

基于 DirectShow 流媒体采集和播放系统设计和实现^{*}

李智芳, 黄 超

(中山大学电子与通信工程系, 广东 广州 510275)

摘 要: 阐述了 DirectShow 的体系结构。研究了开发基于 DirectShow 应用系统的一般方法, 着重于流媒体采集和播放的关键技术: DirectShow 过滤器和过滤器图表的构建及管理。提出了一种基于 DirectShow 流媒体采集和播放系统结构, 并通过对系统的需求分析, 得到系统的功能。实现了基于 DirectShow 流媒体采集和播放系统。实践证明, DirectShow 简化了流媒体应用系统的开发, 缩短了开发周期。

关键词: 流媒体; DirectShow; 采集; 播放; 视频; 音频

中图分类号: TP32; TN911.73 **文献标识码:** A **文章编号:** 0529-6579 (2005) S2-0166-04

流媒体 (streaming media) 的出现使人们避免了收听、收看网上节目时漫长的下载等待过程, 有了流媒体网络用户就可以边看边下载。流媒体已经广泛地应用于在线电视、视频点播、远程教育、在线影院等。目前支持流媒体播放和采集的最为流行产品有 Microsoft Media、Real Media 和 QuickTime。特别是 Microsoft 公司不仅为用户提供免费的流媒体播放器下载, 而且为用户提供二次开发流媒体系统提供免费的 SDK (Software Development Kit) 即 Directx SDK。用户可以利用 Directx SDK 定制适合自己需要的流媒体系统。

Directx 已经发展到 Directx V9.0。Directx SDK V9.0 中包含了支持流媒体开发的工具包 DirectShow。DirectShow 是 Microsoft 为 Microsoft Windows 平台流媒体提供结构体系。它提供了高质量的多媒体流的录制和播放, 支持各种媒体格式, 包括 ASF, MPEG, AVI, MP3, WAV 等几乎所有当前流行的媒体格式, 支持使用 WDM (Windows Driver Model) 设备录制^[1-4]。

DirectShow 简化了基于 Windows 平台的数字媒体的应用开发。将应用与复杂的数据传输、硬件差异和同步隔离开来。同时, 它为需要定制应用解决方案提供底层流控制结构。用户也可以创建自己的 DirectShow 组件以支持新的媒体格式。

1 DirectShow 体系结构

DirectShow 主要解决的问题是: 快速处理媒体流所包含的大量数据; 音频和视频同步; 处理来自各种源数据如本地文件、计算机网络、广播电视、

摄像机等; 支持各种格式如 AVI、ASF、MPEG、DV; 能够兼容未知的硬件设备。DirectShow 体系结构如图 1, 图中给出了应用程序, DirectShow 组件和 DirectShow 所支持的一些硬件和软件关系。

为了满足流媒体所需要的数据流量, Directshow 尽可能地使用 Directdraw 和 DirectSound。Directshow 通过以时间戳 (Time Stamped) 样本的形式封装媒体数据来使视音频同步。为了处理各种格式媒体和各种硬件, Directshow 模块结构。在模块结构中应用混合和匹配不同软件组件称为过滤器 (filter)。如图所示 DirectShow 过滤器与各种不同的设备通信并控制他们。包括声卡, 电视卡, VFW 多媒体数字编码器, 及视频显示及声卡, 因此 Directshow 把应用与这些设备内部细节隔离开来。DirectShow 也为某些文件格式提供原有的压缩和解压缩。

