

多媒体技术

第五章 基于时间媒体的表现与同步

主要内容:

基本概念: 理解基本概念

参考模型

同步关系说明: 掌握常用的同步关系说明方法

分布环境中的同步

第五章 基于时间媒体的表现与同步

1. 概述

1.1 时间依赖媒体的定义

基于时间的媒体 (Time-based Media, 又称时基媒体), 或称时间依赖媒体 (Time-dependent):

多媒体是在不同环境中文本、图像、声音、视频等各种媒体的集成。在这些数据中, 有的媒体与时间密切相关, 例如动态图像中的音频和视频, 它们与时间有着强烈的依赖关系, 在采样和回放时需要以时间为参数系进行有序的组织。这些媒体就称为基于时间的媒体 (Time-based Media, 又称时基媒体), 或者称为时间依赖媒体 (Time-dependent)。

第五章 基于时间媒体的表现与同步

1. 概述

1.2 多媒体同步或合成

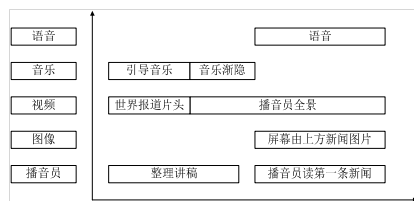
组织时基的、非时基的多种媒体序列以达到某种表现效果的任务, 就称为多媒体同步或合成。

同步即可以用于并发或顺序的数据流布局, 也可以用于对所产生的外部事件进行安排。

第五章 基于时间媒体的表现与同步

1. 概述

1.2 多媒体同步或合成



多媒体同步的时间线表示方法: 在各种媒体之间, 包括时基媒体和非时基媒体, 都根据时间进行了组织, 对他们之间的时间关系进行了描述。

第五章 基于时间媒体的表现与同步

1. 概述

1.2 多媒体同步或合成

多媒体同步需考虑的其他问题:
除了简单的时间依赖媒体序列进行线性播放外, 多媒体系统也应该支持对数据的各种操作, 例如反转、快进、快退及随机存取等。

在多媒体系统中, 由于非顺序存储、数据压缩、数据分布和随机通信延迟等各种技术和特性的引入, 给上述操作带来了困难。

因此, 研究时基媒体以及时基媒体的时间特性, 将有利于多媒体系统的设计、时基媒体的处理、同步关系的描述等各个方面。

问题: 时基媒体的表示方法、时间模型、用户交互以及系统实现方法。

第五章 基于时间媒体的表现与同步

1. 概述

1.3 媒体内同步和媒体间同步

根据数字视频和音频的时间采样特性，要求从产生到恢复表现都要严格的限制延时和抖动的范围，这个要求就称为媒体内同步。

如果有几个连续媒体流并行表现，从不同的产生点或恢复点对它们进行相互时序关系限制则称为媒体间同步。

7

第五章 基于时间媒体的表现与同步

1. 概述

1.4 现场直播同步和人工合成同步

现场直播同步与人工合成同步的最大区别是时序关系的决定方式不同。

现场直播同步的目标是精确的再现在捕获过程中自然存在的时序关系。

人工合成同步的时序关系则是由人工指定的。

8

第五章 基于时间媒体的表现与同步

1. 概述

1.4 现场直播同步和人工合成同步

现场直播同步的主要要求是根据媒体数据捕获过程中已存在的时序关系来展现媒体对象。

人工合成同步：时序关系被人工的赋给独立创建的媒体，人工合成同步经常用于将存储好的媒体对象重新安排组合成新的多媒体对象的表现系统。对人工合成媒体同步，必须使用一个模型来说明时序同步的条件和具体的操作。人工合成同步分成两个明显的阶段：

