

# 基于 .NET 的培训课件设计与制作

## Design and Making Methods of Training Courseware Based on .NET

黄天文 (广东省肇庆市气象台 广东 肇庆 526040)

焦 飞 (肇庆市工贸学校 广东肇庆 526000)

**摘 要:** 介绍一种基于 .NET 的培训课件的设计思想与制作方法, 讨论并分析了软件实现中有关图形、音频、视频、多线程安全和 API 函数调用等关键性技术。利用现代信息技术改革传统教学模式和教学方法, 结合可视化编程工具, 制作的辅助培训教学软件的功能可以非常强大。

**关键词:** C#.NET 多媒体 培训 课件

### 1 引言

以传统授课方式为主的培训教学, 方式单一, 内容多, 时间短和资料匮乏等, 往往造成学员学习积极性下降。如何在这些不利因素的情况下, 使学员掌握知识和技能, 一直是培训工作所面临的实际问题, 而使用多媒体辅助教学不失为一种好的手段。

计算机多媒体技术和网络技术飞速发展, 随着硬件成本的下降, 计算机辅助教学(CAI)课件在教育上得到广泛应用, 对教学质量的提高起到了很大作用。虽然目前有很多教学软件, 但多数具备了通用性而失去了专用性。如何利用现代信息技术改革传统教学模式和教学方法, 结合可视化编程工具, 制作辅助培训教学软件, 是应该努力思考的问题。

利用可视化编程工具开发课件, 可以显示文字内容, 辅以声音、图形、图像、动画, 增加学员的视觉和听觉的刺激, 同时还可以让学员与计算机进行双向交流, 激发了学习的主动性, 极大缩短了教学时间, 提高培训效果。事实上, 可视化编程一直是培训教学课件制作的主要途径<sup>[1]</sup>。

该课件的开发平台是微软的 C#.NET。基于 .NET 的开发首选 C#语言, 它在多媒体应用程序开发方面具有强大的功能, 大大降低了设计难度, 而且培训课件的功能可以做得非常强大。

### 2 课件制作的分析与设计

#### 2.1 需求分析

培训课件是将文字内容、图形或图表、解说词、背景音乐、教学视频动画等按一定规律连接起来形成的一种计算机软件。利用 VC#.NET2005 开发课件应满足以下基本需求:

(1) 实现教学的互动性, 使其能够进行网上的课程学习交流; 通过声音、图像、动画等多媒体技术做到图文并茂, 增强教学的生动性<sup>[2]</sup>。

(2) 用户可以通过上网进行学习资料的浏览和下载。

(3) 学员在客户端可以通过网络上传作业, 教员可以及时反馈意见给学员, 并可以发送素材等给他们。

(4) 交互性是学员与教学课件之间的双向交流, 是决定一个课件水平高低的关键部分。课件应该让学员可以自由选择学习的章节、内容和采取何种方式。根据实际情况设计若干菜单或工具条, 在每一个菜单或工具栏上加以提示, 让学员很方便地选择其中的某一部分来学习。

#### 2.2 设计思想

课件采用二层体系结构可以设计简单些, 该软件的表现层与服务层都在客户端实现<sup>[3]</sup>, 数据层用来存放教学内容、教学所需要的素材和学员的个人资料(座号、机号、

作业情况、作业存放路径等)。基本结构如图 1 所示:

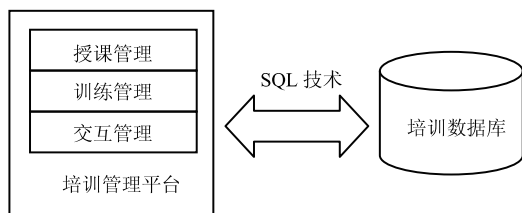


图 1 系统结构图

课件的一个重要特点就是交互性,具体表现在两个技术上的问题:操作的随意性和重复性。要解决这两个交互性问题,一般的课件制作软件就勉为其难了。毕竟,它们在对课件的控制方面还是比较弱的,问题的解决还得依赖于程序设计语言<sup>[4]</sup>。系统的设计流程图如图 2 所示:

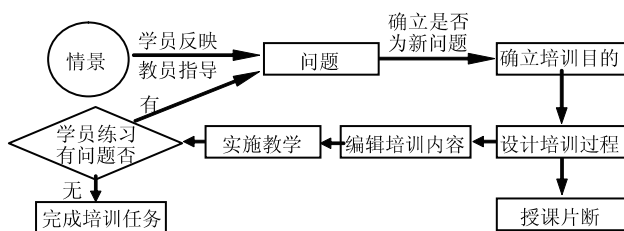


图 2 系统设计流程图

### 3 课件制作的关键技术

#### 3.1 文字内容

文字内容是用来表达课程的定理、定义、概念、公式等内容的文本,是培训课件中不可缺少的部分。为了使文字更美观,课件很多地方利用 DrawString 函数显示。C#提供了一整套相当丰富的类库、方法以及事件以供开发者使用。C#还引入了 GDI+,它是由 GDI 演变而来,具有比 GDI 更强大的功能,而且简化了程序员的编程工作。开发者运用这些,可以很方便地开发出具有强大图形图像功能的应用程序。下面的例子可帮助读者简单了解 C#图形编程的基本知识。

```
String drawString = " Sample Text" ;
Font drawFont = new Font(" Arial", 16) ;
SolidBrush drawBrush = new SolidBrush ( Color.
Black) ;
float x = 150.0F;
```

```
float y = 150.0F;
```

```
e. Graphics. DrawString ( drawString, drawFont,
drawBrush, x, y) ;
```

代码中 DrawString() 函数可以完美无缺地再现文字内容。若要用白色的背景则可由 gph. Clear( Color. White) 实现。这样就可以将教员写在黑板上的内容搬到计算机屏幕上。如果内容过多,一屏显示不下,还可以用滚动条翻页。

#### 3.2 图表

图表可以很形象地表现出事物之间的对比关系,是培训课件中一个重要的组成部分,起图文并茂的作用。例如学员提问最多的问题的统计图表可以直观地告诉教员今后教学中的重点。制作时可以先统计好数据并存入数据库,然后通过图表控件显示。ZedGraph 是 C#编写的 .NET 类库,提供了用户控件和 web 控件。它可以创建 2D 的线性图、条形图和饼图。它功能完整且有详细的功能自定义,不过使用默认的选项就足够好用了。该控件的使用可参考 [http://zedgraph.org/wiki/index.php?title=Main\\_Page](http://zedgraph.org/wiki/index.php?title=Main_Page)。

#### 3.3 声音

声音分成解说词和背景音乐两部分,利用声音可加强学员对需要重点掌握部分的印象,它也是培训课件中一个重要的组成部分。固定不变的解说词可以用录音机程序录好,转换成 WAV 文件或 MP3 格式的文件存储在磁盘上,然后可以根据实际情况,选择 C#的多媒体组件进行播放。

如果要经常改变解说词,那么最好把它以文本保存,便于修改。发音问题则转化为如何实现读取文件的中文发音或中文语音识别。首先需要安装微软的 Speech Application SDK ( SASDK ), 我们使用的版本是 SAPI 5.1,它能够识别中、日、英三种语言,可以在这里下载: <http://www.microsoft.com/speech/download/sdk51/>。需要安装两个文件: Speech SDK 5.1 和 5.1 Language Pack, 其中 5.1 Language Pack 可以选择安装支持的语言。

安装好以后,就可以开始进行语音程序的开发了。先设计一个能够朗读中英文混合语言的类,类的代码这里不详述,只是把重点部分给予解释:

```
private SpeechLib. SpVoiceClass voice = null; 这
里,定义一个用来发音的类,并且在第一次调用该类
```

时,对它用 BuildSpeech 方法进行初始化。再定义两个属性 Volume 和 Rate,能够设置音量和语速。

SpVoiceClass 有一个 Speak 方法,发音主要就是给该方法传递一个字符串,让它负责读出该字符串,如下所示:

```
private void Speak( string strSpeak ) {
    try {
        //SpeechVoiceSpeakFlags. SVSFlagsAsync 表示异步发音
        voice. Speak ( strSpeak, SpeechVoiceSpeakFlags. SVSFlagsAsync ) ; }
    catch( Exception err ) {
        throw ( new Exception ( " 错误:" + err. Message ) ) ; }
}
```