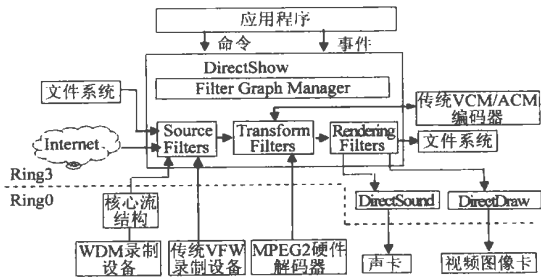


图 1 DirectShow 系统结构

Fig. 1 Architecture of DirectShow

^{*} 收稿日期: 2005-09-01

作者简介: 李智芳 (1966 年生) 男, 硕士, 讲师; E-mail: isslzf@zsu.edu.cn

2 DirectShow 应用开发方法

DirectShow 单元是称为过滤器 (Filter) 的软件组件。过滤器是可以在媒体流上执行操作的软件组件。如 DirectShow 过滤器能够读文件；从视频捕获设备获得视频；对各种不同的格式进行解码，如 MPEG-1 视频；将数据转送到图像卡或声卡。过滤器接收输入产生输出。例如，如果 Filter 对 MPEG-1 视频解码，输入是 MPEG 编码流，输出是一系列没有压缩的视频帧。在 DirectShow 中，应用所执行的任何任务都是通过将系列过滤器连接在一起。因此一个过滤器的输出是另一个过滤器的输入。一组相互连接的过滤器称为过滤器图表 (Filter Graph)。如图 2 展示了 Filter Graph 播放一个 AVI 文件。

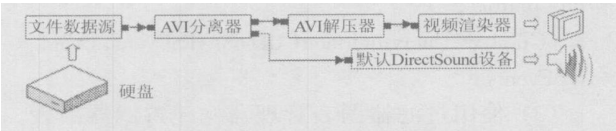


图 2 播放 AVI 文件的过滤器图表
Fig 2 Filters Graph of playing AVI file

文件数据源过滤器从硬盘读取 AVI 文件。AVI 分离器过滤器将文件解析成两个流，压缩的视频流和音频流。AVI 解压器过滤器对视频帧解码。视频渲染器 (Video Renderer) 过滤器使用 DirectDraw 或 GDI 将视频帧在显示器上画出来。默认 DirectSound 设备过滤器使用 DirectSound 播放音频流。应用程序不必管理所有的数据流。称为过滤器管理器 (Filter Graph Manager) 的高层组件控制着过滤器。应用程序进行高层的 API 调用，如“Run”或“Stop”。如果需要对流进行更多的控制，则可直接通过 COM 接口访问过滤器。过滤器管理器也将事件消息转送给应用程序。过滤器管理器也有另一个目的是通过将过滤器连接在一起，为应用程序建立过滤器提供方法。

DirectShow 应用程序执行三个任务：应用程序创建过滤器管理器实例；应用程序使用过滤器管理器构建过滤器图表；应用程使用过滤器管理器控制过滤器图表和流数据通过过滤器。当处理完成，应用程序释放过滤器管理器和所有的过滤器。DirectShow 是基于 COM 的；过滤器图表管理器和过滤器都是 COM 对象。

3 使用 DirectShow 设计流媒体采集与播放系统

3.1 系统主要功能概述

系统具有如下功能：①流媒体采集功能：系统

能够从多种信号源俘获视频和音频信号，如电视卡、摄像头、磁带录放机、VCD (DVD) 播放机等视音频设备。并能够按照事先创建的录制表自动定时录制。②素材信息管理功能：存储待播放的素材信息，素材信息包括素材名称，编号，长度，所在位置等，采用 Adb 数据库技术，使用 Access 数据结构。实现素材的管理，如素材的增减，素材信息的修改，检索，统计等。用户选择所需播放的素材建立播放表。③播放表创建功能：通过创建播放表允许用户根据需要按照顺序播放，也可以随机播放，或随意点击选择播放某条节目；也可以定时播放。④流媒体播放功能：通过底层 DirectShow 根据播放表提供素材信息，取得流媒体文件，并判断媒体格式利用相应的解码方法对流媒体进行解码，将媒体数据流，在已知绑定的窗口显示出来。实现解码回放有关的功能。如播放，停止，暂停等基本功能。同时还具有其它的附加功能如取得当前播放位置、上一条节目、下一条节目、循环播放、重复播放、设置播放速度、音量调整、时间码的显示、显示当前播放信息、文本的显示、快速定位、允许随时打开单个或一个目录的流媒体文件进行解码播放、播放网络流媒体等等。