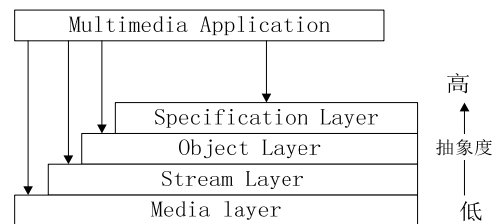
说明阶段：定义媒体对象之间的时序关系；

表现阶段：系统在运行时以同步模式来展现媒体数据。

9

第五章 基于时间媒体的表现与同步

2. 多媒体同步的参考模型



10

第五章 基于时间媒体的表现与同步

2. 多媒体同步的参考模型

2.1 媒体层

在媒体层，应用程序通常对单一连续的媒体流进行操作，这些媒体流通常被看做一些逻辑数据单元的序列。在这一层提供的抽象封装接口是一些与读、写操作相关的设备无关的函数。

11

第五章 基于时间媒体的表现与同步

2. 多媒体同步的参考模型

2.2 流层

流层是作用在一组连续的流上，在这一组的所有流都是通过流间的同步机制来实现并行表现。该层提供的抽象封装接口是具有时间参数的流，这些流具有流内的同步和流间的同步。

12

第五章 基于时间媒体的表现与同步

2. 多媒体同步的参考模型
2.3 对象层

对象层对各种媒体类型进行操作，它隐藏了离散媒体与连续媒体的区别。提供给应用程序的抽象封装是完全的同步多媒体表现，这一层将同步要求作为参数给出，并对整个表现的正确调度负责。

功能：计算和执行完全的表现规划，调用流层的服务。

第五章 基于时间媒体的表现与同步

2. 多媒体同步的参考模型
2.4 同步关系说明层

同步关系说明层是一个开放的层，它不提供任何显式的编程接口。在这一层可以创建同步关系说明要求。

详细说明见《多媒体技术：计算、通信和应用》P326。

第五章 基于时间媒体的表现与同步

3. 同步关系说明

多媒体对象的同步关系说明描述了多媒体对象中包含的各对象之间的时间依赖关系。它是由同步关系说明层的工具产生，并由对象层的接口所使用。由于同步关系说明决定了整个表现，因此，它成了多媒体系统的重要问题。同步说明应该包括以下内容：

- 对象内的同步关系说明，主要用于媒体对象的表现
- 对象内同步的服务质量描述
- 媒体对象表现中的对象间的同步关系说明
- 对象间同步的服务质量描述

第五章 基于时间媒体的表现与同步

3. 同步关系说明

在现场直播同步中，时序关系在捕获时就隐含定义了，对某一媒体服务质量的要求时在开始捕获前就定义了。

在人工合成同步中，同步关系说明必须显式创建。

第五章 基于时间媒体的表现与同步

3. 同步关系说明

同步关系说明应满足下述要求：

- 它应支持对象的一致性，媒体对象在说明中作为一个逻辑单元；
- 它应能对媒体对象内容进行抽象，允许对媒体对象的某一个部分的时间关系进行说明，但同时保持把媒体对象作为一个逻辑单元；
- 同步关系的所有类型应容易描述；
- 应支持时间相关媒体对象和时间无关媒体对象的集成；
- 应支持服务质量要求的定义，最好是由该方法直接表达；
- 应支持层次式同步，以适应大型复杂的同步场景。

第五章 基于时间媒体的表现与同步

3. 同步关系说明
3.1 基于间隔的同步说明方法

时间点：时间点是时间的零长度运动。

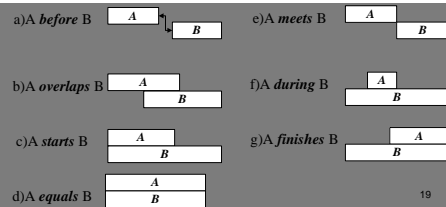
时间段：是由两个时间点定义的，所以它们就有了一个持续区间。用区间而不用端点指明时间，可以把区间从绝对的或瞬时的参考系中分离出来，而给出一个时间上的相对关系。

