结束;*h.alarming*:句柄为 h 的时钟报警.

一旦对象接收到 Agent 发来的展示请求消息,则立刻返回展示开始的状态事件.在此之后,若对象接收到 Agent 发来的终止命令或对象已自行展示结束(对主动对象来讲),则返回对象展示结束消息 *object.e* 给 Agent.在我们的调度模型中,Agent 有一个时钟子对象 *clock*,可以利用时钟的延时报警方式,让 Agent 控制对象的展示行为,时钟报警所产生的事件为 *h.alarming*,这里 h 表示所对应的报警设置产生的消息.

在主动数据库理论中,条件(Conditions)被定义为对数据库状态及事务处理历史状态的监视.[13]在我们的模型中,条件的含义是指 Agent 事件发生历史的监视.事件发生的历史状态决定了 Agent 对其对象的展示控制的执行(即动作的实施).亦即,只有在条件满足的情况下,Agent 才向其子对象传递控制命令.这样可充分发挥对象(尤其是主动对象)展示的主动性,适应于系统负荷或网络通信状态的动态变化.

我们用 e_i 表示事件(内部事件或外部事件),引进事件谓词 $occur(e_i)$,$occur(e_i)$=TRUE(FALSE)表示事件 e_i 已发生(未发生).另外,我们引入逻辑运算符"∧","∨".条件(Conditions)由事件谓词及运算符"∧","∨"递归定义得到.

定义 1. (1) 设 e_i 为任何事件,则 $occur(e_i)$为条件;(2) 有限个条件经逻辑运算符"∧","∨"的有限次运算所得的逻辑表达式仍为条件.

例如,设 e_1, e_2, e_3 为事件,则 $occur(e_i)(i=1,2,3)$,$occur(e_1) \wedge occur(e_2)$,$occur(e_1) \vee occur(e_2)$,$occur(e_1) \wedge (occur(e_2) \vee occur(e_3))$ 均为条件.对于任何多媒体对象(*MM_Object*),均对应于两种条件:展示启动条件 SC(start condition)和展示结束条件 EC(end condition).一旦展示启动条件 SC 为 TRUE,Agent 传递展示消息给该对象,随后 Agent 将监视该对象的结束条件 EC 是否为 TRUE,以便终止该对象的展示.

动作(Actions)描述了 Agent 对其多媒体对象的展示控制或对时钟对象的报警设置.动作的实现通过消息传递机制,这保证了面向对象模型的封装性.动作(消息)的格式为 *Message_Name*(*receiver*, *para_list*).在我们的多媒体对象展示集成模型中,动作类型主要有以下几种:

(1) Agent 对主动多媒体对象(*audio*,*video*)控制动作

```
start(receiver, para_list);        // 展示开始
fast(receiver, para  list);        // 展示加速
```

backward(*receiver*, *para_list*); // 倒播

fastBackward(*receiver*, *para_list*); // 快倒

stop(*receiver*); // 展示结束

(2) Agent 对被动多媒体对象(*image*, *text*)的控制动作

start(*receiver*, *para_list*); // 展示开始

stop(*receiver*, *para_list*); // 展示结束

(3) Agent 对时钟对象的控制为

setclock(*handle*, *delay*); // 设置以 *handle* 为柄,延时 *delay* 后报警

cancel(*handle*); // 取消柄为 *handle* 的报警设置

动作的执行(消息的发送)会导致对象展示状态变化的事件发生,我们特别注意的两种展示状态变化事件为展示开始 *obj*. *s* 和展示结束 *obj*. *e*. *Start* 动作的完成导致对象 *obj* 展示开始事件 *obj*. *s* 的发生,而 *Stop* 动作的完成导致对象结束事件 *obj*. *e* 的发生. 主动对象可识别到自身的展示状态的结束,自动返回事件 *obj*. *e*. 用于展示同步的另一事件为时钟报警 *handle*. *alarming*.

3 控制元对时序关系的表示

Agent 动作与受控对象展示事件的关系如图2所示.

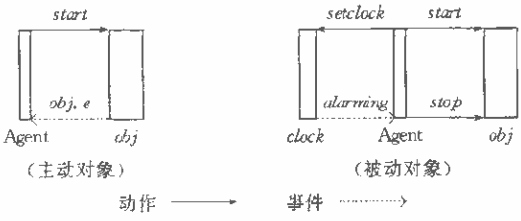

图2 动作与事件之间的关系

任何对象的展示活动均由 Agent 协调控制,针对每个对象在 Agent 控制结构中存在一个控制元 *cell*. 控制元由展示启动条件 SC,展示动作 SA,结束条件 EC,结束动作 EA 描述. 我们用 *cell* 表示图2所示的控制结构如下:设 SC 为 x(条件变量),

(1) $cell_1$={SC:x;SA:*start*(*obj*, *para_list*);EC:*occur*(*obj*. *e*);EA:∅;}(∅表示空,下同).

