

ASF 流媒体切割技术的研究与实现

赵志敏, 吴中福, 李 华, 朱郑州, 陈晓铭

(重庆大学 计算机学院, 重庆 400044)

摘 要: ASF 文件作为当前主要的流媒体格式之一, 给人们的生活带来极大的方便。但同时受到广告泛滥、内容模糊, 不能分离精彩片段的严重制约, 必须开发一种对其进行编辑的软件, 既节省时间, 又可以反复观看经典内容。并且通过使用 WMF SDK 提供的生成器、读取器、编辑器可以忽略 ASF 文件底层的实现细节, 提高开发效率。针对 ASF 文件切割中存在的问题, 对切割精度、绕开生成器的压缩编码过程、属性编辑等方面进行了研究, 并实现了对 ASF 文件的切割。

关键词: 流媒体; 头对象; 读取器; 生成器; 数字版权管理

中图分类号: TP311.52

文献标识码: A

文章编号: 1673-629X(2007)07-0175-03

Study and Realization of ASF Dismemberment Technology

ZHAO Zhi-min, WU Zhong-fu, LI Hua, ZHU Zheng-zhou, CHEN Xiao-ming

(College of Computer Science, Chongqing University, Chongqing 400044, China)

Abstract: The advanced systems format (ASF) is the file format used by Windows media. Audio and/or Video content compressed with a wide variety of codes can be stored in an ASF file and played back with the Windows media player. But nowadays, Some ASF Video files contain many advertisements, some frames are illegible and it is hard to separate interesting content from other frames. So it is necessary to develop software which can edit ASF file effectively, enable interesting content played repeatedly. It can facilitate the development of the software by using reader object, writer object, editor object provided by WMF SDK, in spite of the details of ASF files. Points out the problems in the process of ASF file dismemberment. At the same time, dismemberment precision, decode/encode technology avoiding the writer object, properties revision are also studied. Finally software is developed which realized the dismemberment of ASF files.

Key words: streaming media; header object; reader object; writer object; digital rights management

0 引言

流媒体是指在 Internet 网络中使用流式传输进行播放的技术。流式传输将整个 A/V(Audio/Video)数据经过特殊的压缩编码后, 放到专用的流媒体服务器上, 由视频服务器向用户计算机连续、实时传送数据, 用户不用将整个的音视频文件下载到本地磁盘, 便可以在线观看, 即节约空间, 又节省时间^[1,2]。

自从 1995 年 4 月 RealNetworks 公司推出了世界上第一个流产品 RealAudio 软件以来^[3], 流媒体信息已经爆炸性地增长。网上教学、网络电视、视频点播等技术已经非常普遍, 成为人们生活中重要的组成部分。然而网络上传输的流媒体信息大部分都没有进行编辑, 充斥着大量的广告信息; 不能增加制作者的版权信

息; 不能基于文件内容进行切割, 以方便观众迅速定位精彩片段。通过开发一种 ASF 文件切割软件可以节省观看时间, 快速定位精彩内容, 在当前形势下非常有意义。

1 ASF 文件格式

ASF(Advanced Systems Format)文件逻辑上包含三种顶级对象: 头对象(Header Object)、数据对象(Data Object)和索引对象(Index Object)^[4]。头对象和数据对象是每个 ASF 文件所必需的, 索引对象是可选的。头对象用于描述文件的全局信息, 包括文件大小、流的数量、编解码信息等; 数据对象以固定大小的数据包存储流数据; 索引对象主要保存时间索引和数据位置的映射信息, 便于应用程序的随机访问。

ASF 作为一种文件系统格式, 是一个具有包容性的定义。Windows Media 文件是 ASF 文件格式的一种具体应用, 只是采用的编解码方式不同, 仍然是 ASF 文件。具体区别如下: 只存在非压缩的或使用 Windows Media 视频编码器编码的视频流, 则保存成

收稿日期: 2006-09-27

基金项目: 国家发改委科学研究计划项目(CNGI-04-15-3A)