但是,这个方法本身并不知道你给的字符串是什么语言,需要我们告诉它这个字符串用什么语言读出。SpVoiceClass 类的 Voice 属性就是用来设置语种的,我们可以通过 SpVoiceClass 的 GetVoices 方法得到所有的语种列表,然后在根据参数选择相应的语种,比如设置语种为汉语如下所示:

```
private void SetChinaVoice( ) {
    voice. Voice =
    voice. GetVoices( string. Empty, string. Empty ). Item
    ( 0 ) ; }
```

0 表示汉语,1234 都表示英语,就是口音不同。这样,就设置了语种,如果结合发音方法,就可以设计出一个只发汉语语音的方法:

```
private void SpeakChina( string strSpeak ) {
    SetChinaVoice( ) ;
    Speak( strSpeak ) ; }
```

对于一段中英文混合的语言,让程序读出混合语音的方法是:编程把这段语言的中英文分开,对于中文调用 SpeakChina 方法,英文调用 SpeakEnglish 方法;至于怎样判断一个字符是英文还是中文,采用的是判断 ASC 码的方法。

这样,对于一段中英文混合文字,我们只需把它作为参数传递给 AnalyseSpeak 就可以了,它能够完成中英文的混合发音。当然,对于发音的暂停、继续、停止等操作,也很容易实现。

### 3.4 背景音乐

背景音乐在教学课件中起烘托气氛,造就身临其境的意境的作用。我们可以根据培训课件的内容选择相应的背景音乐。开发时要注意的是让解说词和背景音乐同时播放,并且都放在后台进行,丝毫不影响教员、学员对课件的控制、操作和信息交互。

```
public class APIClass {
    \[DllImport( " kernel32. dll " , CharSet = CharSet.
    Auto ) \]
    public static extern int GetShortPathName (
        string lpzLongPath ,
        string shortFile ,
        int cchBuffer ) ;
    \[DllImport( " winmm. dll " , EntryPoint = "
    mciSendString" , CharSet = CharSet. Auto ) \]
    public static extern int mciSendString (
        string lpstrCommand ,
        string lpstrReturnString ,
        int uReturnLength ,
        int hwndCallback ) ;
    }
    public void play() //播放{
        TemStr = " " ;
        TemStr = TemStr. PadLeft ( 127 , Convert. ToChar
        ( " " ) ) ;
        APIClass. mciSendString ( " play media " , Tem-
        Str , TemStr. Length , 0 ) ;
        mc. state = State. mPlaying ; }
```

### 3.5 动画

动画是培训课件中吸引学员的注意力,表现事物发展的渐进过程的部分。动画分两类:

一类是界面复杂,但播放时间有限。这样的动画可以用 3DMAX 等软件制作,再用 C#进行调用:选择菜单中的“工具”/“自定义工具箱”,打开“自定义工具箱”窗口,在“COM 组件”中选择“Window Media Player”这个选项,点击“确定”,于是在“工具箱”中便出现了“MediaPlayer”这一项,然后再将其拖至 Form 上,调整大小。使用 this. axWindowsMediaPlayer1. openPlayer( aviFileName) 打开视频文件。

但有时是用 C#播放一小段 AVI 文件,并且背景一

定要透明,而不是黑色或其它颜色。使用 MediaPlayer 播放的文件为透明的状态难以解决。在 VB 中有一个控件 Animation 可播放 AVI 文件,且它有一个属性选项是使背景透明。它是个 OCX 控件,如果装了 VS. NET 2005 则不必用 regsvr32 注册该控件了。把该控件拖入到窗体中,布好局,将其属性 AutoPlay 设置为 true,然后加入以下代码即可:

```
private void Form1_Load(object sender,
    System.EventArgs e) {
    this.axAnimation1.Open ( Application.
    StartupPath +
    @" \FILECOPY. AVI" );}
```

