Microsoft Media Stream Technology and Online Media Teaching

Delin Tan, PhD
Southern University at New Orleans
dtan43@yahoo.com

Abstract

In this presentation, we explore how to use the Microsoft Media Streaming Technology to create software that can do class room teaching, recording and live online education broadcasting simultaneously. This software not only can play voice with the lecture notes like any commercial online presentation, but also can show the instructor's mouse writing and keyboard typing. Moreover, with the help of media server, the instructor can broadcast the online voiced lecture via Internet to students from any place if there is a fast network connection. We use this software design as an example to educate our computer science students to understand the basic coding architecture and operations of Microsoft Encoder technology and Media Server services.

§ 1. Weakness of Some Commercial Presentation Creators

There are many commercial online presentation creators in the market, such as from Adobe and Articulate software companies [Adobe][Articulate]. Those two commercial presentation creators can make good voiced online presentations. However, they have some weakness。 The biggest weakness is the audio replaying cannot be closely associated with the lecture notes. When the slide content is too bigger and the instructor's narration is too long, then the online students could lose their focus because they do not know where the instructor's talking is referred to. The best way to overcome this is to add mouse drawing action on the presentation. So far, we don't know if any commercial presentation creators could do so.

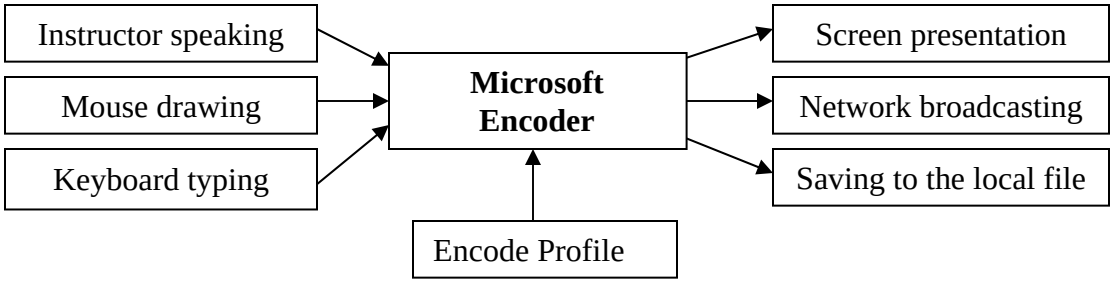
The second weakness of those two commercial presentation creators is that they have to pre-record the teaching voice chip for each slide before the converting. They cannot create online voiced presentation during the classroom teaching time.

The third weakness of the current commercial presentation creators is that they need some special server to deploy the replay, which not only costs school thousand dollars a year, but also make the deployment too complicated for the instructors.

§ 2. Microsoft Media Encoder and Program Diagram

With the Microsoft Media Encoder Technology, we can create better online voiced presentation without the above weaknesses. First, we need to install the Media Encoder 9 series to our window system, which can be freely downloaded from the Microsoft official

website. Our programming language is .NET 2003 C#. Therefore this software can run in the window XP system. The programming diagram is the following



The output of the program is an audio data with *.wma extension, which carries scripting code about information of mouse drawing, images and keyboard typing.

§ 3. Microsoft Encoder Profile

To encode the instructor's voice for Internet broadcasting or saving to a local file is quite simple. However, in order to encode mouse drawing and keyboard typing into the voice data, we need a special encoding profile. This profile actually is an XML format file, which is used to configure Encoder about how to add scripts code into the voice. We can use Microsoft Profile Editor to generate this file. In this profile if the scripting bit rate not bigger enough, then the fast mouse writing action would not be possibly recorded. The recommend scripting bit rate is 4800-9600 bits per second. Also, the audio stream bit rate should be bigger enough to carry scripting data. The recommend audio stream bit rate is 96k-144k bits per second. The beginning several lines of this profile look like that

```
<profile version="589824"
  storageformat="1"
  name="MyProfile"
  description="Script Commands">
      <streamconfig majortype="{73647561-0000-0010-8000-00AA00389B71}"
          streamnumber="1"
          streamname="Audio Stream"
          inputname="Audio409"
          bitrate="96000"
          bufferwindow="-1"
          reliabletransport="0"
          decodercomplexity=""
          rfc1766langid="en-us"
      >
```

