

## 一个基于对象的分布式多媒体系统模型\*

曾庆凯 杨国良 高瀚昭 谢立

(南京大学计算机软件新技术国家重点实验室 南京 210093)

(南京大学计算机科学与技术系 南京 210093)

**摘要** 提出了一个基于对象的分布式多媒体系统模型。由媒体存取层、多媒体对象层和多媒体表现层等构成的层次模型将媒体异构环境存取、分布式多媒体对象处理和语义合成表现等问题在不同层次分别进行处理,并能够全面处理多媒体各种时序关系以及 QoS(quality of service)等特殊问题。

**关键词** 分布式多媒体系统,功能构件,分布式多媒体对象,合成。

**中图法分类号** TP391

随着 Internet 的广泛使用,人们越来越清楚地认识到网络环境多媒体应用的巨大潜力。然而,开发的困难严重影响着分布式多媒体技术的广泛运用。分布式多媒体系统的开发涉及许多比较繁琐的问题,如多媒体 I/O 设备的控制、多媒体信息的处理、网络环境的异构平台的连接等等。为简化应用的开发问题,建立良好的系统模型是一个关键。

以提高软件再利用能力为目标,一些项目致力于分布式多媒体应用的开发环境和编程方法的研究。Medusa 环境的研究提出了基于逻辑模块的机制。<sup>[1]</sup>研究强调模块的软件实现和连接的灵活性。Gibbs 等人提出了一个多媒体处理的框架<sup>[2]</sup>,其中包括构件类及其连接、合成的概念。MADE (multimedia application development environment) 研究计划目标在于开发多媒体应用的开发环境。<sup>[3]</sup>它将处理模型抽象为各种媒体对象和处理媒体对象间关系的复合对象。Bates 等人开发了多媒体应用开发平台。<sup>[4]</sup>他们设计了抽象的数据对象模型、媒体播放类和各种播放服务,其中提出了类似连接机制的播放结构。Pinto 等人提出了一个包括构筑分布式多媒体应用的模型、语言以及运行环境的框架。<sup>[5]</sup>这个模型基于参考点同步描述。

我们认为研究有以下不足:(1) 内容主要分布于两个方面:多媒体处理的基本形式<sup>[1,2]</sup>和多媒体应用的构成。<sup>[3-5]</sup>而将两个方面结合起来的统一研究不够。(2) 研究比较注重处理形式的抽象,而结合分布式多媒体处理特性,如分布、同步及 QoS 等问题的研究不够。

本文提出了一个基于对象的分布式多媒体系统的层次模型 FDM(fenbu duomeiti model),描述了模型的对象化实现方法及其应用实例。模型具有如下特点:

(1) 功能模块化。由于时序媒体处理的特点,其处理功能的基本单元是相对固定的。随着技术的进步,各种媒体处理技术的研究越来越深入,同时也越来越专门化。因此,功能模块化符合多媒体处理的要求和发展趋势。

(2) 结构构件化。虽然各种时序媒体的处理技术千差万别,但处理形式是一致的,都表现为从媒体数据源至表现目的地的流结构。构成这个流结构的节点是构件化的功能模块(即构件)。构件通过连接机制构成基本控制流结构。

(3) 编程对象化。对象化方法的模块化、封装性、继承性等优点适应于多媒体处理的要求。多媒体应用以多媒体对象作为多媒体表现合成的基本单元。

\* 本文研究得到高等学校博士学科点专项科研基金和江苏省应用基础项目基金资助。作者曾庆凯,1963年生,副教授,主要研究领域为分布式多媒体系统。杨国良,1974年生,硕士,主要研究领域为分布式多媒体系统。高瀚昭,1973年生,硕士,主要研究领域为分布式多媒体系统。谢立,1942年生,教授,博士生导师,主要研究领域为分布计算与并行处理。

本文通讯联系人:曾庆凯,南京 210093,南京大学计算机软件新技术国家重点实验室

本文 1997-11-24 收到原稿,1998-01-23 收到修改稿

多媒体包括静态、时序等多种媒体。在本文中,多媒体通常是指时序媒体。

### 1 系统模型

#### 1.1 系统结构

我们认为,系统模型应该是多层次的,层次模型适应于解决:

(1) 多层次的多媒体时序关系。时序关系是多媒体的最重要特性之一。将所有的时序关系同样处理,不利于问题的解决。我们把媒体之间的关系分为紧密的同步关系和松散的合成关系。同步关系包括连续同步,是指在表现过程中多媒体之间必须进行相互通信的紧密时序关系,合成关系指较为松散的时序关系;只在起始和结束时具有约束的同步关系;在表现过程中,媒体之间可以独立进行的时序关系。这样定义的目的在于将两类时序关系分层次处理。紧密的同步关系处理成多媒体对象内部媒体之间的关系,松散的合成关系则在多媒体对象基础上进行处理。

应用层
多媒体表现层
分布式多媒体对象层
媒体存取层
基本系统层

(2) 多层次的分布式多媒体处理问题。在分布式多媒体系统中需要解决各种环境的多媒体存取、分布资源的分布存取与控制以及多媒体语义表现等问题。多层次模型可以将这些问题分层分解,在不同层次分别处理。

图1 系统结构示意图

图1为分布式多媒体系统的层次结构,由基本系统、媒体存取、分布式多媒体对象、多媒体表现以及应用等层次构成。

#### 1.2 系统功能

基本系统层对应于操作系统和通信子系统,提供了对系统资源的管理和使用的功能,通常包括对设备、文件、存储器和处理器等硬件资源的存取控制以及数据通信等服务。系统层的服务是通用的,没有多媒体语义。

媒体存取层实现媒体处理所需的各种基本功能,提供各种媒体特定语义的存取服务。例如,声音、声频等媒体数据的采集、播放、压缩、存储及传输等功能。这一层利用系统层的服务对物理资源进行存取,各种媒体的输入/输出设备、数据组织形式以及处理方式等均不相同,因此,通过规范各种媒体数据的组织格式、处理功能等调用要素,以取得服务功能调用接口的一致,从而保证上层服务功能与具体环境无关。本层不仅实现媒体处理的基本模块,而且提供模块级的 QoS 调节以及媒体数据的时间标记等控制功能。

多媒体对象层完成分布式多媒体处理功能的对象化封装。利用媒体存取层提供的基本处理功能,可以实现多媒体资源的分布式存取服务。分布式多媒体对象服务不仅提供多媒体对象的外部操作接口,而且实现对象内各种媒体之间的同步时序关系控制以及 QoS 协商控制等功能,为上层的多媒体合成表现提供支持。

多媒体表现层为分布式多媒体应用提供多媒体表现的合成机制;以多媒体对象层实现的分布式多媒体对象服务为基础,按照应用的语义要求进行合成表现。在此,合成关系主要是指时序关系,并分为对象合成和界面交互两类来实现。

应用层提供分布式多媒体处理的应用服务。它利用下层实现的服务功能来利用系统的多媒体资源。利用中间服务层(指媒体存取、分布式多媒体对象及多媒体表现层)提供的分布式多媒体服务功能,可以构成各种应用系统或模块。

### 2 模型的对象化设计

系统层通常由计算机的基本系统提供。以上的中间服务层以对象类 API 提供对分布式多媒体应用的开发支持。本节介绍三层的对象化实现方法。

#### 2.1 媒体存取层

本层实现的功能与要求包括:

(1) 媒体基本处理功能。媒体基本存取功能是本层的主要目标。媒体的各种处理功能以功能模块的形式来实现。因此,对象类库是一组媒体处理功能模块的集合。

(2) 媒体的逻辑存取要求。本层利用系统层的服务处理各种物理设备资源的存取控制,并为上层提供与具

体环境无关的媒体存取功能接口。利用面向对象方法的继承性,可以在抽象接口与具体实现之间取得协调一致。

(3) 组件化要求。基本功能模块应该可以方便地结合起来,构成应用要求的更大的功能模块。因此,构件方法及其连接机制<sup>[2]</sup>符合这种要求。但是,有必要加入和强化同步处理和 QoS 调节的功能和机制。

为了方便对基本功能模块的调用和控制,应该提供如下功能:

- (a) 运行状态。包括处理速度、位置、QoS 等参数的查询功能。
- (b) 媒体时间标记功能。向调用者提示媒体运行位置。
- (c) QoS 调节功能。QoS 状态以及协调的方法。
- (d) 媒体流调节功能。包括速度、位置的调整功能。