3.2 系统结构

根据解码回放系统的主要功能我们得到系统的结构如图 3 所示。主要包含素材管理子系统、流媒体播放子系统、流媒体采集子系统。各子系统的主要功能如前所述。

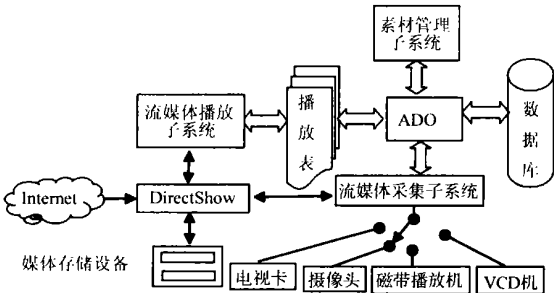


图 3 流媒体采集播放系统结构

Fig 3 Architecture of Captuning and Playing Streaming Media System

3.3 流媒体采集实现

如图 3 所示，视频信号源可以来自各种视音频设备，因此流媒体由各种采集设备来进行。在具体实现上，Directshow 通过构建不同采集设备的过滤器图表称为捕获图表 (capture graph)。为方便创建捕获，Directshow 提供了称为捕获图表构造器 (Capture Graph Builder) 来进行，其中包含建立和控制捕获图表的方法。过滤器图表构造器的使用比较

简单, 只要创建这个组件并获得 ICaptureGraphBuilder2 接口, 然后用接口方法 SetFiltergraph 将过滤器图表管理器上的 ICaptureGraphBuilder 接口设置给它, 就可以调用 ICaptureGraphBuilder2 的其他接口继续过滤器图表的构建了。那些复杂的前端链路都可以通过 ICaptureGraphBuilder2 的 RenderStream 函数轻松做到。

采集设备有一个预览 (Preview) 输出 Pin 和一个捕获 (Capture) 输出 Pin, 前者主要用于视频图像的预览, 而后者主要用于视频数据的采集。预览和采集这两种应用在采集设备前端链路上是相同的, 仅在后端有些差异: 预览时只接视频渲染器; 而在采集时可能还要接视频编码过滤器, 最终使用过滤器写入器 (Filter Writer) 将数据写到文件中。所以为采集设备构建预览用的过滤器时, 也采用捕获图表构建器组件来简化它的过滤器图表创建过程, 但为了不影响采集的应用, 系统将过滤器链路抽象成两部分: 采集设备过滤器以前的 (包括采集设备过滤器本身) 定义为输入部分, 而采集设备过滤器以后的定义为输出部分。两部分的过滤器分开创建, 然后再连接。这样可以在程序设计时, 对于预览和采集两种应用, 输入部分的构建是相同的, 只要定制不同的输出就可以了。

系统中设计了三个控制类: CGraphController、CPreviewController 和 CCaptureController。CGraphController 是 CPreviewController 和 CCaptureController 的父类, 主要定义公共接口, 以及实现一些预览和采集都要用到的功能函数; CPreviewController 主要负责预览用的过滤器图表的构建; CCaptureController 主要负责采集用的过滤器图表的构建。CGraphController 类的 Activate 函数体现了程序创建过滤器图表通用过程, 其中 CreateInputFilters 是公共的, 而 CreateOutputFilters 和 ConnectFilters 都可以在子类 CPreviewController 或 CCaptureController 中定制。

3.4 流媒体播放实现

(1) 初始化 COM 库。通过调用 CoInitialize 初始化 COM 库。

```
HRESULT hr=CoInitialize (NULL);
```

(2) 创建过滤器图表管理器实例。调用 CoCreateInstance 创建 Filter Graph Manager 实例。IGraphBuilder *pGraph; HRESULT hr=CoCreateInstance (CLSID_FilterGraph, NULL,

```
CLSCTX_INPROC_SERVER, IID IGraphBuilder, (void **)&pGraph);
```

