# 第五章 多媒体软件开发环境

# 5.1 多媒体数据获取

## 多媒体应用软件的开发过程

## 明确使用对象, 了解用户需求;

## 选择开发方法;

## 准备多媒体数据;

## 完成系统集成。

## 多媒体节目特点是以内容情节为导向的, 而其它的软件如文本处理软件、数据库管理系统等一般是工具性的软件。

## 用户需求往往是针对多媒体技术从内容和设备配置方面具体要求, 如

## 用户是否有不使用鼠标和键盘而直接通过触摸屏幕来获取信息的要求；

## 系统中是否需要语音和音乐；

## 数据类型中有无图象、视频、动画、字幕的要求；

## 等等。

# 

## 有两种方法可供选择：

## (1)由开发人员编码来实现多媒体节目；

## (2)利用多媒体开发工具来制作多媒体节目。

## 前者的优点是不需较大的投资, 但需编制大量的程序，维护也不方便; 后者需要一定的投资，但开发周期短，维护问题少，关键是要选择一种功能较强价格合理的工具软件。

# 

## 当开发方法确定后,就进入了具体实施阶段。实施阶段的基本工作是多媒体数据的准备。

## 多媒体节目里包括音频、视频、动画、静态图象、文字、图形等多种媒体素材, 这些素材在系统集成之前必须准备好。

# 

## 制作者通过所选择的开发方法将节目情节具体化程序化, 并将准备好的多媒体素材按照需要进行编辑加工, 最终集成为一个由程序和数据组成的软件产品。

## 开发的节目往往又记录在某种介质(如CD-ROM)中，便于销售和使用。

# 多媒体数据获取方法

## 多媒体数据的准备又包括数据的获取、数据的整理、数据的编辑加工等阶段。

## **数据的整理**主要是对采集的媒体数据按照指定的方法进行登记与分类，便于后面步骤的使用。

## **数据的编辑加工**指的是根据情节的要求对媒体数据进行剪辑、修改、格式转换等处理，一般通过专门的软件工具来完成.

# 图象数据获取方法

## 购买数字化的图象或图片；

## 自己动手使用特定软件创作电子图象；

## 用扫描仪将照片、图片做数字化处理；

## 用摄像机或帧捕捉器捕捉视频画面并进行数字化处理；

## 向有关单位无偿或有偿交换拷贝或通过网络获取公开的图象文件。

# 音频数据获取方法

## 最常用的方法是利用录制设备录制音源,然后数字化处理并存入计算机。

## 另一种方法就是购买或和有关部门交换音频数据文件,如大型图书馆、电台等部门存放有珍贵的名人讲话原始录音,制作者可通过合适途径获取。

# 文本和数据文件的获取方法

## 人工输入。这是最自然、最易掌握的普遍采用的方法。英文输入方法比较简单，中文输入方法较为复杂，流行的有几十种方法。

## OCR扫描识别。通过扫描原稿输入文字数据图表技术已经成熟，市面上有多种产品问世。目前印刷体文本的识别率可达99%以上，手写体文本识别率一般也能达到75%以上。

## 购买和交换文本

# 图形和动画的获取方法

## 图形和动画的视觉效果是由人类创作设计的虚幻的或仿真的画面所表现的，它表达了人类对空间和物体的主观认识。

## 图形和动画的获取以用户制作为主要途径，有时也可通过购买或交换得到 。

# 音频编辑软件

## 若要在多媒体节目中加入完整的音频,则必须有赖于编辑声音的软件及PC上加装音频卡。

## 音频编辑软件一般包括下列各种工具或功能：

### (1)菜单条

### (2)控制板

### (3)显示板

### (4)剪辑板

# 数字音频编辑工具

## Microsoft提供的“录音机”, 使用它可以录制、播放、编辑Microsoft的波形数字音频

## 两种。

# 图象处理软件

## Photoshop可完成简单的亮度和色彩校正以及复杂的图象编辑,例如图象扫描、校色、滤色、润色、混色、特殊效果及颜色分离;

## 支持大量的流行的图象文件格式, 如TIFF、TGA、PCX、GIF、BMP等格式, 并很容易文件格式转换；

## 支持RGB、CMYK、HLS多种彩色模式及灰度图象。

# PhotoShop的功能

## (1) 扫描图象

## (2) 选择和蒙片

## (3) 图象编辑、转换和特殊效果

## (4) 图象合成

## (5) 图象校正和加强

## (6) 绘图和润色工具

## (7) 图象处理过滤器

## (8) 颜色分离打印

# 5.2 图形和动画的制作

## 图形数据

## 计算机图形学是研究怎样用计算机生成、处理和显示图形的一门学科。它着重讨论怎样将数据和几何模型变成可视的图象; 或对自然界已存在对象通过获取相关数据建立几何模型生成的图象, 以便于分析处理。

## 图形的显示形式也称为图象, 但和一般意义的图象处理技术不同, 后者侧重于将客观世界中原来存在的物体映象处理成新的数字化图象, 关心问题是如何滤掉噪声、压缩数据、提取特征、三维重建等内容。

## 在多媒体数据中,无法从客观世界直接摄取的可视信息,就可用图形技术来制作,这些数据主要包括文字、图形、动画。

## 图形包括二维空间及三维空间图形, 其中二维图形仅能表现图形中各部分简单的位置关系

## 三维图形经真实感处理, 将使图形能表达出空间、位置、材质、明暗等接近自然的真实感效果。

## 动画是图形对象赋予运动属性后制作的连续画面效果,需要专门的软件工具制作。

# 图形数据

## 图形文件的格式通过图形原语和它们的属性来描述。图形本身决定了哪些原语被支持,诸如线型、线宽、颜色之类的属性影响着图形画面的输出。

## 这种表示需要在图形显示时被转换成低级图象的点阵。高级原语的优点是减少了存储每幅图象需要的数据,容易操纵图形图象,缺点是需要更多的步骤来将图形原语及其属性转换成它的图象象素。

## 对图形软件开发产生广泛影响的标准有PHIGS，GKS和OPEN GL等 。

# 计算机动画

## 计算机动画(Computer Animation)是用计算机生成一系列可供实时演播的连续画面的技术,它可把人们的视觉引向一些客观不存在或做不到东西,并从中得到享受。

## 计算机动画是使用计算机作为工具来产生动画的技术,计算机在动画制作过程中起着大量的不同的作用,表现在画面创建、着色、录制、特技剪辑、后期制作等各个环节。

# 历史与发展

## 1831年, 法国人Joseph Antoine Plateau利用视觉滞留原理发明了一种称之为“诡盘”的机器创造了运动画面的幻觉。第一部动画片是1906年由美国人J. Steward Blackton 制作的, 名字叫“Humorous Phases of Funny Face” 。