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.1 基于间隔的同步说明方法

两个时间间隔可以以13种不同的方式同步。可以精简为7种（因为其中一些是互逆的）。



19

第五章 基于时间媒体的表现与同步

3. 同步关系说明

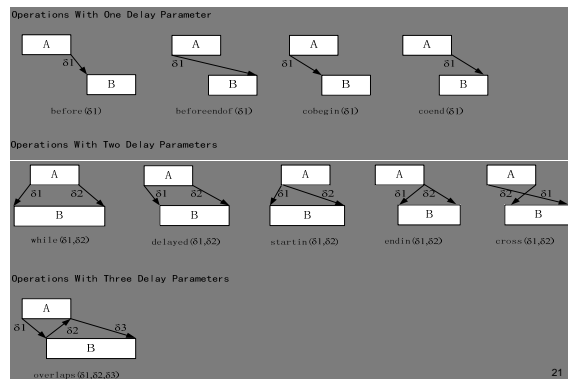
3.1 基于间隔的同步说明方法

用这几个基本的关系可以对多媒体表现进行建模。用区间表示多媒体对象的时间成分，用相对的位置来表示它们在时间上的依赖关系。

基于这种时间关系的增强方法如下图所示：

20

第五章 基于时间媒体的表现与同步



21

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.1 基于间隔的同步说明方法

例：一个有几个幻灯片Slide i ($1 \leq i \leq n$) 和一个音频对象的幻灯片放映，在这个模型中可以被说明如下：

slide1 cobegin(0) audio

Slide i before (0) slide $i+1$ ($1 \leq i \leq n-1$)

22

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.1 基于间隔的同步说明方法

这个模型允许对时间相关媒体对象和时间无关媒体对象的持续时间进行说明。

优点	缺点
可以保持逻辑对象	说明复杂
媒体内容的抽象较好	对服务质量的偏差需要额外说明
时间无关对象集成容易	支持媒体对象的时序关系的直接说明，但对其子单元却不行
交互对象的集成容易	
支持不确定性的时序关系的说明	

23

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.2 基于基准轴的同步说明

基准轴：被所有对象的表现共享。

在基于基准轴的同步说明中，开始和结束等表现事件被映射到基准轴上。

可分为两种方式：

📅 基于全局定时器的同步

📅 基于虚拟轴的同步

24

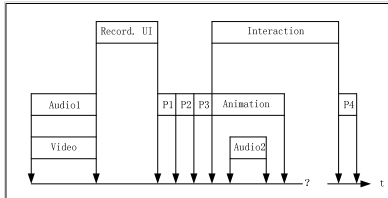
第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.2 基于基准轴的同步说明

3.2.1 基于全局定时器的同步

对于基于全局定时器的同步而言，所有多媒体对象被附加到由实际时间抽象而来的时间轴上。删除一个对象不影响其他对象的同步。



25

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.2 基于基准轴的同步说明

3.2.1 基于全局定时器的同步

上图给出了同步说明的例子。可以看出，由于用户交互的持续时间是不可预测的，因此对这种对象的处理就会产生一些问题。

优点	缺点
容易理解	无法预测持续时间的对象集成困难，需要扩充
容易实现层次式说明	必须给出附加的服务质量说明
时间无关对象集成容易	
由于对象之间的无关性，容易进行同步维护	