图 2 为功能构件的结构示意图。 $F$  为模块的处理功能,例如,媒体的输入、压缩、存储等。 $B_i$  和  $B_o$  为与其他模块连接的外部缓冲区。通过与其他模块的连接形成媒体处理的流结构。 $IF$  为模块与调用方的接口,接收调用方的命令  $E_c$ ,如运行模式改变、调整等命令;也向调用方发出同步标记  $E_s$  以及 QoS 溢出  $E_q$  等事件。 $S$  为模块运行的状态和模式参数。 $R$  是处理功能相关联的物理资源。

功能模块实现为功能构件,包括输入、输出及传输三种类型。

(a) 输入类。输入构件可作为流结构的源头,产生时序媒体数据。它通常从硬件设备或媒体文件中读取数据,具有输出端口,向可能的后续构件提供数据。输入构件包括实时的声音输入、视频输入以及存储型媒体输入等构件。

(b) 输出类。输出构件可作为流结构的目的地。媒体数据到达输出构件后被消耗。与输入构件类似,它也通常与硬件设备或文件对应,具有至少一个输入端口以读取处理的媒体数据。典型的输出构件包括声音输出、窗口图像输出以及文件保存构件。

(c) 传输类。传输构件位于流结构的中段,负责对媒体数据进行处理、传递。传输构件负责媒体的数据格式、编码、组织形式等处理。典型的构件包括视频、音频的压缩与解压缩、格式变换、分解和合并等处理构件。

功能构件类的层次如下:

*Mmcomp*

*Mmcomp\_in*

*Mmcomp\_in\_live*

*Mmcomp\_in\_live\_video/audio*

*Mmcomp\_in\_stored*

*Mmcomp\_in\_stored\_video/audio*

*Mmcomp\_out*

*Mmcomp\_out\_video/audio*

*Mmcomp\_trans*

*Mmcomp\_trans\_compress/uncompress*

*Mmcomp\_trans\_send/receive*

...

以下是存储型视频输入构件的 API 说明示例。

```
class Mmcomp_in_stored_video extends Mmcomp_in_stored {
    Mmcomp_in_stored_video(R, Es, Q, Bo);    finalize(); readstart(); readstop();
    setposition(); getposition(); setQos(); getQos(); ... }
```

其中  $R$  为与功能构件实例化相关的多媒体资源;  $E_s$  为构件为同步控制需要的同步事件;  $Q$  参数是构件工作的 QoS 要求;  $B_o$  为构件实现时的输出端口。

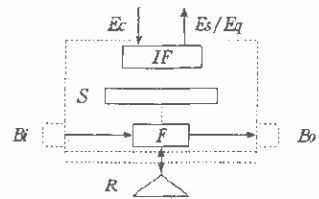


图2 功能构件结构示意图

### 2.2 多媒体对象层

对象层实现分布环境的多媒体对象存取功能,它的主要目标包括:

(1) 对象化操作.多媒体资源的对象式存取,利用媒体存取层支持的媒体处理功能,实现多媒体存取的对象化封装,使对媒体的处理粒度单位从媒体数据提高到多媒体对象.

(2) 位置无关的分布媒体存取.媒体存取层提供的媒体处理功能基本上是局部的,对象层实现分布的对象存取,使多媒体存取与媒体资源位置无关.

(3) 多个媒体源的同步存取.多媒体对象可以包含多种媒体的多个媒体源.这主要是为了处理具有密切时序关系的多种媒体的同步控制,这样既可以方便处理媒体间的同步通信,又保持了多媒体对象的完整性.

从实现的角度看,主要的功能包括:

(a) 对象的接口处理.实现多媒体对象操作的处理方法,包括播放、停止、抓取、进退等操作.

(b) 媒体流结构的维护.为实现多媒体对象的功能,需要对一组功能构件进行组织和管理.通过协调构件间的工作来实现对象的功能.因此,需要提供构件的建立、撤销、连接等操作.

(c) 多媒体的同步控制.同步关系以参考点方法描述.按照媒体的运行状态,对多个媒体源的运行进行调度控制,以取得对象内的多媒体同步.