## 1909年美国制作第一部卡通片; 1915年引入Cel动画技术; 1928-1938,Disney时代,米老鼠、唐老鸭风靡一时;

## 1963年,BELL实验室制作第一部动画片;

## 1960年代, 2维计算机动画发展时期。

## 1970年代, 动画片屡屡获奖; 三维动画系统是研究重心。

## 1980年代后期, 随着以SGI为代表的高性能图形工作站出现以及图形学理论的飞速发展，推出了一批可生成具有高度真实感的实用化、商品化的三维动画系统。

## 1990年代后,计算机动画在影视制作方面取得辉煌成就。

# 

## 我国的计算机动画的研究与应用虽然起步较晚，举办1990年亚运会时,开始有所起步;

## 90年代发展很快,一些大学和科研 单位相继开展计算机动画的研究工作,缩短了与国际上的差距，有些领域的工作在国际上产生一定的影响。

## 在应用方面,以电视片头、卡通片、动画广告制作为重点,在社会上产生了动画热。

## 目前我国的动画研究水平逐步缩短了与国际差距,越来越多的学者活跃于国际学术界.

# 计算机动画研究内容

## 运动控制方法

## 动画描述模型与动画语言

## 中间画面的生成技术

## 三维动画中的物体造型技术

## 动画绘制技术

# 运动控制方法

## 计算机动画中用于控制动画物体随时间而运动或变化的运动控制模型主要有

#### 运动学方法；(起点,终点,时间)

#### 物理推导方法；(受力分析)

#### 随机方法；(分形技术,粒子系统)

#### 行为规则方法；(角色分类,行为规则)

#### 自动运动控制方法等。(人工智能,机器人)

# 运动控制方法

## 粒子系统动画

## 分形动画

# 动画描述模型与动画语言

## 用户和动画制作系统的交互方式是评价动画制作系统的重要因素之一,这种交互方式的抽象层次和自然语言化程度主要依赖于动画描述模型的影响。

## 对动画描述较有影响的描述模型有面向对象方法,角色理论,记号系统,时间轴描述,基于时序算子的描述,基于知识的描述等.

## 基于动画描述模型开发的动画描述语言主要有3类: 记号语言(动作编码);通用语言(扩展的流行语言);图形语言(专门设计)

# 中间画面的生成技术

## 动画的中间画面的生成主要有3种途径：

### 关键帧方法；

### 算法生成；

### 基于物理的动画生成。

# 三维动画中的物体造型技术

## 动画中物体表示可分为3个层次

## 线框 物体由一系列线框表示。

## 表面 物体由一系列面素表示。

## 体 物体看作一系列体素组成或看作3D空间包围部分。

## 曲面造型 代数曲面;拟合曲面(Bezier, B样条, NURBS曲面)

## 实体造型 边界表示,CSG,推移,八叉树,单元分解

## 人体造型 骨架-肌肉-皮肤

## 人体动画 造型和运动控制十分困难,计算机动画技术的最大挑战

# 动画绘制技术

## 真实感图形绘制技术是计算机图形学研究的一个重要内容,人们已经提出了许多光照模型和绘制算法, 其中有代表性的常用的光照模型有Phong模型, Cook-Torrance模型, Whitted模型等。

## 绘制技术有扫描线算法,Phong明暗处理算法, 光线跟踪技术,辐射度技术等。动画绘制除了使用上述方法外, 还可根据其目标是生成一系列连续画面图象的特点, 利用相关性来加速绘制过程。

# 计算机动画的应用

## **面向影视制作的应用**不强调画面的真实性,只追求观赏性和趣味性,其中角色的运动可以有些虚幻,但绘制技术要求较高,能模拟出各种真实感效果。

## **面向模拟的应用**着眼于各种真实问题的仿真研究,它追求数据的正确性和结果的可信性以及能使各种以前仅能得到大批数据的科学试验可视化,这类动画对绘制效果没有前者要求的高。

# 计算机动画的应用

## (1) 影视制作

## (2) 广告制作

## (3) 教育领域中的辅助教学

## (4) 科研领域

## (5) 工业领域

## (6) 视觉模拟

## (7) 娱乐工业

# 三维动画制作软件3DS max

## 3DS max是美国Autodesk公司所推出的在PC机上运行的实体设计及动画制作软件, 提供给专业绘图人员建立高质量图象或制作动画所需功能。

## 3DS max提供用户组合各种色彩的设置、各种透明度的控制、表面纹理以及各种反射的特性, 以帮助产生任何可以想象出来的材质。

## 3DS max由5个模块组成: 二维造型, 三维放样, 三维编辑, 材质编辑, 关键帧编辑。

# 动画制作过程

## 用二维造型(2D Shaper)绘制各种平面几何图形

## 在三维放样(3D Lofter)中将平面几何沿着给定路径放样成三维立体形体, 并转入三维编辑(3D Editor)

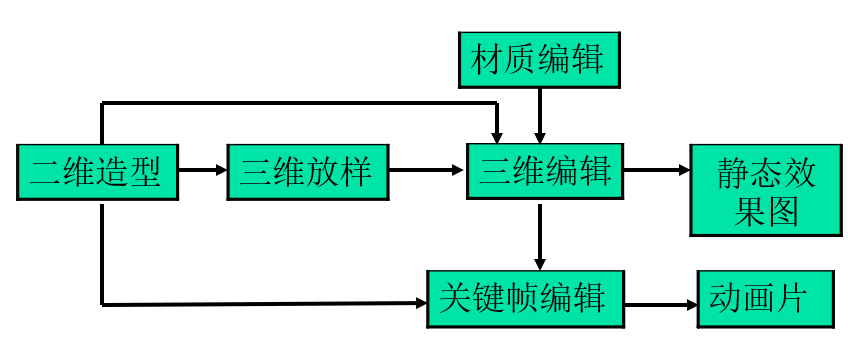
## 再在三维编辑器中, 对物体所在的场景进行各种设置和调整, 包括设置场景中的光源和摄象机。