作者简介: 赵志敏(1980-), 男, 湖南祁东人, 硕士研究生, 主要研究方向为计算机网络与通信、现代远程教育; 吴中福, 教授, 博士生导师, 主要研究方向为计算机网络与通信、网络安全、电子商务。

wmv;没有视频流,且只有非压缩的或使用 Windows Media 音频编码器编码的音频流,则保存成 .wma;其他的情况则保存成 .asf。本软件采用 VC 6.0 + WMF SDK 开发,主要为了实现包括 .wmv, .wma 在内的所有 ASF 文件的切割。

2 数据流程

WMF SDK 是 Microsoft 公司提供的用于编辑 ASF 文件的开发包,包括一系列对象:如读取器、生成器、编辑器等等^[5]。正是这些 SDK 对象屏蔽了 ASF 文件切割的技术实现细节,使得在编写程序时不需要太多的考虑流媒体底层技术,程序实现的主要过程如下:

- (1)通过 WMCreateReader 函数创建读取器对象;
- (2)获取源文件中的 Profile 信息;
- (3)生成器对象根据 Profile 信息的描述,创建一个 ASF 文件头对象;
- (4)复制源文件中的属性信息、编码信息、脚本信息、标志信息,并实现对属性信息中的标题、版权等进行修改;
- (5)按照呈现时间顺序,把读取器输入的 Sample 数据传送给生成器。生成器根据 Profile 的描述交替打包,并写入 ASF 文件的数据对象;
- (6)自动添加索引,以提供 ASF 文件的随机定位能力(图 1 为程序流程图)。

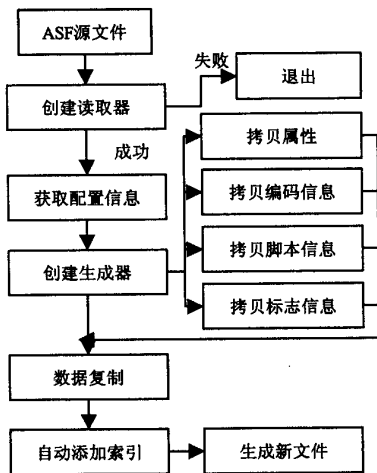


图 1 程序流程图

3 系统实现中的问题与解决方法

为了实现 ASF 文件的切割,数据信息的复制是必不可少的。从程序实现的角度考虑,可以采用两种方法:一是使用 API 函数 CopyFile(LPCTSTR SourceFileName, LPCTSTR NewFileName, BOOL FlagIfFileExists)。经过 CopyFile 函数复制之后,源文件与目标文

件的内容完全一样,不能对元数据属性进行修改、开始与结束时间的设置,达不到对文件切割的目的。另一种方法就是使用 WMF SDK 来生成新的 ASF 文件。利用源文件的 Profile 信息创建一个新的 ASF 文件头对象,并且复制源文件中所有的属性信息、编码信息、脚本信息和标志信息等,以保证不丢失数据,最终生成一个新的合法的 ASF 文件。

3.1 开始时间和结束时间的修改

为了实现对 ASF 文件开始时间的修改,可以采用 WMF SDK 提供的 Start 函数。Start 函数的原型如下:

```

HRESULT Start(
    QWORD cnsStart, //开始时间
    QWORD cnsDuration, //播放时长
    float fRate, //播放速率
    void * pvContext //回调状态通知接口);
  
```

Start 函数支持给读取器对象设置开始时间和结束时间,并通过 OnStreamSample 函数的回调方式读取数据,本程序就是通过这种方式实现 ASF 文件的切割。在创建读取器、获取 Profile 信息、创建生成器等一些必要属性之后,通过 Start(mStartTime, 0, 1.0, 0) 函数将开始时间输入,但是这时候并不能够修改播放持续时间,所以 cnsDuration 的值为零。对结束时间的修改是通过 OnStreamSample 函数来完成的,当采用时间小于开始时间不复制数据,而只有当采用时间大于开始时间才向生成器中写入数据,直到当前时间大于或等于结束时间。程序关键代码如下:

```

BOOL CASFCut::Start(CString& inSourceFile, CString&
inTargetFile, long inStartTime, long inMaxDuration)
  