(d) QoS 协商和控制.一组符合要求的功能构件通过连接构成一个从媒体源至目的端的流结构.构成流结构的多个功能构件应该在以一致的 QoS 参数工作时才能取得良好的效果.

多媒体对象的实现结构如图 3 所示,IF 为对象的外部接口,接收多媒体对象的操作命令  $Op$ ,发出对象的事件  $Eo$ (多媒体对象的始终标记).S 为对象内部的状态和运行参数,包括结构控制参数、同步信息以及 QoS 参数.  $St$  结构控制模块,负责构成多媒体流结构的构件维护.  $Sy$  为同步控制模块,监视构件的同步标记,按照同步时序关系进行同步调整与控制.  $Q$  为 QoS 控制模块,进行 QoS 状态监视和协调:将对对象级的 QoS 要求映射到各种功能构件的参数,如媒体输入的采样率、分辨率、压缩比、播放窗口大小等等.当较高级的 QoS 不能满足时,进行协商下调.  $COs$  为实现多媒体对象功能所调用的功能构件集合.

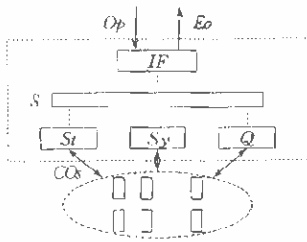


图 3 多媒体对象实现结构示意图

多媒体对象的对象层层次如下:

- MmOBJ
- MmOBJ-live
- MmOBJ-stored

存储型多媒体对象类的说明示意如下.

```
class MmOBJ-stored extends MmOBJ {
    MmOBJ-stored(Str, Qos, SynR); finalize(); play(); stop();
    pause(); resume(); forward(); backward(); position(); ... }

```

其中  $Str$  是多媒体对象的结构信息;  $Qos$  为多媒体对象建立时的 QoS 参数要求;  $SynR$  为多媒体对象内部的多种媒体间同步关系.

### 2.3 多媒体表现层

多媒体表现层处理较为松散的多媒体合成表现,处理多个多媒体对象之间的关系.合成表现的功能包括:

(1) 多媒体对象合成.合成功能处理多媒体表现中的静态语义.它反映多媒体作品的创作语义,是相对静止的,由多媒体对象之间的关系表示.合成关系包括时序关系和空间关系.空间关系可以通过对象表现的位置属性来确定.因此,此处的对象合成主要是指时序合成.

(2) 交互功能.交互功能处理多媒体表现中的动态语义,体现读者的表现语义要求,是随机动态产生的.交互功能涉及多媒体对象和界面对象的互动关系.

多媒体对象合成实现的功能包括:

(a) 合成时序控制.多媒体对象按照相互之间的时序关系进行合成表现.

(b) 对象表现处理.包括对象的变换、片段内容选取等处理.

(c) 表现合成关系的复合对象的操作接口.复合对象作为多媒体单元,具有多媒体单元的特性.可以进行表

现的启停、进退等控制操作。

图 4 为合成对象的结构示意。IF 为复合对象的操作接口,接收操作命令  $Op$ ,发出对象事件  $Eo$ 。TR 为合成时序关系表。T 为计时器,按照时间关系 TR,为合成模块 CM 提供计时时间功能。CM 负责对象合成的控制协调工作。Os 为参加合成的对象集合。合成对象类说明示意如下:

```
class Mmcomposite {Mmcomposite(TR);finalize();piay();pause();
    forward();backward();position();...}
```

TR 为多媒体对象合成关系,以时间轴方法来表示。

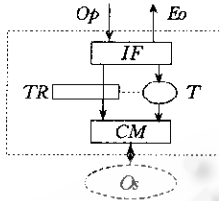


图4 合成对象结构示意图

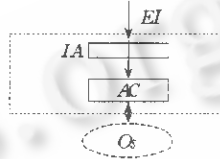


图5 交互对象结构示意图

交互功能处理界面对象与多媒体对象之间的交互关系。具体表现为界面交互事件与多媒体对象操作之间的关系;界面事件对多媒体动作的驱动。

交互对象的结构如图 5 所示。EI 为界面对象产生的交互事件。IA 为交互事件的反应规则。AC 为反应执行的控制模块。按照 IA 定义的交互驱动方法进行交互操作。Os 为参加交互动作的对象集合。交互对象类说明示意如下:

```
class MMinteractive {MMinteractive(IA);finalize();...}
```