## 材质编辑(Material Editor)的作用是制作物体所需的各种表面材质和纹理质感。

## 在关键帧编辑(Keyframer)中, 可以对三维编辑器中制作的三维场景设置关键帧, 以定义各种物体的运动轨迹。

## 真正的动画效果还需要绘制过程才能体现

# 3DS max各模块之间关系



## 动画创意

## 动画是科学与艺术相结合的产物,又称之为计算机艺术,好的动画作品,需要完美的艺术构思和创造性设计,这就是动画的创意。

## 动画创意至少需要下列3方面知识：

### 设计者必须具备一定的艺术修养和创造能力；

### 设计者必须知道计算机能否实现它,实现的复杂程度如何；

### 设计者必须具有大量的计算机动画制作经验。

# 

## 二维造型

## 创作各种二维的线条和图形, 这种平面造型可由三维放样模块沿着一条路径扩展长成三维空间的对象,然后这些对象可由三维编辑器编辑。

## 二维造型可用来设计某个对象运动的路径,交关键帧编辑器从而形成立体对象的动画。

## 在进行三维放样前,需保证二维图样是闭合且无重叠交叉,其合法性可用Shape/check来检查。

# 

## 三维放样

## 工作步骤如下：

### ①读入二维造型所建立的造型；

### ②赋予读入造型一条在二维空间里的放样路径；

### ③将二维造型沿着路径增加厚度放样成一个三维的对象。

### 所产生的对象会自动由三维编辑器处理成三维的网状对象。造形和路径共同组成一个所谓的模型。在三维编辑里用户可以直接编辑这个对象,在对象表面上赋予材质,或指定光源, 将对象着色。

## 三维编辑

## 用户使用这个模块创造、排列、着色一个三维空间的场景。也可以建立及调整灯光作特殊效果, 并使用材质编辑创造的材质以增加真实感。

## 三维编辑和三维放样模块都提供6种观察方式(Front, Back, Top, Bottom, Left, Right)来观察正在建立的三维对象; 此外, 它还提供了摄象机观察方式, 用户可以通过此方式来进行对象材质的赋予及着色等动作。

# 

## 关键帧编辑

## 使用关键帧编辑器来制作动画的过程为：

## 先定义整个动画是包含那几个关键帧;

## 然后指定这些帧中每个元素位置与状态,该编辑器会计算出画面之间移动的过程以及状态的变化，以自动产生关键帧间的画面,使动画过程平滑和缓。

## 材质编辑器

## 主要给用户提供不同方式的贴图功能,所谓贴图是指可以赋予对象表面以图象效果。

## 用户可以建立以下种类贴图：

### ①纹理贴图，其效果就象将图象直接画到对象表面一样。

### ②透明度贴图，用来设置对象表面的透明程度，贴面里颜色越深的地方透明程度越高，越浅则越不透明。

### ③反射贴图,用来制造出图象从对象表面反射出来的效果。这些可通过材质编辑器调整物体材质的属性参数来实现。

# 5.3 多媒体编著工具

## 多媒体创作工具(或称编著工具)是一种高级的软件程序或命令集合。

## 创作工具旨在提供给设计者一个自动产生多媒体节目的综合环境, 使设计者可将不同的内容与各种功能结合在一起,形成一个结构完整的节目。

## 多媒体创作工具通常应包括制作、编辑、输入输出各种形式的数据,以及将各种数据组合成为一个连续性序列的基本工作环境 。

# 

## 多媒体创作工具分类

## (1) 以卡或页为基础的创作工具

## (2) 以图符为基础,基于事件的创作工具

## (3) 以时间为基础的创作工具

## (4) 以传统程序语言为基础的创作工具

## (5) 其它专用的创作工具

# 5.3.2 以卡或页为基础的多媒体创作工具

## 大多数以卡或页为基础的创作工具提供一种可以将对象连接于卡或页的环境。

## 一页或一张卡便是数据结构中的一个节点，它类似于书的一页或数据袋里一张卡片。

## 这种页或卡片上的数据比书上的一页或数据袋里一张卡片的数据更多样化。

## 在卡或页上的图符很容易理解和使用

## 以Asymetrix公司开发的ToolBook为例介绍这类软件。

# ToolBook简介

## ToolBook是一个面向对象开发环境

## 提供一种面向对象的程序设计语言OPENSCRIPT。

## 从使用观点来看,ToolBook分两个层次：读者和作者。在读者层次用户可以执行该书,阅览其内容。在作者层次上,设计者可以使用命令来编写新的书。

## ToolBook采用Windows用户接口。也采用下拉菜单、图符驱动、放弃和剪贴板功能、综合帮助功能,设置操作约定。

## ToolBook支持大量流行的文件格式,方便数据交换。

# 

## **设计编程制作一体化环境**

## ToolBook具有把图形、文字、数字视频图象、声音及动画集成为一个交互式节目的能力。

## 它提供了高级脚本语言OPENSCRIPT,配置了许多命令去播放各种类型的媒体,管理各种数据以便改变对象的性质。

## 擅长于制作把其它Windows应用软件集成在一起的多媒体节目,还可在媒体单元之间建立链接关系。