26

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.2 基于基准轴的同步说明

3.2.2 基于虚拟轴的同步

虚拟轴同步说明方式是时间轴同步说明方式的推广。

在这一说明方法中，可以为用户定义的度量单元说明坐标系统，然后根据这些坐标轴来执行同步说明。也可以使用几个虚拟轴来创建一个虚拟坐标空间。

例如，由音符描述音乐的例子，音调的频率由音符线的位置来定义，顺序和持续时间由轴上的节拍度量单元来定义。

虚拟轴到实际轴的映射是在运行时动态进行的。

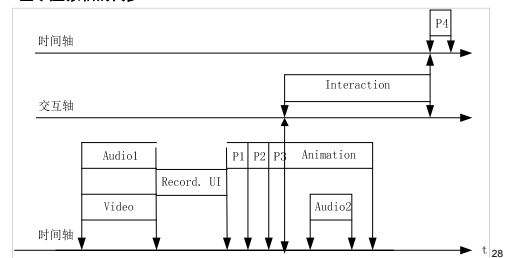
27

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.2 基于基准轴的同步说明

3.2.2 基于虚拟轴的同步



28

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.2 基于基准轴的同步说明

3.2.2 基于虚拟轴的同步

优点	缺点
容易理解	多个轴，则同步说明复杂度增加
容易实现层次式说明	运行时的轴映射将增加复杂度，耗时
时间无关对象集成容易	服务质量：只能隐含定义或通过附加说明定义
由于对象之间的无关性，容易进行同步维护	
说明通常是根据问题空间的可能性进行	
通过使用特殊轴，可以将交互对象包含进来	

29

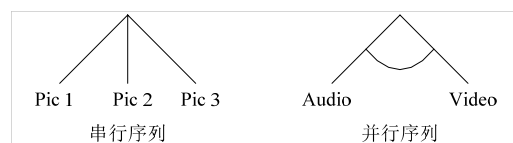
第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.3 基于控制流的同步说明

基本的层次式说明

层次式同步描述是基于两种主要的同步操作：动作串行同步和并发同步。在层次式同步说明中，多媒体对象被看做是一个树节点，该节点包括被标记为并行或串行表现的子树。



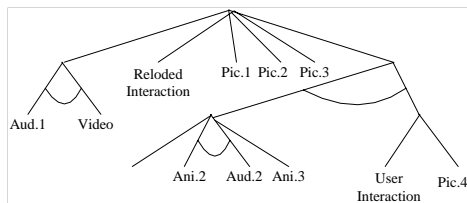
30

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.3 基于控制流的同步说明

基本的层次式说明



31

第五章 基于时间媒体的表现与同步

3. 同步关系说明

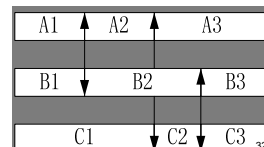
3.3 基于控制流的同步说明

基本的层次式说明

层次式结构容易被处理并被广泛的使用。

其限制：每一操作行为只能在其开始或结束时被同步。在某些应用如视频流与字幕的配合，就必须把视频流分成几个连续的部分。

另外，有的同步不便于用层次模型表达。例如，任何两个对象都有同步关系，但独立于第三各对象。如图。



32

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.3 基于控制流的同步说明

基本的层次式说明

优点	缺点
容易理解	需要额外的服务质量偏差描述
自然支持层次结构	对时间无关媒体的表现，必须加入表现的持续时间
可集成交互式对象	为了同步的目的，有时必须对媒体对象做分解

33

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义

有向网(directed net)，简称网(net)，为一个三元组，我们将其表示为 $N(S, T, F)$ ， $X=S \cup T$ 为其元素集。其完全形式化涉及三元谓词Net:

【定义4-1】有向网 (Directed Net) :

$Net(S, T; F) \Leftrightarrow S \cup T \neq \emptyset \wedge S \cap T = \emptyset \wedge F \subseteq S \times T \cup T \times S \wedge dom(F) \cup cod(F) = S \cup T$

“x”为笛卡尔积，

$dom(F) = \{x | \exists y: (x, y) \in F\}$ ， $cod(F) = \{y | \exists x: (x, y) \in F\}$ 分别为F的定义域和值域。

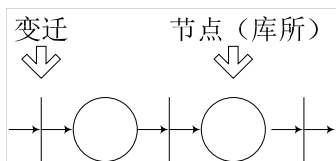
34

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义



35

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义

S和T分别称为有向网的库所 (place) 集和变迁 (transition) 集，F为流关系 (flow relation)。库所和变迁又分别称为S_元素和T_元素，或S_元和T_元。 $X=S \cup T$ 称为Net的元素集。