其中 IA 为交互关系,由界面事件与多媒体对象的操作方法来表示。

### 3 实例分析

图 6 是一个视频场景框的示意图。它是许多分布式多媒体系统,如视频会议等的核心功能。视频场景框包含 4 个远程视频窗口。每个窗口对应于一台视频输入设备。利用本模型的开发支持类库 API,视频场景框可以用一个容器合成对象来实现。每个视频窗口实现为一个视频对象:

```
video1=new MmOBJ_live([host1]camera1,SynRT,QoS);
video2=new MmOBJ_live([host2]camera2,SynRT,QoS);
...
vedioBox=new MmContainer(video1,video2,video3,video4);
```

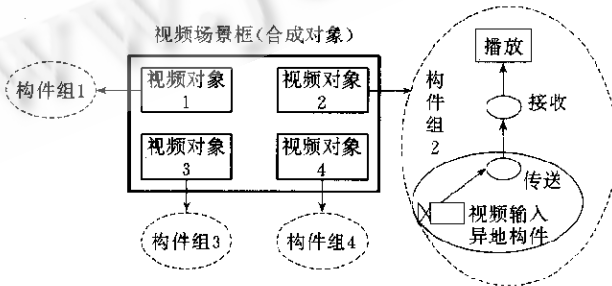


图6 场景视频框实现示意图

基于本模型,应用编程以分布式多媒体对象以及合成机制为基础,不必涉及底层较复杂的编程工作。开发的起点较高,使用比较方便。文献[1,2]中采用的模块和构件方法具有较好的软件再利用能力,但缺乏规范的应用构成方法。应用开发时,开发者需要对模块或构件的创建、撤销、连接和运行等过程进行控制和管理,而且必须

解决同步控制、分布处理等问题,因此,使用有一定难度.文献[3~5]以对象为基础提供应用开发方法,但是对象的实现缺乏灵活性和适应性.本层次模型则可弥补这方面不足.模型还考虑了 QoS 调节机制,可以从功能构件、多媒体对象等不同层次上进行调节.

#### 4 结束语

系统实现采用 JAVA 作为编程语言.开发基于 PVM 环境并扩充了多媒体支持功能.目前系统实现正在进展中.

#### 参考文献

- 1 Wray S *et al.* Networked multimedia: the medusa environment. IEEE Multimedia, 1996, 3(4):55~63
- 2 Gibbs S J, Tsichritzis D C. Multimedia Programming. Reading, MA: Addison-Wesley Publishing Company, 1995
- 3 Herman I, Reynolds G J, Davy J. MADE: a multimedia application development environment. In: Kavanaugh M E ed. Proceedings of the International Conference'94 on Multimedia Computing and Systems. Los Alamitos, CA: IEEE Computer Society Press, 1994. 185~193
- 4 Bates J, Bacon J. A development platform for multimedia applications in a distributed ATM network environment. In: Kavanaugh M E ed. Proceedings of the International Conference'94 on Multimedia Computing and Systems. Los Alamitos, CA: IEEE Computer Society Press, 1994. 154~163
- 5 Pinto P F, Inington P F. A language of the specification of interactive and distributed multimedia applications. In: Meer J D, Mahr B, Storp S eds. Open Distributed Processing (II). North-Holland: Elsevier Science B. V., 1994. 247~264

### An Object-oriented Model for Distributed Multimedia Systems

ZENG Qing-kai YANG Guo-liang GAO Han-zhao XIE Li

(State Key Laboratory for Novel Software Technology Nanjing University Nanjing 210093)

(Department of Computer Science and Technology Nanjing University Nanjing 210093)

**Abstract** In this paper, a model for distributed multimedia systems based on objects is presented. The layered model includes the component, the multimedia object and the representation layer on which respectively provides media access in heterogeneous platforms, distributed multimedia object process and semantics composition services. It is feasible to deal with the special problems in distributed multimedia systems such as synchronization and QoS (quality of service).

**Key words** Distributed multimedia system, component, distributed multimedia object, composite.