## ToolBook的书形隐喻符很容易使人理解,因此它缩短了用户开发节目之前学习创作工具的时间。

# 节目设计思想

## ToolBook按书的结构组织应用程序。

## ToolBook电子书的每屏被描述为一页,每页内可有多级的对象,它们分为背景和前景,其中背景的设置是满足用户要将生成的一系列页共享一些通用元素的要求,如一幅图象或象NEXT、QUIT这样的命令按钮。

## 开发电子书的过程是：

## 在屏幕上画出各种各样的对象, 然后生成潜在的“脚本”, 它在一给定对象以某种方式被选中或触发时, 引发一个或多个结果。这些脚本事实上是用OPENSCRIPT语言写的小段程序 。

# 

## 脚本特点综述如下:

### (1)脚本是一系列OPENSCRIPT语句或指令,它们告诉对象要做些什么。

### (2)脚本可分为一些处理单元，它们描述特定文件出现时,如读者触发按钮或按某个键,将会发生的事件。

### (3)脚本可以控制对象也可以控制信息。

### (4)页面上对象如字段、按钮以及图形的脚本往往对该页面或者同一本书中的一个页面发生影响。

### (5)书、页面以及背景和页面上的对象一样也可以有脚本。

# 5.3.3 基于图符和事件的创作工具

## 基于图符的创作工具提供可视化的程序设计环境。

## 设计之初须先用其他软件来制作各种元素;

## 然后在此系统中建立一个流程图,在流程图当中可以包括起始事件、分支、处理及结束等各种图符;

## 设计者可依流程图将适当的对象从所谓的图符库按下拉至工作区内。这些图符可以包括菜单条的选项、图形、图象、声音及运算等;

## 这个流程图也是事先安排的次序, 同时也表示整个节目的逻辑蓝图;

## 这类创作工具最典型是Authorware。

# Authorware功能特点

## Authorware是一个交互式多媒体节目创作工具, 它使用图符设计流程图,无需编程,非常方便使用。

## Authorware可以用流程图来当作导航图,设计者只要将图符用鼠标按下拉至流程图的某个位置上, 便可以使每一个环节相互连接。

## 变量可以互相传输参数。Authorware提供了200个以上的系统变量及功能来决定属性、数据抓取、对象处理及显示等工作, 甚至控制作业流程的分支, 跳画面及循环等效果。

## Authorware 最大特点是使用15个图符组成的界面 。

# 5.3.4 以时间为基础的创作工具

## 常见的一种多媒体编辑系统,常用于制作电影与卡通片的节目。

## 大多是以时间轴来决定事件的顺序与对象显示上演的时段。

## 这种时间关系可以许多频道形式出现,以便安排多种对象同时呈现。

## 这类系统中都会有一个控制播出的控制面板,它很象录音机、录放像机的控制板, 含有倒带、倒退、停止、演出及快进等按钮。

## 如Action!

# Action!

## 由Macro Media公司所发行,可在Windows与Macintosh下执行的多媒体编辑创作工具。

## 结合了动作、声音、文字、图形、动画多媒体显示环境,使用时间轴来组织其元素。

## 使用时间轴及控制面板来组织一个场景。

## 有一个内容表可以显示出一节目的全部场景,以及每一个场景当中的全部对象。

## 另有一个场景排序器可显示出节目中某一场景的全貌、场景各称及其连接的模板,也显示出每一个场景最后的状态及场景之间的声响等。

# 5.3.5 传统程序语言为基础的创作工具

## 精通编程的程序员对于多媒体编辑创作系统的限制及依赖工具箱产生对象的方式较不容易接受

## 因此,一方面保留传统语言的特性,另一方面改进其程序设计环境成为可视化的操作系统。

## 这样程序员既可以用传统的语言来编写程序又可方便地使用媒体开发工具箱，使这些工具箱内的编码可以直接被采用成为重用的编码 。

## Visual BASIC

## Visual C++

# Visual BASIC

## Microsoft Windows环境程序语言。

## VB提供各式的图形界面。

## VB是基于事件的语言,程序的行为附着于对象,等到对象被调用或被用户引发时才被执行。

## VB提供给鼠标与键盘双重的输入管道。同时也可摄取剪辑板,动态数据交换及对象连接与嵌入等设备,并通过MCI使音响、影片、动画等均可融入其中。

## 它还可将数据文件引进来使用.在完成一个多媒体产品后,可以将它制作成为一个可以直接执行的EXE文件而成为单独的一个应用程序。

# Visual C++

## Visual C++是Microsoft所推出多媒体程序设计软件, 它与Visual BASIC很相似 。

## Visual C++的工具包括有Visual Workbench、AppStudio、AppWizard、ClassWizard等模块。

## 设计VC程序的方法是先利用Visual Workbench及AppStudio来产生或编辑新的资源, 接着利用ClassWizard来产生类, 最后将这些资源在AppWizard中组织起来通过Build来完成构造一套新的应用节目或多媒体的节目。