设 $x \in X$ 为N的任一元素，则： $x' = \{y | (y, x) \in F\}$ 为x的前集 (pre-set) ；

$x'' = \{z | (x, z) \in F\}$ 称为x的后集 (post-set) 。

36

$x = \{y \mid (y, x) \in F\}$

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义

对于有向网 $N(S, T; F)$ ，我们记 $N_0 = \{0, 1, 2, \dots\}$ ， $N = \{1, 2, 3, \dots\}$ ，并以 ω 表示无穷； $\omega = \omega + 1 = \omega - 1 = \omega + \omega$ 。

(1) N 的容量函数： $K: S \rightarrow N \cup \{\omega\}$ 称为 N 的容量函数；

(2) 对给定的容量函数 K ， $M: S \rightarrow N_0$ 称为 N 的一个标识的条件是：
 $\forall s \in S: M(s) \leq K(s)$ ；

(3) $W: F \rightarrow N$ 称为 N 上的权函数，对 $(x, y) \in F$ ， $W((x, y))$ 称为 (x, y) 上的权。

M ：资源分布标识。标识是资源在库所中的一种分布。
 W ：定义每个变迁发生一次引起的有关资源数量上的变化。

37

$x = \{y \mid (y, x) \in F\}$

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义

网系统的定义：

六元组 $\Phi = (S, T; F, K, W, M_0)$ 构成网系统的条件是：

(1) 其中， $N = (S, T; F)$ 构成有向网，称为 Φ 的基网；

(2) K 、 W 、 M_0 依次为 N 上的容量函数、权函数和标识， M_0 称为 Φ 的初始标识。

38

$x = \{y \mid (y, x) \in F\}$

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义

网系统的容量函数 K 和权函数 W 可以分为三种情况

(1) $K \equiv 1$ ， $W \equiv 1$ ；每个 S 元只有“有Token”和“无Token”两种状态

(2) $K \equiv \omega$ ， $W \equiv 1$ ；传统上被称为P/T_网

(3) K 和 W 为任意函数；通常被称为库所/变迁系统（place/transition system）或P/T_系统

39

$x = \{y \mid (y, x) \in F\}$

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.1 Petri网基本定义

Petri的三种不同含义：

(1)指上述定义中的有向网 $N = (S, T; F)$ ；

(2)指上述定义中的网系统，但 $K \equiv \omega$ ， $W \equiv 1$ ；

(3)泛指以有向网为基础的整个学科；

40

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.2 时间化的Petri网

时间化Petri网的规则：

一个Transition，称为一个变迁，一个变迁有输入节点和输出节点。在这些变迁和节点中，有令牌在移动。当一个变迁的所有输入点都包含一个未阻塞的令牌，则这个变迁被激活。一个变迁被激活以后，其每个输入节点的令牌都被拿走，并在其每个输出节点中放入一个令牌。

一个令牌被放到一个新的节点中，将被阻塞。阻塞的时间即这个节点被标注的持续时间。

对时间相关的媒体对象，Petri网的每一节点表示一个逻辑数据单元，通过“变迁”将适当的逻辑数据单元联系起来，就对同步进行了描述。

41

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.2 时间化的Petri网

幻灯片放映的例子，可以用时间化petri网来表示。

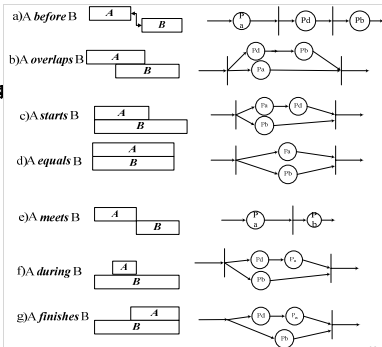
42

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.2 时间化的Petri网



43

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.2 时间化的Petri网

时间关系	参数关系	总持续时间
$P_a \text{ before } P_b$	$t_d \neq 0$	$T = t_a + t_b + t_d$
$P_a \text{ meets } P_b$	$t_d = 0$	$T = t_a + t_b$
$P_a \text{ overlaps } P_b$	$t_a < t_b + t_d, t_d \neq 0$	$T < t_a + t_b$
$P_a \text{ during } P_b$	$t_a + t_d < t_b, t_d \neq 0$	$T = t_b$
$P_a \text{ starts } P_b$	$t_a + t_d = t_b, t_d \neq 0$	$T = t_b$
$P_a \text{ finishes } P_b$	$t_a + t_d = t_b, t_d \neq 0$	$T = t_b$
$P_a \text{ equals } P_b$	$t_a = t_b, t_d = 0$	$T = t_a = t_b$