该控制元表示 Agent 检测到条件 x 为真后,传递开始消息给对象 *obj*,*obj* 展示结束条件为自身产生的终止消息,结束动作为空,表示无需终止命令.

(2) $cell_2$={SC:x;SA:*start*(*obj*, *para_list*), *setclock*(h, *delay*);EC:*occur*(*h*. *alarming*);EA:*stop*(*obj*)}

$cell_2$表示条件 x 为真后,触发对 *obj* 展示消息的传递,并同时设置延时为 *delay*,句柄为 h 的时钟报警. 一旦时钟报警事件 *h*. *alarming* 发生,则 *obj* 展示结束.

Allen[11]将时间关系定义为13种:before, meets, overlaps, during, starts, finishes, equals 及前6种的逆关系. 因为逆关系为非本质的,实际仅有7种基本关系. 为节省篇幅,我们仅给出两种时序关系的表示.

(1) O_1 before O_2

(1.a) O_1,O_2均为主动对象时,

$cell(O_1)$={SC:x;SA:*start*(O_1, p_1);EC:*occur*(O_1. e);EA:∅;}

$cell(\tau)$={SC:*occur*(O_1. e);SA:*setclock*(h, τ);EC:*occur*(*h*. *alarming*);EA:∅;}

$cell(O_2)$={SC:*occur*(*h*. *alarming*);SA:*start*(O_2, p_2);EC:*occur*(O_2. e);EA:∅;}

(1.b) O_1,O_2均为被动对象时,

$cell(O_1)$={SC:x;SA:*start*(O_1, p_1), *setclock*(h_1, d_1);EC:*occur*(h_1. *alarming*);EA:*stop*(O_1)}

$cell(\tau)$={SC:occur(h_1. *alarming*);SA:*setclock*(h_τ, d_τ);EC:occur(h_τ. *alarming*);EA:∅;}

$cell(O_2)=\{SC:occur(h_r.alarming);SA:start(O_2,p_2),setclock(h_2,d_2);EC:occur(h_2.alarming);$
$EA:stop(O_2);\}$

(2) O_1 overlaps O_2

(2.a) O_1,O_2均为主动对象时，

$cell(O_1)=\{SC:x;SA:start(O_1,p_1),setclock(h,\tau);EC:occur(O_1.e);EA:\varnothing;\}$

$cell(O_2)=\{SC:occur(h.alarming);SA:start(O_2,p_2);EC:occur(O_2.e);EA:\varnothing;\}$

(2.b) O_1,O_2均为被动对象时，

$cell(O_1)=\{SC:x;SA:start(O_1,p_1),setclock(h_1,d_1),setclock(h_r,d_r);EC:occur(h_1.alarming);$
$EA:stop(O_1)\}$

$cell(O_2)=\{SC:occur(h_r.alarming);SA:start(O_2,p_2),setclock(h_2,d_2);EC:occur(h_2.alarming);$
$EA:stop(O_2)\}$

以上介绍了两种用cell表示的时序关系，其余不再赘述.可见此方法对多媒体同步策略设置的灵活性.

4　多媒体展示的空间布局

多媒体展示集成的另一个重要问题是多媒体对象展示的空间布局结构(Spatial Layout Architecture).对于复合媒体对象，其各个对象的空间关系应缺省存在.这一特性应成为多媒体对象集成的一部分.一般来讲，对于*image*,*video*,*animation*等对象，在创建时已经保存了其内在显示尺寸.但是，复合媒体对象作为一个整体的布局考虑，其展示时的大小必须重新确定.*Text*对象属性一般不包含其显示区域的大小，因此在复合媒体中必须增加.而*audio*对象的显示设备为扬声器，在集成中有两种参数需考虑：音量(Volume)和平衡(Balance).除了自身的显示尺寸外，各媒体对象之间的空间布局必须给定.