类标识符 (CLSID) 是 CLSID_FilterGraph。过滤

器图表管理器由运行中的 DLL 提供, 执行环境是 CLSCTX_INPROC_SERVER。DirectShow 支持多线程模型 (free threading model), 所以也可以调用带 COINIT_MULTITHREADED 标记的 CoInitializeEx。调用 CoCreateInstance 返回 IGraphBuilder 接口, 该接口包含构建过滤器所需的大多数方法。还需要以下两个接口。

IMediaControl 控制流。它包含启动和停止图表的方法。IMediaEvent 具有从过滤器图表获得事件的方法。该接口用于等待播放结束。这两个接口可以从过滤器图表管理器得到。使用返回的 IGraphBuilder 指针来查询它们。

```
IMediaControl * pControl; IMediaEvent * pEvent; hr=pGraph->QueryInterface( IID IMediaControl, (void **)&pControl);
```

```
hr=pGraph->QueryInterface( IID IMediaEvent, (void **)&pEvent);
```

(3) 使用过滤器图表管理器构建过滤器图表。构建过滤器图表播放文件通过调用单个方法调用。

```
hr=pGraph->RenderFile(L"C:\\myavi.avi",NULL); IGraphBuilder::RenderFile 方法构建一个过滤器图表以播放某个文件。第一个参数是文件名, 表示一个宽字符 (2 字节) 串, 第二参数是保留参数必须为 NULL。如果文件不存在或不能识别文件的格式则该方法失败。然而, 假如方法成功, 则过滤器图表准备播放文件。
```

(4) 运行图表, 使数据通过过滤器。为了运行 Graph, 调用 IMediaControl::Run 方法。

```
hr=pControl->Run ();
```

过滤器图表运行时, 数据流过过滤器并以视频和音频形式呈现出来。播放在一个独立的线程上进行。调用 IMediaEvent::WaitForCompletion 方法等待播放结束。

```
long evCode= 0;
```

```
pEvent->WaitForCompletion (INFINITE, &evCode); 该方法直到文件播放结束或直到某一特定的时间到来。值 INFINITE 意味着一直播放到文件结束。应用程序结束时释放接口指针并关闭 COM 库。
```

4 结 语

流媒体的应用方兴未艾, 各种基于流媒体的应用研究正成为热点, 取得的成果是有目共睹。

定制开发满足特定要求的流媒体系统也成为必然。本系统使用 DirectX SDK 9.0 在 windows Server 2003 采用 Visual C++ 6.0 开发设计。系统采用开放式模块结构, 使其容易扩充功能, 今后考

虑为系统追加非线性编辑子系统，使其功能更完善。

参考文献：

[1] 陆其明. DirectShow 实务精选[M] . 北京: 科学出版社, 2004.
[2] 郑晶晶, 高胜法. 一种基于 DirectShow 的 MPEG2 流式网

络播放器[J] . 计算机应用, 2003, 10: 138— 142.
[3] 方波, 曾致远. 基于 DirectShow 的流式立体视频播放器的设计与实现[J] . 宽带网络与传输, 2004, 10: 39— 42.
[4] 朱英南, 陈晓, 曾学文. 一种嵌入式流媒体播放器 VCR 控制的设计与实现[J] . 中国有线电视, 2003(19/20) : 8— 11.

Developing System Of Capturing And Playing Streaming Media Based On DirectShow

Li Zhi fang, HUANG Chao

(Department of Electronics & Communication Engineering, Sun Yat sen University, Guangzhou 510275, China)

Abstract: Architecture of DirectShow and general method of developing application based DirectShow are described. Several key technology, such as creating and managing filter and filter graph, of Capturing and playing Streaming Media are emphasis. System architecture and requirement analysis and functions are given. A kind of streaming media application system based on DirectShow is implemented. DirectShow simplifies media playback, format conversion, and capture tasks in practice and the developing cycle is cut short.

Key words: streaming media; DriectShow; capture; playback; video; audio