44

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.2 时间化的Petri网

当一些逻辑数据单元与其他逻辑数据单元之间不存在对象间的同步时，也可以把这些连续的逻辑数据单元集成为一个节点。

层次化结构的建立：在每一个节点上创建子网就可建立层次结构。子网中最长路径的持续时间就被赋予相关的节点。

时间化Petri网允许所有的同步方式的说明，其主要的缺点是说明复杂，而且媒体对象内容的抽象不够充分。这主要是由于媒体对象必须被解剖为子对象才能描述其时间上的同步关系。

45

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.4 时间化Petri网

3.4.2 时间化的Petri网

优点：可创建层次。容易集成时间无关对象。容易集成交互对象。

缺点：说明复杂。需要将媒体对象分为子对象。

46

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

脚本：用来表示特定领域内的特定问题。

SMIL：同步多媒体集成语言（Synchronized Multimedia Integration Language），由3W(World Wide Web Consortium)组织发布的标准。

SMIL基于XML语法，与HTML（超文本标记语言）的语法格式较为相似。HTML主要针对普通的网络媒体文件进行操作（文字、图片、声音、动画、视频的简单安排）。

SMIL则操纵多媒体片断（对多媒体片断的有机的、智能的组合）。

47

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL语言的优点：

SMIL语言是一套由W3C定义的标记语言，标记简单。它用来规定多媒体片断(这里多媒体包括的范围有：声音文件、视频文件、动画、图片、文字等)在什么时候、在什么地方、以什么样的方式播放。

48

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL语言的优点:

- (1) 避免使用统一的包容文件格式
- (2) 同时播放在不同地方(服务器上)的多媒体片断
- (3) 时间控制: 在同步控制上, SMIL采用了全局时间轴和相对时间结合的时间轴模型。

- (4) 支持对整个演示进行布局
- (5) 多语言选择支持
- (6) 多带宽选择支持

49

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL语言的结构: 和HTML类似

```
<smil>
<head>
<meta name="copyright" content="Your Name" />
<layout><!-- layout 标记,安排布局 --></layout>
</head>
<body>
<!--媒体标记 -->

</body>
</smil>
```

50

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL和HTML语言的语法格式相似。但由于SMIL是基于XML的标记语言,所以它的语法要求比HTML严格。

- 在<smil></smil>中,分为header和body两个部分。
- 遵循XML语法,大小写敏感。这里,SMIL标记和属性的名称都用小写的方式
- 有匹配的结束标记,空标记的结束也用"/":
- 属性值必须用引号
- SMIL文件的拓展名为*.smil或者*.smi
- 附加信息写在<head></head>之间
- 用<!-- ... -->进行注释

51

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明

两个基本标记: <seq> <par>

若干属性:

- a) dur属性: 持续时间
- b) begin和end属性: 什么时候开始播放。
- c) clip-begin和clip-end属性,
- d) fill属性:
- e) repeat属性

52

第五章 基于时间媒体的表现与同步

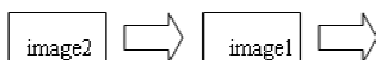
3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明

顺序标记<seq>

<seq></seq>



53

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明

并行标记<par>: 在<par></par>中间的多媒体片断并行播放

54

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明——时间属性

(1) <dur>: 持续时间

(2) begin和end属性: 什么时候开始播放。例:

```
<body><seq dur="5">
  
</seq> </body>
```

image1.jpg所在的组的持续时间为5秒, 而image1.jpg自己要求持续10秒, 实际上image1.jpg的显示时间只有5-2=3秒。

55

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明——时间属性

(3) clip-begin和clip-end属性, 例

```
<video src="test.rm" clip-begin="5s" clip-end="10s"/>
```

clip-begin和clip-end属性是用内部时间控制的属性。这里的内部指的就是多媒体片断自己的时间线(timeline)。前者规定在什么地方开始播放, 后者规定放到什么地方结束播放。