文献[14]把空间关系(Spatial Relations)分为两大类：物理位置关系和逻辑位置关系.物理位置关系是指地理性的位置关系，如，东、西、南、北等，而逻辑位置关系是指抽象的位置关系.多媒体对象的空间关系是指逻辑位置关系，因此，下面所述的空间关系均指逻辑位置关系.根据文献[14]中的讨论，空间关系可分为DJ(分离)，EC(外切)，IS(内含)，OL(交搭)，CV(内切)，EQ(相等)(如图3所示).而DJ关系又可细分为LT(左)，RT(右)，AB(上)，BL(下)等方向关系.复合媒体的空间关系通过每个对象展示区域的屏幕坐标及大小被隐含确定.根据文献[15]对Screen(屏)与Frame(帧)的定义，进行适当扩充与修改，我们有：

screen::=SCREEN:{⟨frame⟩*|(⟨frame⟩*⟨speaker⟩)*|⟨speaker⟩*},

frame::=⟨x_origin,y_origin,width,height⟩,

speaker::=⟨volume,balance⟩.

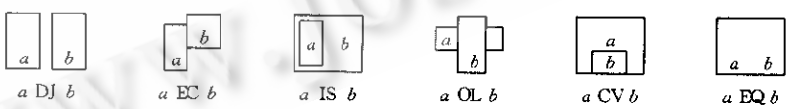

图3　空间关系

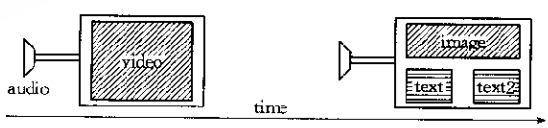

图4　屏幕布局设计

*Screen*概念为显示设备的逻辑表示，它由一个或多个*frame*及*speaker*的组合描述.而*frame*定义了可视化对象的显示区域，它由区域位置与区域大小描述.*speaker*为声频对象的显示特性描述，它由音量与平衡组成.图4刻画了复合媒体的一种布局设计.

5 Agent 展示集成模型结构

Agent 模式(Pattern) 刻画了一组多媒体对象之间的展示行为关系,完成整个复合对象的展示任务. Agent 模式给出了复合对象展示的形式化描述. 其描述要素概括为:(1) 外部事件,定义了其他 Agent 或用户对复合对象的展示操作所产生的事件;(2) Agent 所集成的每个多媒体对象的类型名;(3) 报警时钟类型名,对每一组对象集成只有一个类型名;(4) 屏幕布局描述,刻画了各对象的显示位置与大小及声音输出特性;(5) 每种多媒体类型的展示调用消息集合;(6) 每种多媒体类型展示状态产生的事件——当多媒体响应展示调用消息后,由于其展示状态的变化所产生的状态事件;(7) 一组基于 events-conditions-actions 的展示控制元 cell. cell 的定义描述了一个或几个多媒体对象并行同步关系. Agent 模式中所有控制元所构成的集合刻画了整个复合对象的展示调度行为.

定义2. 一个 Agent 集成模式 α 的 BNF 形式描述如下:

⟨α⟩::=⟨agent name⟩⟨set of external events⟩⟨set of roles⟩⟨screen layout⟩⟨set of cells⟩,

⟨agent name⟩::=string,

⟨set of external events⟩::=EXT_EVT:⟨start_com⟩⟨stop_com⟩⟨choice⟩*,

⟨set of roles⟩::=ROLES:⟨role⟩*,

⟨role⟩::=⟨object type⟩⟨message List⟩⟨status⟩⟨events⟩,

⟨object type⟩::=string,

⟨message List⟩::=⟨message⟩*,

⟨message⟩::=⟨mes_name⟩⟨receiver⟩⟨parameter_list⟩,

⟨status⟩::=⟨alive⟩⟨dead⟩,

⟨events⟩::=⟨start events⟩⟨stop events⟩,

⟨screen layout⟩::=SCREEN:{⟨frame⟩*|(⟨frame⟩*⟨speeker⟩)*|⟨speaker⟩*},

⟨frame⟩::=⟨x_origin,y_origin,width,height⟩,

⟨speaker⟩::=⟨volume,balance⟩,

⟨set of cell⟩::=CELLS:⟨cell⟩*,

⟨cell⟩::=⟨start condition⟩⟨start actions⟩⟨end condition⟩⟨end actions⟩.

6 应用举例

某城市旅游信息由下列多媒体对象组成:(1) 城市风光视频对象:*City_Video*;(2) 相对于 *City_Video* 的声频注解对象:*City_Audio*;(3) 两张古建筑照片:$Image_1$,$Image_2$;(4) 与每张古建筑照片相对应的一段中文文字介绍:$Chinese\_Text_1$,$Chinese\_Text_2$;(5) 相对于每张古建筑照片的英文文字介绍:$English\_Text_1$,$English\_Text_2$. 由以上各多媒体对象构成多媒体复合对象 *City_Scene*,对 *City_Scene* 中各对象之间的展示关系描述如下:

(1) 复合对象展示开始播放 *City_Video* 时,声频对象 *City_Audio* 并行播出. 因为 *City_Video*,*City_Audio* 皆为主动对象,其展示状态具有主动性,要求在这两个对象都结束后,才进行以后的展示.