```

```

{
    .....
    hr = mIReader->Start(mStartTime, 0, 1.0, 0); //开始
    拷贝数据,mStartTime 是开始时间,结束时间通过 OnStream-
    Sample 函数确定
    .....
  }
  
```

```

HRESULT CASFCut::OnStreamSample(
    WORD wStreamNum,
    QWORD cnsSampleTime,
    QWORD cnsSampleDuration,
    DWORD dwFlags)
  
```

```

{
    .....
    mCurrentTime = cnsSampleTime;
    if(cnsSampleTime < mStartTime)
        return hr; //小于开始时间不拷贝
    .....
    if (mTargetDuration > 0 && cnsSampleTime /10000 > (
  
```

```
mTargetDuration + mStartTime / 10000))
```

```
{
    mIsEOF = TRUE;
    return hr;
}
.....
```

3.2 切割精度的设定

本地文件切割精度可以采用关键帧,也可通过时间来确定。目前,本程序只实现了精确到 ms 的定位,所以切割后的文件与源文件存在一定的差别,生成后的文件会比设定的时长多几十 ms,但由于人的视觉暂停时间公认为 80~100ms,所以不会产生太大的影响。程序关键代码如下:

```
BOOL CASFCut::Start(CString& inSourceFile, CString&
inTargetFile, long inStartTime, long inMaxDuration)
```

```
{
    .....

    mStartTime = inStartTime * 10000; //SDK 中时间采用
    100ns 为单位,inStartTime 参数单位是 ms,所以再扩大 104
    mTargetDuration = inMaxDuration; //不乘倍数。因为
    mTargetDuration 虽然定义的是 QWORD,但在时长较大时还是
    溢出,所以不乘以倍数,在使用时统一到 ms
    .....
}
```

3.3 绕开生成器的压缩编码过程

读取器的工作流程是:首先将数据流输入到读取器中,然后进行流分离,再分别解压缩,输出数据。生成器的工作流程是:首先对输入的数据进行预处理,然后压缩编码,交错打包后发布到接收器。默认情况下,读取器会自动解压缩,而生成器会自动压缩编码,这样在生成新的 ASF 文件时就产生了先解码后编码的无用操作,所以需要绕开生成器的压缩编码过程^[6]。程序关键代码如下:

```
HRESULT CASFCut::CreateWriter(void)
```

```
{
    .....

    mIReaderAdvanced -> SetReceiveStreamSamples(mStream-
Numbers[i], TRUE); //绕开读取器的解压缩过程
    .....

    for(DWORD i = 0; i < inputCount; i++) //遍历生成器
    的每个输入
    {
        hr = mIWriter->SetInputProps(i, NULL); //设置 Set-
InputProps 的第二个参数为 NULL,以绕开生成器的压缩编码过
        程
    }
}
```

3.4 元数据属性的修改

对于 ASF 文件中的大部分信息,SDK 的编辑器并不允许修改,否则将影响文件的正常播放。SDK 能够修改的属性有 Title, Author, Copyright, Description 等。程序关键代码如下:

```
void CInfoDlg::OnModifyAttributes(CString &mSourceFile)
{
    *((DWORD*) pValue) = atol("cqu"); //设置标题为 cqu
    hr = pHeaderInfo3 -> ModifyAttribute(streamNum, 21,
WMT_TYPE_STRING, 0, pValue, MAX_PATH);

    *((DWORD*) pValue) = atol("cquzz"); //设置作者为
    cquzz
    hr = pHeaderInfo3 -> ModifyAttribute(streamNum, 22,
WMT_TYPE_STRING, 0, pValue, MAX_PATH);

    ..... //其他的两项类似
}
```