# 5.4 Windows多媒体开发环境

# 5.4.1 媒体控制接口(MCI)

## MCI在控制音频、视频等设备方面,提供了与设备无关的API接口。

## 用户应用程序可使用MCI控制标准多媒体设备

## 不同设备其驱动控制方式不同:

## 一些MCI设备驱动程序(影碟机)直接控制目标设备;

## 一些MCI设备驱动程序(MIDI函数)可使用MMSYSTEM函数间接控制目标设备;

## 还有一些MCI设备驱动程序(影片演播器)则提供了与其它Windows DLL的高层接口。

## 应用程序通过设备的类型来区分设备 。

## 如果要通过MCI去控制设备,必须将相应的MCI驱动程序和设备的驱动程序,DLL(如果需要)装入。

## MCI驱动程序的安装可通过Windows中的控制面板来完成。在Windows中SYSTEM.INI文件中的〔mci〕部分包括了一个已安装了的设备类型表 。

# MCI接口分类

## Windows采用两种MCI接口:

## 一是使用命令消息接口函数,直接控制MCI设备;

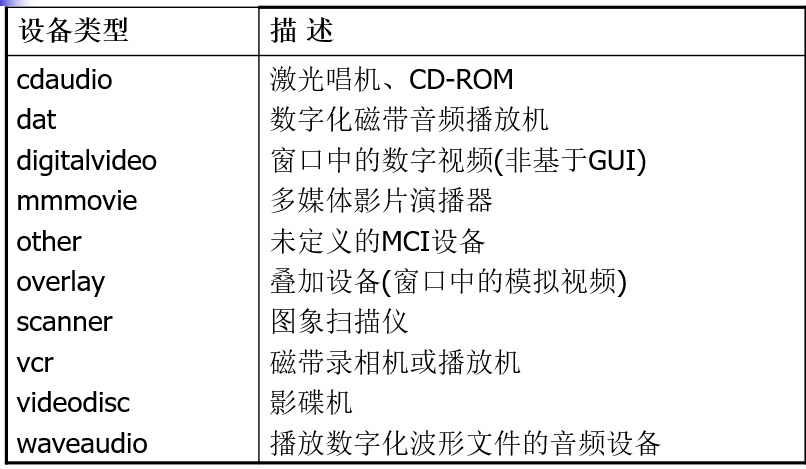
## 二是使用命令字符串接口函数,基于文本接口或命令脚本来控制MCI设备。

## 不同之处在于它们基本命令结构及其发送消息到设备的原理不同。

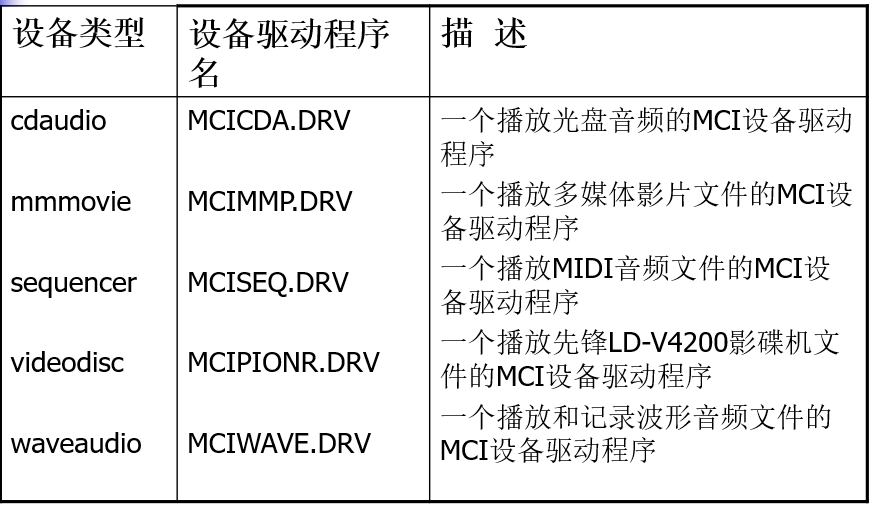
## 命令消息接口使用消息控制MCI设备；

## 命令字符串接口使用文本命令控制MCI设备。

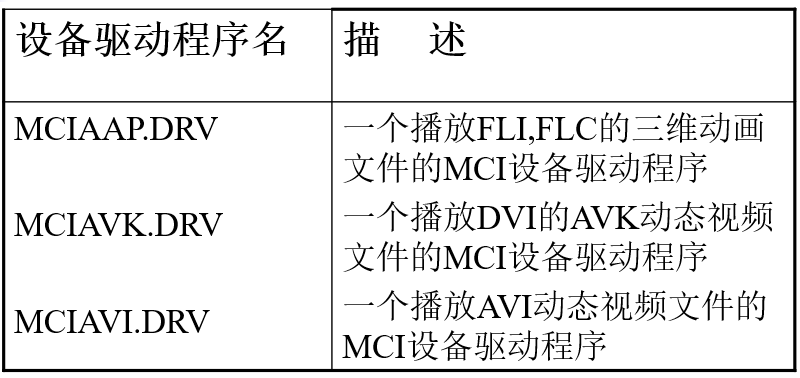
# MCI设备类型



# Windows提供的MCI设备驱动程序



# Microsoft其它MCI设备驱动程序



# 5.4.2 命令消息接口

## 使用命令消息接口发送MCI命令3个函数:

### MciSendCommand 发送一个命令消息到一个MCI设备;

### MciGetDeviceID 当打开一个设备时,返回这个设备的ID号;

### MciGetErrorString 返回对应于一个错误代码的字符串。

# 

## 发送命令消息 MciSendCommand函数定义:

## DWORD MciSendCommand(WORD DeviceID,WORD Message,DWORD Param1,DWORD Param2)

## 其中: DeviceID标识一个MCI设备;

## Message 标识要发出的消息,如MCI-OPEN等;

## Param1 为消息指定标志;

## Param2 为指定一个指向消息数据结构的指针。

## 该函数调用如果成功返回0,否则返回一个错误代码, MciGetErrorString 可获得对这个错误的文本描述。

## MCI命令消息分类:

## 直接由MCI解释的命令;

## 由所有的MCI设备所支持的命令;

## 基本命令;

## 扩展命令。

## 对于不同的设备类型, MCI使用一组不同的扩展命令控制此类设备特殊性能:

## 

## 第一组是MCI元素文件操作扩展命令组, 包括MCI\_COPY,MCI\_CUT,MCI\_DELETE, MCI\_PASTE,一般具有编辑MCI数据能力的设备支持;

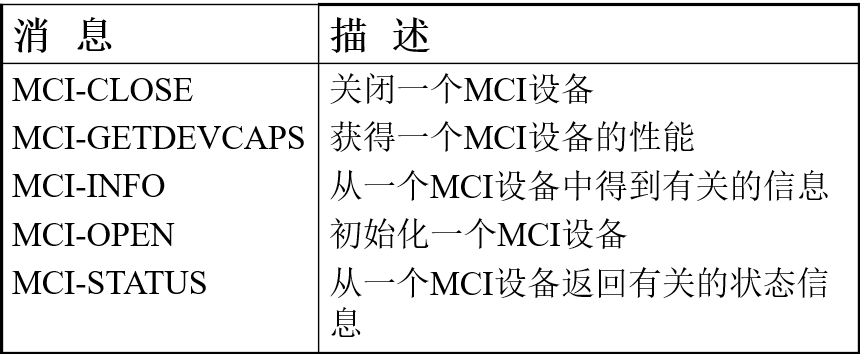
## 第二组是MCI设备操作及定位扩展命令组, 包括MCI\_CUE,MCI\_ESCAPE,MCI\_SEEK,MCI\_STEP;