(2) *City_Video*,*City_Audio* 并行播放完后,自动进入古建筑照片的展示,照片的展示顺序为 $Image_1$,$Image_2$. 每张照片显示时,同时伴随其中文文字介绍及英文文字介绍. 因为照片、文本皆为被动对象,我们利用时钟设置其时间,每张照片展示 t 秒. 最后一张照片展示完后,整个复合对象展示结束.

Agent 展示集成定义如下:

```
City_Scene
{
EXT_EVT:start_com,stop_com;
ROLES:
  video:City_Video;
```

audio:$City\_Audio$;
image:$Image_1$,$Image_2$;
text:$Chinese\_Text_1$,$Chinese\_Text_2$,$English\_Text_1$,$English\_Text_2$;
SCREEN:
$City\_Video\_frame=(x_1,y_1,width_1,height_1)$;
$City\_Audio\_speaker=(volume,balance)$;
$Image_1\_frame=Image_2\_frame=(x_2,y_2,width_2,height_2)$;
$Chinese\_Text_1\_frame=Chinese\_Text_2\_frame=(x_3,y_3,width_3,height_3)$;
$English\_Text_1\_frame=English\_Text2\_frame=(x_4,y_4,width_4,height_4)$;
CELLS:
$cell_1=$
{
SC:$occur(start\_com)$;
SA:$start(City\_Video,City\_Video\_frame)$,$start(City\_Audio,City\_Audio\_speaker)$;
EC:$occur(City\_Video.e)\wedge occur(City\_Audio.e)$;
EA:$\varnothing$;
}
$cell_2=$
{
SC:$occur(City\_Video.e)\wedge occur(City\_Audeo.e)$;
SA:$start(Image_1,Image_1\_frame)$,
$start(Chinese\_Text_1,Chinese\_Text_1\_frame)$,
$start(English\_Text_1,English\_Text_1\_frame)$,
$setclock(h_1,t)$;
EC:$occur(h_1.alarming)$,
EA:$stop(Image_1)$,$stop(Chinese\_Text_1)$,$stop(English\_Text_1)$;
}
$cell_3=$
{
SC:$occur(h_1.alarming)$;
SA:$start(Image_2,Image_2\_frame)$,
$start(Chinese\_Text_2,Chinese\_Text_2\_frame)$,
$start(English\_Text_2,English\_Text_2\_frame)$,
$setclock(h_2,t)$;
EC:$occur(h_2.alarming)$,
EA:$stop(Image_2)$,$stop(Chinese\_Text_2)$,$stop(English\_Text_2)$;
}

7 总　结

时序属性与空间属性为多媒体对象所固有的性质.因此,多媒体对象的集成不但要包括内容集成,而且还必须考虑时序集成与空间集成.同样的内容,不同的时序集成或空间集成会有不同的视觉效果,甚至影响对复合对象的理解.

在多媒体数据库理论研究文献中[1~5],一般比较一致地选择面向对象的范型作为多媒体对象存储与管理的模式.通过 PART_OF 关系定义复合对象的内容构成.但是,这些文献对于多媒体复合中的时序关系与空间关系的讨论较少,甚至没有涉及.例如,文献[4]将对象之间的空间关系抽象地表示为 Left,Right,Above,Below 等,时序关系表示为 Before,After,During,At 等,而这些关系的实现只能在应用中解决.文献[3]是在对象概念之上,利用算子的方式定义复合对象的展示关系,该方法存在同样的问题.另一方面,在商业化的数据库系统中(DB2,Informix,Oracle,Sybase 等),通常提供抽象数据类型(Abstract Type)及用户自定义函数(User Defined Functions)等特性,将多媒体数据存储为 BLOB 方式,使关系模式扩充为对象-关系型模式.DB2是这一技术的典型代表.[16]这种范型依赖于关系数据库成熟的技术,因此具有高效、可靠、实用等特点.对于多媒体对象的集成,可以利用在同一元组(Tuple)中定义多个 BLOB 项或联结(Join)运算实现内容上的集成.但对于对象之间的时序与空间关系描述没有提供有效的手段,仍需在应用中解决.