3.5 DRM 保护的判断

DRM(Digital Rights Management)就是数字版权管理,是 Microsoft 公司通过一定的技术手段,控制用户观看的权限,以保护媒体内容的版权、保障创作者的利益。对于受 DRM 保护的媒体文件,必须获取相应的许可证才可以观看,并且合法用户也不能够对受 DRM 保护的媒体文件进行复制。所以本软件先对媒体文件进行判断,如果是受 DRM 保护的文件,不能进行切割。程序关键代码如下:

```
HRESULT CASFCutDlg::IsDRMProtect(CString & inSource-
File)
```

```
{
    .....

    m_pHeaderInfo -> GetAttributeByName(&wStreamNum,
pwszName, &wmType, pData, &cbLen);
    .....

    fProp = *(int*) pData;
    if(FAILED(hr))
    {
        MessageBox("不能从读取器中获得 fProp 信息", "失败",
MB_OK);
    }
    else if(fProp)
    {
        MessageBox("受 DRM 保护,不能进行切割", "失败",
MB_OK);
    }
}
```

4 结 论

简要叙述了 ASF 文件格式,以及开发过程中遇到

(下转第 182 页)

```

returnvalue = True
ElseIf UserRole < > newPermissionUser. JIAOWUYUAN
Then
'学院级用户需判别是否本学院学生
StudentCollege = newStudentBaseInfo. GetStudentCollege
If UserCollege = StudentCollege Then
returnvalue = True
End If
End If
Catch ex As Exception
Throw ex
End Try
If returnvalue = True Then
Me. DetailView(xh) '有限权查看新生详细信息
Else
Throw New UnauthorizedAccessException
End If
End Function
.....
End Class

```

由上可见,凡涉及到新生详细信息查看这一事件,StudentDetailInfoProxy 代替真实的 StudentDetailInfo 与其他程序打交道,确定是否有限权查看新生详细信息。只有 StudentDetailInfoProxy 认为合适时,才会将客户端的请求传递给真实的主题对象 StudentDetailInfo。采用 Proxy 模式同样也可以保证系统的可扩展性,用户权限发生变化时,只需要修改 Proxy 函数,不必修改

业务逻辑。

4 结束语

Observer 设计模式的应用使得系统的可扩展性、灵活性获得了大大的提高,Proxy 模式的使用更保证了系统的安全性。该系统已在本校实际投入运行,极大地提高了各部门的工作效率。但系统仍可以作进一步的改进,如在专业分班方面可以考虑设计分班算法,实现一键分班,进一步减少用户的工作量;同时还可以设计目标的代理类实现模式代码,使得目标的业务逻辑与模式代码分离^[6],进一步提高系统的可扩展性和代码复用度。

参考文献:

- [1] 伽 玛. Design Patterns: Elements of Reusable Object - Oriented Software[M]. 北京:机械工业出版社,2002.
- [2] 库 柏. 设计模式:C# 语言版[M]. 北京:清华大学出版社,2004.
- [3] 张振兴,宋 雨,韩立森,等. Observer 模式分析及其在 .NET 中的改进[J]. 微机发展,2003,13(8):36-37.
- [4] 汤 姆. VB. NET 设计模式高级编程——构建强适应性的应用程序[M]. 北京:清华大学出版社,2003.
- [5] 彭展阳. JAVA 实用系统开发指南[M]. 北京:机械工业出版社,2004.
- [6] 付登科. AOP 改进观察者模式——实现关注点的分离[J]. 计算机应用,2005(25):410-412.

(上接第 177 页)

的关键技术,解决了视频切割中常见的黑屏问题,绕开了生成器的压缩编码过程,并保留了原文件中的所有属性,最终实现了对 ASF 文件的切割。本软件已经通过重庆软件中心的测试,证明了它的正确性、高效性和健壮性等。程序执行界面如图 2 所示。

参考文献:

- [1] 钟玉琢,向 哲. 流媒体和视频服务器[M]. 北京:清华大学出版社,2003.
- [2] 白 煜,辛向晖. 网络流媒体案例教程[M]. 北京:清华大学出版社,2004.
- [3] 邸春红,于淑玲,杜 勇. 流媒体技术概述[J]. 沈阳医学院学报,2004,6(1):50-52.
- [4] Fleischman E. Advanced Streams Format (ASF) Specification, Internet - Draft[EB/OL]. 2004. <http://www.microsoft.com/windows/>.
- [5] Windows Media Format SDK. [CP/DK]. 2003. <http://msdn.microsoft.com/library>.



图 2 程序执行界面

- [6] 陆其明. Windows Media 编程导向[M]. 北京:清华大学出版社,2005.