56

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明——时间属性

(3) clip-begin和clip-end属性

例:

```
<par>
  <audio src="testone.rm" clip-begin="5" dur="10s"/>
  <audio src="testtwo.rm" begin="7s" clip-begin="2s" clip-end="15s"/>
</par>
```

首先是testone.rm从自己的5秒处开始播放, 播放7秒以后, testtwo.rm从自己的2秒处开始与testone.rm一起播放; testone.rm播放了10秒, 在自己的15秒处停止播放。testtwo.rm播放到自己得15秒处停止播放, testtwo.rm播放了15-2=13秒。

57

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的同步关系说明——时间属性

(4) fill属性: 当演示中的某个片断播放完成以后, 我们可以用fill属性来规定它的显示状态。Fill属性只有remove和freeze两个取值, 默认的值是remove。

(5) repeat属性: 如果我们希望我们演示中的某个片断重复播放若干次, 可以用repeat属性来实现该效果。

58

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的布局设计——定义显示窗口

口布局标记必须以<layout>开头, 以</layout>结束, 其他具体的标记都在这中间。<layout></layout>必须放在<head></head>之间。

口root-layout表明的是规定最基本的、最底层的窗口。其他一切窗口都在它的基础上划分出来。例:

```
<root-layout width="300" height="200" background-color="black" />
```

width="300"表明窗口宽为300个像素点; height="200"表明窗口高为200个像素点; background-color="black"表明窗口的背景颜色为黑色。

口定义的显示窗口的大小和我们的多媒体片断的尺寸大小不一致时, 可以使用fit属性来控制。fit属性的属性值有hidden、meet、fill、scroll和slice。其中hidden是默认的属性值。

59

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的布局设计——示例

```
<smil> <head> <layout>
  <root-layout width="300" height="300" background-color="yellow" />
  <region id="vedio_region" left="5" top="5" width="290" height="260" />
  <region id="text_region" left="5" top="270" width="290" height="25"
    background-color="white"/>
</layout> </head>
<body><par>
  <vedio src="test.rm" region="vedio_region" />
  <text src="test.txt" region="text_region"/>
</par>
</body>
</smil>
```

60

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的布局设计——层次控制

z-index属性规定相互重叠的窗口的显示次序。数字大显示就在上层。

- ❏ root层窗口总是在最后一层，并且不用z-index属性。
- ❏ z-index属性值可以是负数。
- ❏ 没有重叠的窗口可以使用同一z-index属性值。

61

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的其他特性——交互<a>标记

```
<body>
  <a href="1.rm">
    <video src="videotest.rm" region="videoregion"/>
  </a>
</body>
```

62

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的其他特性——<anchor>标记

1) 分段链接的方法

```
<video src="1.avi" region="videoregion">
  <anchor href="1.jpg" begin="0s" end="10s" />
  <anchor href="videotest.rm" begin="10s" end="20s" />
</video>
```

63

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的其他特性——<anchor>标记

2) 链接部分SMIL

第一个SMIL文件，取名test11.smil

```
<smil><head><layout> ..... </layout></head>
<body>
  <video src="1.avi" region="videoregion">
    <anchor href="part-anchor-test.smil#testlink"/>
  </video>
</body>
</smil>
```

part-anchor-test.smil 片段

```
<body>
  <par id="testlink">
    <video src="....." />
  </par>
</body>
```

64

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的其他特性——语言选择和智能流

```
<body>
  <switch>
    <video src="English.rm" system-language="en-us"/>
    <video src="Chinese.rm" system-language="zh-cn"/>
  </switch>
</body>
```