§ 3. Initial Settings of Encode Object

First of all, we have to add Window Encoder reference to the window program. Because the Window Encoder is a COM component, we have to add

```
using WMEncoderLib;
using System.Runtime.InteropServices ;
```

at the program head position. Next, we create object

```
WMEncoderClass  MyEncoder = new WMEncoderClass();
```

which is the start point of our program. The following are the initial settings:

```
SrcGroupColl = MyEncoder.SourceGroupCollection;
SrcGroup = SrcGroupColl.Add("SG_1");    // pick up any name
SrcAudio = SrcGroup.AddSource(WMENC_SOURCE_TYPE.WMENC_AUDIO);
        // input source is audio
SrcAudio.SetInput("Default_Audio_Device", "Device", "");
        //  setup audio input is microphone
MyScriptSource = rcGroup.AddSource(WMENC_SOURCE_TYPE.WMENC_SCRIPT);
        // set up scripting configuration
MyScriptSource.SetInput("UserScript:// ", "", "");
        // set up scripting format
WMEncProfile2Class Profile = new WMEncProfile2Class();
        // create WMEncProfile class object
Profile.LoadFromFile(MyProfile.prx)
        // load profile from current folder
SrcGroup.set_Profile(Profile);
        // setup scripting profile
```

§ 4. Scripting Instructor's Actions into Voice

1) Mouse Writing

The window mouse writing actually is set of many small line segments. We only need to record the positions inside the voice. The following are the crucial code

```
protected override void OnMouseMove(MouseEventArgs e)
{
    int x = e.X ;
    int y = e.Y ;
    Graphics g = CreateGraphics();
    Pen pen = new Pen(color, 2);
    g.DrawLine(pen, lastX , lastY , x,y);
    lastX =x;
    lastY =y;
    MyEncoder.SendScript(0, "newX", "" + x);   // encode x position
    MyEncoder.SendScript(0, "newY", "" +y);   // encode y position
    MyEncoder.SendScript(0, "Draw", "1");     // draw line command
}
```

2) Forward or Backward Slide Notes

Make Bitmap objects for all slide images and store them in an array. When instructor forward or backward slide notes, we just update the current index of slide image and call On_Paint() method. The code is

```
protected override void OnPaint(PaintEventArgs e)
{
    Graphics g = e.Graphics;
    g.DrawImage(SlideImage[currentIndex], 0, 0, 691, 518);
    MyEncoder.SendScript(0, "TEXT", "Slide" + currentIndex);
}
```

3) Record Instructor's Typing

We allow instructor to change its typing. So the current slide image including mouse writing must be saved as a backup. We use the following event handling function of TextBox

```
private void txtInput_TextChanged(object sender, System.EventArgs e)
{
    graphics[current].DrawImage(BackupImage, 0,0,691,518);
    graphics[current].DrawString(text, font, new SolidBrush(color),
                            new Point(lastX, lastY-15));
```

```
    Graphics g = this.CreateGraphics();
    g.DrawImage(backs[current],0,0,691, 518);
    MyEncoder.SendScript(0, "AddText", text+"-" +size);
}
```

Here, BackupImage is the image of current slide, graphics[current] is an off screen Graphics object. We actually draw text on the off screen buffer first and copy to window later. That can easily handle instructors' undo action later.

§ 5. Save Output to Local File and Setup Internet Broadcasting

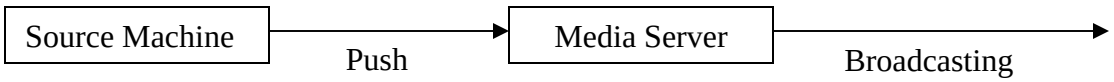
First we save the encoded voice file in a local folder in extension *.wma by the code

```
MyBroadcast = new Encoder();
MyOutFile = MyEncoder.File;
MyOutFile.LocalFileName = "lecture.wma";  // local file
```