这样当窗体加载的时候就会自动播放此 AVI 文件了。

还有一类是简单的动画。动画本质上是由若干静态画面快速交替显示而成的,因为人的眼睛会产生视角暂留,对上一个画面的感知还未消失,下一张画面又出现,就会有动的感觉<sup>[5]</sup>。这种动画可以直接用 C# 的定时器 Timer 来制作。

在 C# 里定时器类有 3 个。System. Windows. Forms. Timer 应用于 WinForm 中,它通过 Windows 消息机制实现,类似于 VB 或 Delphi 中的 Timer 控件,内部使用 API SetTimer 实现。它的主要缺点是计时不精确,而且必须有消息循环,Console Application 无法使用。System. Timers. Timer 和 System. Threading. Timer 非常类似,它们都是通过 .NET Thread Pool 实现的,轻量,计时精确,对应用程序、消息没有特别的要求。System. Timers. Timer 还可以应用于 WinForm 中,完全取代上面的 Timer 控件。它们的缺点是不支持直接的拖放,需要手工编码。

### 3.6 多线程控件调用安全性问题

访问 Windows 窗体控件本质上线程是不安全的。如有两个或多个线程操作某一控件的状态,则可能会迫使该控件进入一种不一致的状态,还可能出现其他与线程相关的 bug,包括争用情况和死锁。确保以线程安全方式访问控件非常重要。NET Framework 有助于在以非线程安全方式访问控件时检测到这一问题。在调试器中运行应用程序时,如果创建某控件的线程之外的其他线程试图调用该控件,则调试器会引发一个 InvalidOperationException,并提示消息:“从不是创

建控件 control name 的线程访问它。”

如果代码是在 VS2003 中的话应该不会抛出这个异常,只有在使用 VS2005 的时候,如果你从非创建这个控件的线程中访问这个控件或者操作这个控件的话就会抛出这个异常。这是微软为了保证线程安全以及提高代码的效率所做的改进。

以前在 VS2003 中,通常都会忽略这种问题的存在,无限制的使用线程来完成一些工作,但是当系统慢慢变大的时候,这种滥用的线程不但没有提高效率,反而使找错误的效率降低,并且系统的复杂度大大的增加了。

微软的这一改动可在使用线程的时候,充分考虑一下线程的安全以及线程的使用规范。有一种解决的方法是通过将属性的 CheckForIllegalCrossThreadCalls 值设为 false 来禁用此异常。这会使控件以与在 Visual Studio 2003 下相同的方式运行,但这样做会使程序出现意外。如果程序的各线程之间没有互相争抢控件资源,则可以考虑用这个办法。例如:

```
public Form1()
{
    ... { InitializeComponent();
        CheckForIllegalCrossThreadCalls = false; }
    最好的方法就是利用 delegate 接口以及 Invoke 方法,它也是微软建议采用的跨线程调用的一种通用方法,就是使用代理来实现,就是将所要操作的代码放到一个代理中,然后将这个代理交给创建这个控件的线程来执行代码。例如:
    Thread t1, t2; // 说明为窗体类成员
    t1 = new Thread( new ThreadStart( BackgroundProcess ) );
    t1.Start(); // 启动线程 t1
    private delegate void dd(); // 定义一个代理
    private void BackgroundProcess() {
    dd = delegate() // 将代理实例化为一个匿名代理 {
    int i = 1;
    while (true) {
    listBox1.Items.Add( " Iterations: " + i. ToString
    ( ) );
    i ++;
    Thread.Sleep( 2000 ); // 指定线程休眠的时间
```

```
};  
listBox1.Invoke( dd ); }
```

上面这个代码只是在代码中声明了一个代理,并且用 VS2005 中新加的语法(匿名代理,又名匿名方法)来实例化这个代理,将在线程中要操作的代码都放到这个匿名方法中去,然后通过使用控件的 Invoke 方法(也可以使用控件的 BeginInvoke 方法,Invoke 方法是同步的,BeginInvoke 方法是异步的)来调用这个代理。当然在 Invoke 方法中你也可以输入任何形式的代理。通过这种方式调用的代码就不会抛出那个异常了。