65

第五章 基于时间媒体的表现与同步

3. 同步关系说明

3.5 基于脚本的表示方法——SMIL

SMIL的其他特性——语言选择和智能流

```
<body>
  <switch>
    <vedio src="highspeed.rm" system-bitrate="250000"/>
    <vedio src="midspeed.rm" system-bitrate="80000"/>
    <vedio src="lowspeed.rm" system-bitrate="20000"/>
  </switch>
</body>
```

66

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

分布环境中的同步远比局部环境中的复杂：

- 同步信息的分布式存储；
- 在表现中涉及的媒体对象的不同存储位置；
- 存储多媒体信息的工作站与表现多媒体信息的工作站之间的通信带来了额外的延迟和抖动；

67

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

4.1 同步关系说明信息的传输

在接收节点端，每一个对象在显示时，都需要对表现部件进行同步说明。传送到接收端的同步信息的传送方式可分为三种：

- 1) 在表现前，将所有的同步信息都发送过去。这种方法经常在人工合成同步中使用。
- 2) 使用额外的同步通道：在只有一个源端的情况下较为有效。在所有的同步信息都不能预先知道的情况下，使用专门的同步通道不会导致额外的延迟。缺点：通信通道的占用；同步通道中的数据通信必须遵守某一时间行为；对来自多个源的同步媒体对象很难处理。

68

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

4.1 同步关系说明信息的传输

在接收节点端，每一个对象在显示时，都需要对表现部件进行同步说明。传送到接收端的同步信息的传送方式可分为三种：

- 3) 多路复合数据流：在一个通用信道上多路复合传送多个数据流，相关的同步信息可以和媒体单元一起传送，不需要同步通道。

将媒体与同步信息多路复合的难点在于：不容易选择到适合于所有相关媒体要求的服务质量。不容易实现多个源的媒体的同步，它必须由流层来支持。混合数据流的使用在一些编码标准中是隐含的，如MPEG，它定义了一个位流来集成视频音频和相关的同步信息。因此，这种位流可以被当成流层的一个流。

69

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

4.2 同步操作的定位

源端或目的地端的处理？

在某些情况下，可以通过把多个媒体对象组合成一个复合对象来实现同步。这样可以减少通信的开销。如果用这种方法，一组相关的对象在发送端就被组合成一个新的视频对象来减少带宽，对象的混合必须由对象层支持。

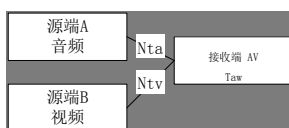
70

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

4.3 时钟同步

在分布系统中，必须考虑发送端与接收端的时钟同步的精度。这一问题在多源同步中尤为重要。例如，在接收端，同步的音频、视频表现应该在 T_{av} 开始，发送端A的音频传送必须在 $T_a = T_{av} - N_{ta} \cdot O_a$ 时刻开始， N_{ta} 为已知的网络延迟。 O_a 为发送端的时钟相对于接收端时钟的偏移值。



O_a 与 O_v 一般不知道。但可以通过一些同步协议来保证差值在某个最大值之间，这样就可以通过缓冲的方式来解决这个问题。

71

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

4.4 表现的处理

在分布环境中，要支持暂停，以不同的速度向前、向后、直接存取、停止重复等功能操作较为困难。

原因是一些必要的信息分布在整个环境中。为表现而预先准备好的对象必须被删除，网络的连接也必须作相应的变化或重建。因此，在执行这些处理功能时不可避免要产生延迟。

72

第五章 基于时间媒体的表现与同步

4. 分布环境中的同步

4.4 表现的处理

首先，选择传送同步关系的说明信息的传输模式。

在运行时，必须决定同步操作的定位、时钟偏差的处理以及组播和广播传输机制的处理。

同步过程中，各个步骤的规划的一致性，还要考虑作用在一些对象上的必要的操作（例如解压缩和压缩的不对称性）。

除此之外，一些与表现有关的操作还要求在运行时重新规划。

73

第五章 基于时间媒体的表现与同步

小结

基本概念：理解基本概念

同步关系说明：掌握常用的同步关系说明方法

分布环境中的同步需要考虑的问题

74