At the same time we create an HTML file for replaying. Mainly this HTML file is used as input data for online reader. We can directly broadcast the encoded voice file from an HTTP port. To broadcast from port of number 8888, we need to setup code like:

```
MyBroadcast.set_PortNumber(WMENC_BROADCAST_PROTOCOL.
                                    WMENC_PROTOCOL_HTTP, 8888);
      // configure the internet broadcasting protocol
MyBroadcast.Start();
```

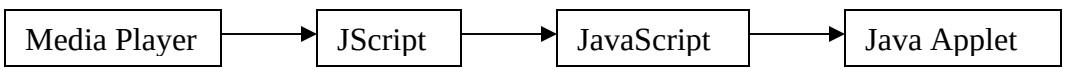
However, this broadcasting has limitation of 15 con-currencies. To increase the number of broadcasting con-currencies, we need to use Microsoft Media Server, which is a part of 2003 Server. There are two methods to do so: pull and push. The pull method means that the media server can pull the stream to server side and rebroadcast again. In this case, the media source machine and the Media server are working independently. However, the media server must be able to access the broadcasting port of the source machine. The better method is to use push as the following diagram shows:



In that case, the source machine can initially send stream to media server and activate the media server broadcasting point. Push method does not require the source machine to have a global IP address. Therefore the instructor can push the teaching stream to the school media server for Internet broadcasting. The students would not feel the time delay, as long as the online lecturing is proceeding continuously. The reader can read the detail explanations of encoding, pull and push operations in [Johnson] and [McEnvoy].

§ 6. Online Teaching Replay

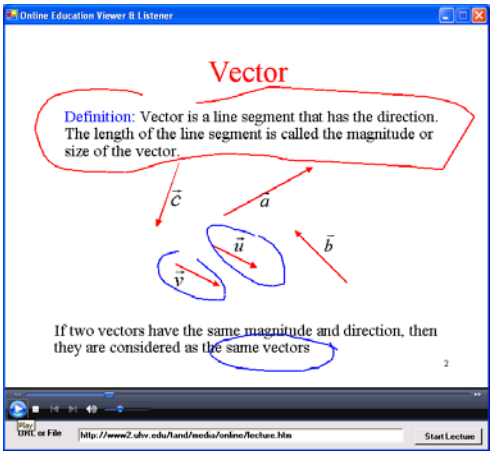
We actually replay the wma audio stream and use the scripting code that stored inside the audio file to draw notes images, mouse writings and keyboard typing. We only need to deploy wma file and html files in a regular web server (even it is a non Microsoft web server) and create a subfolder to store all slide notes images. We can add a Java applet to html to show mouse writing and keyboard typing. The work flow is:



However, the busy JavaScript code could cause overflow for Internet Explore browser in some PCs. So we design a window application to do the online teaching replay. The core object is Microsoft media player, which can fetch all information stored in the voice file. The crucial code is

```
private void Player_ScriptCommand(object obj,
                        AxWMPLib._WMPOCXEvents_ScriptCommandEvent e)
{
    string type = e.scType.ToLower() ;
    string param = e.param.ToLower();  ;
    if(type=="lastx") lastX = Int32.Parse(param);
    if(type=="lasty") lastY = Int32.Parse(param);
    if(type=="newx")  newX  = Int32.Parse(param);
    if(type=="newy")  newY  = Int32.Parse(param);
    if(type=="draw")
    {
        Graphics g = this.CreateGraphics();
        g.DrawLine(new Pen(color ,2), lastX, lastY, newX, newY);
        lastX = newX ;
        lastY = newY;
    }
}
```

The following is a screen shot of voiced online presentation replay window, which has slide image, instructor's voice, mouse writing and keyboard typing



Software can download from web site http://www2.uhv.edu/tand/OnlinePresentation.zip
After unzip, be sure to extract all DLLs in the same folder.

References

Adobe "Breeze Presenter", http://www.adobe.com/resources/breeze/presenter/

Articulate "Articulate Presenter", http://www.articulate.com/products/presenter.php

Johnson, Nels "Windows Media Series 9", CMP Books Publisher, 2003.

McEnvoy, Seth "Window Media Platform", Microsoft Press, 2003.