## 第三组是窗口或视频设备的扩展命令组,包括MCI\_FREEZE, MCI\_PUT,MCI\_REALIZE, MCI\_UNFREEZE, MCI\_UPDATE, MCI\_WHERE, MCI\_WINDOW。

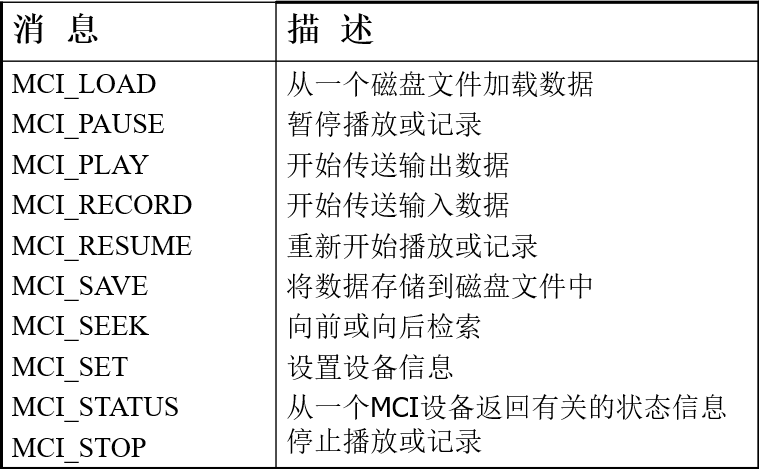
# 直接由MCI解释的命令



# 所有的MCI设备支持的命令消息



# 基本命令消息



# 打开一个设备

## 使用设备之前,必须使用MCI\_OPEN命令消息来初始化该设备

## 打开MCI设备的方法有以下几种:

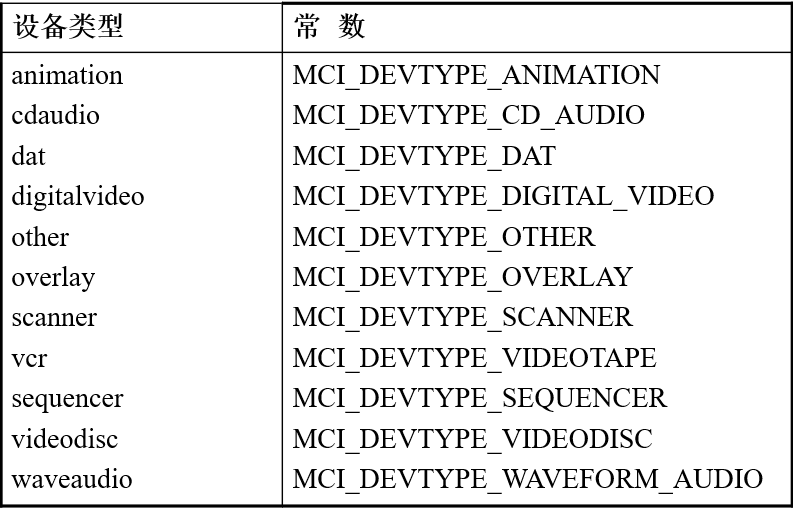
### (1)用设备类型字符串指定待打开的MCI设备

### (2)用MCI设备驱动程序名指定待打开的MCI设备

### (3)用设备类型常数(见下表)指定待打开的MCI设备

### (4)仅用设备元素指定打开的复合设备

# 设备类型及常数



# 关闭一个设备

## MCI\_CLOSE命令消息取消对一个设备或者设备元素的访问,它类似于一个文件的关闭操作。

## 为了有助MCI管理设备,应用程序在它使用完这个设备之后,应该明确地关闭它所使用过的每一个设备或者设备元素。

# 5.4.3 命令字符串接口

## 使用命令字符串接口3个函数:

### MciSendString向一个MCI设备驱动程序发送一个命令字符串。这个函数同时也具有对于回调函数和返回字符串的参数。

### MciGetErrorString返回一个同错误代码相对应的错误字符串。

### MciExecute向一个MCI设备驱动程序发送一个命令字符串。

# 

## 发送命令字符串 MciSendString函数的语法定义如下:

## WORD FAR PASCAL MciSendString(LpstrCommand, LpstrRtnstring, WORD Rtnlength, hcallBack)

## 指针LpstrCommand 指向一个以NULL结尾的MCI控制命令的字符串。这个字符串的形式如:

## *Command device\_name arguments*

## *如：* open 难忘的一天.mp3 alias mysong

## 指针LpstrRtnstring指向一个由应用程序提供的返回字符串缓冲区。 Rtnlength是缓冲区大小.

## 句柄hcallBack用来指定接收并处理MCI向应用程序发出的MM\_MCINOTIFY消息的窗口句柄。

# 

## 使用MciExecute发送命令字符串

## MciExcute函数是MciSendString的简化形式。其语法定义如下:

## BOOL MciExecute(Lpstr Command)

## LpstrCommand是一个指向以NULL结束的控制命令的字符串,字符格式同MciSendString。若函数调用成功返回TRUE,否则返回FALSE。

# 

## MciGetErrorString函数返回一个MCI错误代码的文本描述字符串,其语法如下:

## WORD MciGetErrorString(DWORD Error, LpstrBuffer, WORD Length)

## Error是错误代码,是上一次MciSendCommand或MciSendString函数调用的返回值;

## LpstrBuffer指向一个缓冲区指针,用来接收系统返回的文本描述;

## Length指定LpstrBuffer的长度。函数调用成功返回TRUE,否则表示查询的错误代码未知。

# Windows DirectX9.0家族成员

## DirectX Graphics：集成DirectDraw（视频输入输出）和Direct3D（3D图形API）

## DirectInput（有关鼠标、键盘、游戏杆和其他游戏控制设备，以及力回馈设备的一组API ）：支持输入输出

## DirectPlay（为开发者提供了开发诸如多人游戏或聊天程序的工具，完成了与用户连接相关的所有复杂工作）：网络游戏的通信组织

## DirectSetup：自动安装DirectX组件

# Windows DirectX9.0家族成员

## DirectMusic：MIDI音频播放

## DirectSound：实现.wav格式的波形声音数据的播放控制

## DirectShow：为在Windows平台上处理各种格式的媒体文件的回放、音视频采集等高性能要求的多媒体应用提供了完整的解决方案

## DirectX Media Objects：DirectShow Filter简化模型

# **5.4.5 DirectShow技术**

## 保证大量多媒体数据处理的高效性

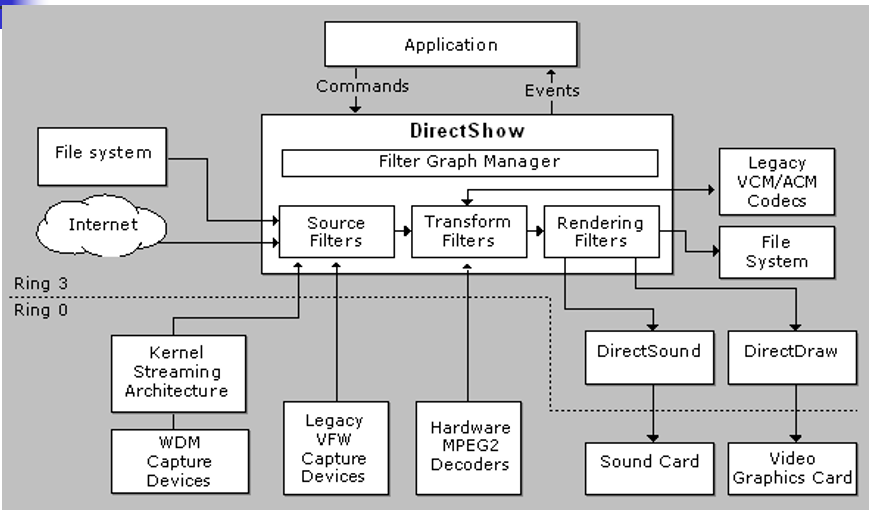