多媒体对象的集成是多媒体数据库系统的重要组成部分.根据我们已经开发成功的多媒体数据库系统 CDB/M,我们对其进行了扩展[6,15],其中包含了多媒体对象的展示集成功能.我们所给出的 Agent 集成模型是

一种对象行为上的集成.该模型集内容、时序、空间关系于一体,描述了复合对象与其子对象之间的通讯关系.所定义的对象之间的展示关系无需翻译,可直接实现.集成模型与对象模型为紧耦合,充分保证了对象的封装性及多媒体数据库的高效率.同时,展示集成模型对用户的交互具有充分的支持,是一种动态集成模型.使用户不但在对象级,而且在对象展示行为上得到共享.我们拟用独立于平台的 Java 语言重写多媒体数据库系统 CDB/M,这样有利于 Agent 在网络环境中的流动.另外,对于网络通信延迟及用户的 QoS 需求对多媒体展示集成的影响,我们仍在研究中.

参考文献

1 Ishikawa H, Suzuki F *et al*. The model, language, and implementation of an object-oriented multimedia knowledge base management system. ACM Transactions on Database Systems, 1993,18(1):1～50

2 Ishikawa H *et al*. A next-generation industry multimedia Database System. In: Stanley Y, Su W eds. Proceedings of the 12th International Conference on Data Engineering. Los Alamitos, CA: IEEE Computer Society Press, 1996. 364～371

3 Schloss G A, Wynblatt M J. Providing definition and temporal structure for multimedia data. Multimedia Systems, 1995,3(5):264～277

4 Narashimhalu A. Desai. multimedia databases. Multimedia Systems, 1996,4(5):226～249

5 Chen C, Roger Y *et al*. Design of a multimedia object-oriented DBMS. Multimedia Systems, 1995,3(5):217～227

6 Gong Zhi-guo, Zhou Long-xiang. Analysis and study of multimedia database system. In: Masao Ito, Zhong Xi-chang eds. Proceedings of the International Symposium on Future Software Technology (ISFST-97). Tokyo: Software Engineers Association, 1997. 125～132

7 ISO. Hypermedia/Time-based structure language: HyTime (ISO 10744). International Standard Organization, 1992

8 ISO. Multimedia and hypermedia information coding expert group. ISO/IEC JTC1/SC29/WG12, MHEG Working Draft "WD. 1.0", Version 1.C, Feb. 1993

9 Little T D C, Ghafoor A. Synchronization and storage models for multimedia objects. IEEE Journal on Selected Area in Communications, 1990,8(3):413～427

10 Diaz M, Senac P. Time stream Petri nets: a model for multimedia streams synchronization. In: Tat-Seng Chua, Tosiyasu L Kunii eds. Multimedia Modeling (MMM'93), the 1st International Conference on Multimedia Modeling. Singapore: World Scientific Press, 1993. 257～273

11 Allen J F. Maintaining knowledge about temporal intervals. Communications of the ACM, 1983,26(11):832～843

12 Shoham Y. Agent-oriented programming. Artificial Intelligence, 1993,60(1):51～92

13 Fraternali P, Tanca L. A structured approach for the definition of the semantics of active databases. ACM Transactions on Database Systems, 1995,20(4):414～471

14 Li J Z *et al*. Spatial reasoning rules in multimedia management systems. In: Courtial J P ed. Multimedia Modeling (MMM96), the 3rd International Conference on Multimedia Modeling. Singapore: World Scientific Press, 1996. 119～133

15 周龙骧,柴兴无.分布式多媒体数据库系统的分层体系结构.计算机学报,1996,19(7):481～491
(Zhou Long-xiang, Chai Xing-wu. Hierarchical Architecture of Distributed Multimedia Database Systems. Chinese Journal of Computers, 1996,19(7):481～491)

16 URL http://WWW.software.ibm.com/data/db2/db2v2.html

An Agent Model for Integration of Multimedia Object Presentation

GONG Zhi-guo ZHOU Long-xiang

(*Institute of Mathematics The Chinese Academy of Sciences Beijing* 100080)

Abstract In this paper, an agent model for the integration of multimedia object presentation is established by using events-conditions-actions mechanism and considering object-oriented characteristics of multimedia objects. This model is a dynamic model, which integrates contents, temporal relations and spatial relations of multimedia objects and supports user interactions. Because it controls objects by message passing, the encapsulations of objects can be maintained. Events-conditions-actions mechanism is employed in active database intensively and has mature technologies. The expressive ability of this model is described by using it to the temporal relations defined by Allen, and it is also verified with an example. Its contributions to multimedia database systems are also analyzed.

Key words Agent model, presentation integration, object-oriented, multimedia objects, temporal relations, spatial relations.