还有一种方法是使用 BackgroundWorker 方法,具体可参考网上的最新资料,例如:

<http://blog.csdn.net/zcy1212/archive/2007/06/20/1659413.aspx> 的介绍就比较详细,适合软件开发初学者。

### 3.7 在课件中访问 Internet

随着 Internet 的发展和远程教学的需要,在多媒体 CAI 课件中实现对 Internet 的访问变得越来越重要。市面上许多书籍介绍的对 Internet 的访问方法都比较复杂,如在程序中调用 API 函数,使用第三方控件等。其实在实践中发现 VC# 已自带了 webbrowser 控件,完全可以通过它建立具有 Web 访问功能的超链接。主要代码是 WebBrowser1.Navigate() [6]。

### 3.8 在课件开发中 API 函数的应用

C# 的类库和控件只提供了 Windows 的基本功能,对于那些复杂和特殊的功能必须使用 API 函数实现。

在 .Net Framework SDK 文档中,关于调用 Windows API 的指示比较零散,并且其中稍全面一点的是针对 Visual Basic .net 讲述的。本文将 C# 中调用 API 的要点汇集如下,希望给未在 C# 中使用过 API 的朋友一点帮助。

调用格式如下:

```
using System.Runtime.InteropServices; // 引用此名称空间,简化后面的代码
```

// 使用 DllImportAttribute 特性来引入 API 函数,注意声明的是空方法,即方法体为空。

```
[DllImport("user32.dll")]
```

```
public static extern Return Type FunctionName( type arg1, type arg2, ... );
```

// 调用时与调用其他方法并无区别,以使用字段进一步说明特性,用逗号隔开,如: `[DllImport("kernel32", EntryPoint="GetVersionEx")]`

DllImportAttribute 特性的公共字段有: CallingConvention 指示向非托管实现传递方法参数时所用的 CallingConvention 值。其中 CallingConvention.Cdecl 为调用方清理堆栈,它使调用者能够调用具有 varargs 的函数。CallingConvention.StdCall 为被调用方清理堆栈,它是从托管代码调用非托管函数的默认约定; CharSet 控制调用函数的名称版本及指示如何向方法封送 String 参数; EntryPoint 指示要调用的 DLL 入口点的名称或序号; ExactSpelling 指示是否应修改非托管 DLL 中的入口点的名称,以与 CharSet 字段中指定的 CharSet 值相对应; PreserveSig 指示托管方法签名不应转换成返回 HRESULT、并且可能有一个对应于返回值的附加 `[out, retval]` 参数的非托管签名; SetLastError 指示被调用方在从属性化方法返回之前将调用 Win32 API SetLastError。

## 4 结束语

以上是在多媒体 CAI 课件开发过程中的一些心得和做法,是对现代化教育教学的初步探索和尝试。在此基础上,可以利用 .NET 强大的多媒体与网络功能,结合数据库服务器构建基于 B/S 模式的网上教学系统,提供 E-mail、BBS 和网上考试等功能。

## 参考文献

- 1 王咸伟. 多媒体课件开发中的关键技术. 中国电化教育, 1998, 7: 46 - 47.
- 2 焦飞, 黄天文, 滕少华. 小图报处理与作图系统的设计与实现. 计算机工程与设计, 2007, 28(9): 2169 - 2176.
- 3 王海东, 赵珺. 基于 .NET 技术的 Web 多媒体课件系统研究. 计算机工程与设计, 2004, 25(4): 550 - 553.
- 4 蒋恩松, 肖辉军. 基于可视化编程的课件制作. 吉林化工学院学报, 2001, 18(1): 49 - 52.
- 5 方亮, 郑耿, 刘国平. 图形系统的设计与实现. 计算机应用研究, 2005, 22(9): 173 - 175.
- 6 黄天文, 彭端. 用 Visual C# 制作浏览器界面. 广东气象, 2005, 4: 41 - 42.