## 音视频同步

## 简单方法处理复杂媒体源问题：文件、网络、广播电视等

## 处理多种媒体格式的问题

## 支持不可预知的硬件

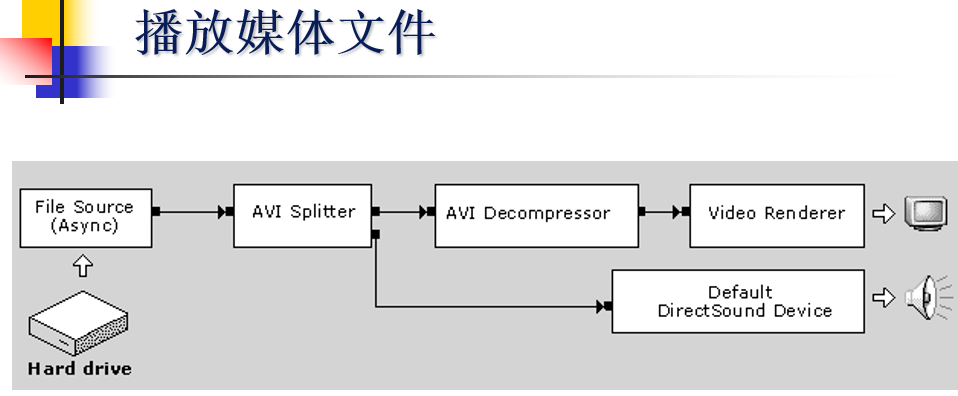
# DirectShow系统



# Filter Graph

## DirectShow的基本模块是Filter软组件,Filter通常实现一个对多媒体流的功能操作,比如读文件、从视频捕捉设备上获取视频、解码特殊的流格式、将数据传给图形卡或声卡等。

## 一系列连接的Filters被称为Filter Graph,通过Filters Graph Manager控制Graph中的数据流,它可以传递事件通知给应用程序, 以便程序能对事件作出反应。



# Filter概述

## Filter是DirectShow中最基本的概念

## Filter Graph是Filter的“容器”，Filter是Filter Graph中最小的功能模块

## Filter由一个或多个Pin组成，Filter之间通过Pin连接，构成一条顺序链路

### 仅含输出Pin，没有输入Pin为Source Filter

### 既有输入Pin，又有输出Pin为Transform Filter

### 仅有输入Pin，没有输出Pin为Rendering Filter

# Pin

## Filters的连接点被称为pin，每一个pin是独特的COM对象，提供IPin接口

## 每一个pin都有一个input或output的方向，必须与相反方向的pin进行连接

## 连接过程的步骤大致如下：

### Filter Graph Manager在输出Pin上调用IPin::Connect（带输入Pin的指针作为参数）

### 如果输出Pin接受连接，则调用输入Pin上的IPin::ReceiveConnection

### 如果输入Pin也接收这次连接，则双方连接成功

# 数据传输

* IPin接口主要是用于Pin连接, 用于数据传送一般是输入Pin上实现的IMemInputPin接口(方法IMemInputPin::Receive)
* 连接着双方Pin拥有同一个Allocator
* 数据传送时, 上一级Filter从输出Pin的Allocator中(调用IMemAllocator::GetBuff)得到一个空闲Sample的数据内存地址, 将数据放入其中
* 将这个Sample传送给下一级Filter的输入Pin

# 数据传送模式

## 数据传送主要有两种模式：

## 推模式（Push Model）

## 拉模式（Pull Model）

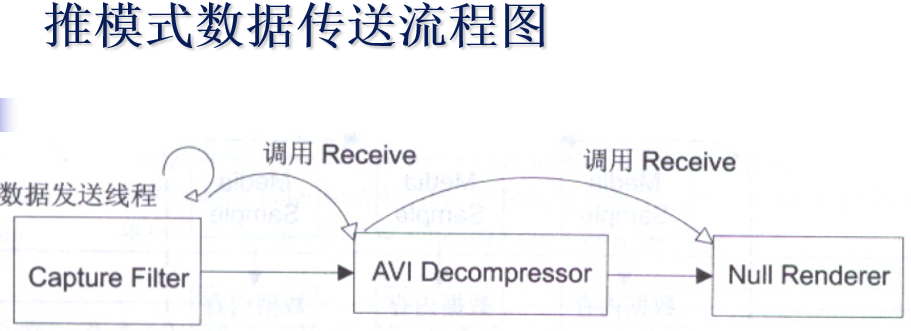
# 推模式

## 推模式最典型情况在Live Source(实时源)中。这种源能自己产生数据，并使用专门的线程将这些数据“推”下去

## 数据从Capture Pin出来，调用Transform Filter的输入Pin上的IMemInputPin::Receive函数，实现数据从Capture Filter到Transform Filter的传送

## 在Transform Filter内部，Filter将这块数据进行转换处理后将数据放到输出Pin的Sample中，调用Rendering Filter输入Pin上的IMemInputPin::Receive函数，而实现数据从Transform Filter到Rendering Filter的传送

## Rendering Filter接收到数据，进行必要的处理后就返回，数据传送的一个轮回完成



# 拉模式

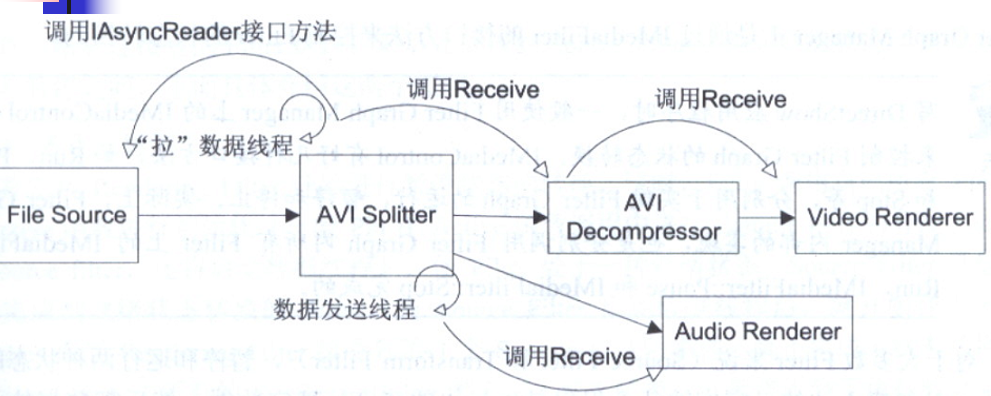
## 拉模式最典型发生在File Source(文件源)中。这种源管理数据, 但它没有把数据“推”下去的能力，而要靠后面的Filter来“拉”

## Source Filter输出Pin上实现IAsyncReader接口

## Source次级Filter的输入Pin上有一个“拉”数据的线程, 不断调用Source Filter输出Pin上的IAsyncReader接口方法来取得数据

## 在Transform Filter内部,将从Source Filter中取得的数据进行处理，然后通过各个输出Pin发送出去，往下的数据传送方式，与推模式相同

# 拉模式数据传送流程图



## DirectShow应用程序至少包含两个线程

## 应用程序主线程：**状态的改变**

## 数据传送子线程：**样本的传递**

# Filter项目的功能分析

## 功能单一化

### 需求过多，适度分解

### 需要在输入Pin上进行缓存的Filter,不需将缓存功能独立出去

## 选择一种Filter模型

### 大多数采用推模式，上级为被动源则用拉模式

### 区分Source Filter、Transform Filter或Rendering Filter

## 定义输入输出

### 输入、输出Pin的数量

### Pin上支持的媒体类型

## 接口定义

### Filter中哪些属性需要被外部知道和定制

## 其它一些特殊需求

### Pin上的数据缓存

### 给输出Sample打时间戳

### Filter内部需要其它线程及其同步问题

# Filter的设计

## 选择一个合适的父类

### 推模式Source Filter用CSource

### Filter接收输入，处理后输出用CTransformFilter（就地处理用CTransInPlaceFilter）

### Pin需要拉数据用CBaseFilter,并且Pin使用工具类CPullPin

### Filter不再需要输出用CBaseFilter（播放有标准Filter）

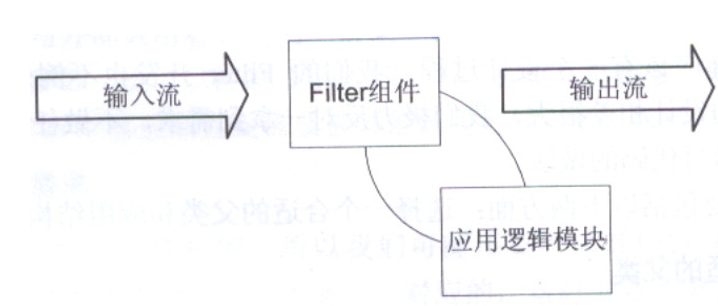
## 应用结构设计

### Filter是应用框架，应和应用逻辑分离

### Filter框架负责输入输出

### 应用逻辑负责数据处理控制策略，包括算法实现、效率优化等

# Filter框架与应用逻辑分离结构



# Source Filter Example：VOD Source Filter

## 流媒体接收Filter是一个典型的Source Filter

# VOD Source Filter功能分析

## 进行数据缓存,为下级Filter提供数据源

## 采用拉模式：MPEG-1 Stream Splitter只工作在拉模式下

## 一个输出Pin,支持媒体类型为MEDIASUBTYPE\_MPEG1VideoCD / MEDIASUBTYPE\_MPEG1System / MEDIASUBTYPE\_Avi

## 接口定义

### **设置流数据读取位置**

### **读取流中指定数量的数据**

### **获得数据总长度和当前可读取长度**

### **数据对齐方式**

### **数据读取同步对象的锁定/解锁**

## 特殊需求

### **网络传输、数据缓存、多线程问题**

# VOD Source Filter的设计

## Asynbase ：静态库项目，实现异步读写基类

### CAsyncStream：纯虚类，标识数据流源

### CAsyncRequest：表示输入输出请求

### CAsyncIo：实现了数据输入输出控制,包括同步/异步模式

### CAsyncOutputPin：输出Pin，从CBasePin派生

### IAsyncReader：拉模式的基本接口

## Async：使用asynbase库,实现拉模式Source Filter

### CMemStream：实现CAsyncStream，指定数据源内存区

### CMemReader：继承IAsyncReader，添加注册功能

# 媒体定位的实现

## 通过Filter Graph Manager上IMediaSeeking接口实现随机定位，其真正的实现在Filter上。

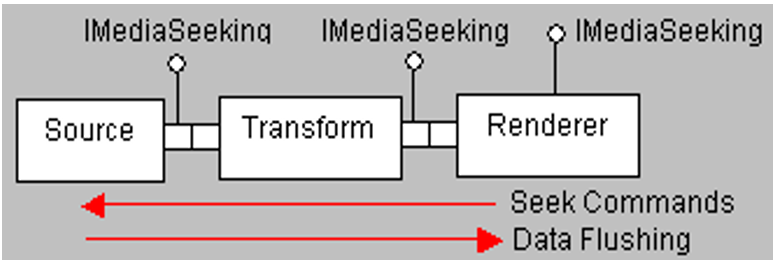
## Filter Graph Manager从Rendering Filter提出请求。

## 一般Rendering Filter不能真正执行定位操作而向上级Transform Filter输出Pin请求。

## 如果该级不能实现则向前继续请求,否则执行并返回

## 不成功的请求一直上溯到Source Filter

# IMediaSeeking接口的获取过程



# Flushing

## 丢弃采样的处理过程称为Flushing

## 当事件改变正常的数据流时,它使得Graph作出相应的反应

## 有BeginFlush和EndFlush两个方法

### BeginFlush：开始丢弃数据

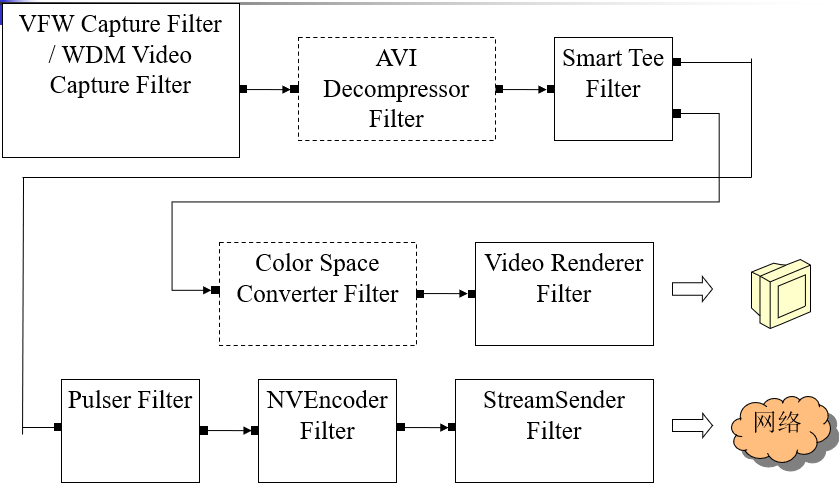
### EndFlush：结束丢弃数据

# Transform Filter Example：Pulser

## 视频媒体采样从用于捕捉的Filter产生后就被送到Rendering Filter进行回放然后再编码发送，可见视频回放的速率与捕捉的速率是一样的，如果没有速率转换Filter(Pulser)的话，编码发送的速率也将等同于捕捉的速率。

## 但是视频流发送速率一般为预定的值(使用前设定),在视频会议系统的客户端中默认为10帧每秒(出于带宽考虑)。但视频捕捉设备的捕捉速率一般为30帧每秒,而且大多数捕捉设备不能对速率进行调节, 因此必须有一种机制能够将媒体采样产生的速率降低，速率转换Filter正是起到这个作用

# 视频会议系统发送端Filter Graph



# Pulser Filter的设计

## 速率转换Filter是一种Transform Filter

## 用CPulserFilter类实现，从CTransInPlaceFilter基类派生而来

## Pulser Filter实现IPulser接口,用于调节速率

## Pulser Filter的输入Pin用CPulserInputPin类来实现，派生自CTransInPlaceInputPin基类

# 本章小结

## 首先介绍了媒体数据的获取方法，图形和动画的制作

## 对常用的多媒体创作工具进行了分类, 并以典型的软件为例介绍了它们各自的功能特点

## 最后介绍了Windows多媒体开发环境,着重介绍MCI的概念和使用方法, 以及DirectShow技术。

## 希望通过本章的介绍对读者开展多媒体应用有所帮助。