基于 Authorware 的听力自测题的设计与实现

Design and Implementation of Listening Test Based on Authorware

孙志杰　（山东警察学院　山东济南 250101）

摘　要： 功能强大的 Authorware 为多媒体课件的制作提供了无限可能，本文针对传统听力自测题课件制作的不足之处提出了针对性的改进方法，实现了声音与试题的同步，为无纸化考试提供了技术保障。

关键词： Authorware　ODBC　听力自测题　外部函数

Authorware 是一套基于流程图的可视化多媒体制作工具，由于其人机互动，图、文、声、像等多媒体手段俱全的特点，被广泛地应用于各种互动式多媒体教学软件的开发。同时，Authorware 还提供了对数据库的弱支持，开发人员可以利用 ODBC 访问、查询、更新数据库里的数据，保证了课件的数据动态更新，尤其是对自测题的制作提供了极大的便利。但是对于英语考试中常见的听力测试，尚没有完美的解决方案。传统的实现方法的不足之处表现在(1)使用 Authorware 提供的声音图标，无法动态生成试题；(2)使用 JumpOutReturn 系统函数，无法使声音与试题同步运行。本文提出一种改进的基于 Authorware 的听力自测题的实现方法，其优势在于(1)充分利用数据库的强大优势，实现自动随机出题；(2)实现了声音与试题的同步，使用户可以边听边做题，更加接近于实际考试场景。

1　数据库设计

考虑到普及性和易用性，本文的数据库采用微软 Office 2003 办公套件中的 Access 数据库。传统的英语听力考试题型主要有两种，一是对话理解，其题型特点是一个对话后跟一个问题；二是短文理解，其题型特点是一篇短文后跟 5 个问题。为此，可分别设计数据表。新建一 Access 数据库，数据库名为 test_listening，使用设计视图设计两个数据表，CONVERSATION 和 PASSAGE，其数据结构分别如图 1(1)、(2)所示。

新建一个名为 sound 的文件夹，存储听力录音文件。其命名规则为(1)对话：conversation + 数字；(2)短文：passage + 数字。

2　Authorware 设计与实现

按照英语听力自测题的常规题型，应当是 10 个简单对话题之后紧接着 2 篇短文理解。为此“听力测试.a7p”文件的整体结构的设计(一级及二级流程图)如图 2 所示。iniPubVariables 计算图标用来初始化全局变量；regDataSource 计算图标用来自动注册数据源，打开数据库；listeningTest 导航图标为出题模块，其实现流程如图 2 右图所示；end 图标为退出提示模块。

	Field Name	Data Type	
🔑	SN	AutoNumber	自动编号
	CHOICE_A	Text	选择项A，文本类型字长为1
	CHOICE_B	Text	选择项B，文本类型字长为1
	CHOICE_C	Text	选择项C，文本类型字长为1
	CHOICE_D	Text	选择项D，文本类型字长为1
	RIGHT_ANS	Text	正确答案，文本类型字长为1
	FLAG	Number	0或1，整型，选中标志

图 1　(1)

	Field Name	Data Type	
🔑	SN	AutoNumber	自动编号
	CHOICE_A	Text	选择项A，文本类型字长为1
	CHOICE_B	Text	选择项B，文本类型字长为1
	CHOICE_C	Text	选择项C，文本类型字长为1
	CHOICE_D	Text	选择项D，文本类型字长为1
	RIGHT_ANS	Text	正确答案，文本类型字长为1
	FLAG	Number	0或1，整型，选中标志
	PASSAGE_NO	Number	长整型，短文编号

图 1　(2)

将创建的数据库文件 test_listening.mdb 和 sound 文件夹复制到与“听力测试.a7p”同一目录下。在计算图标“iniPubVariables”中输入以下内容初始化全局变量：

score：=0 －－初始化得分

getnum：=1 －－初始化题目数量

pre：=Array(0,30) －－初始化数组用以存储用户答题结果，正确，相应题号赋值为1，否则为0

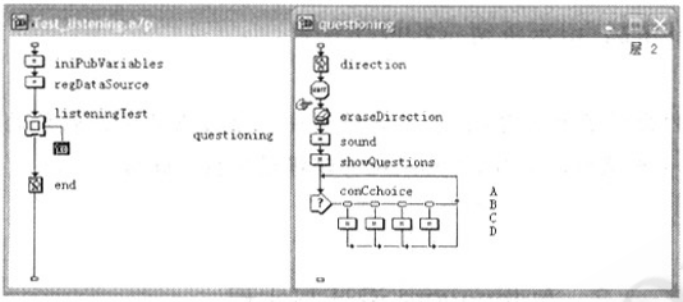

图 2

2.1 自动配置 ODBC 数据源

Authorware 中首先应该创建 ODBC 数据源，然后才能实现对数据库的访问[4,5]。可以通过引入外部函数" tMsSDSN. U32 实现 ODBC 数据源的自动配置，其优势在于通过 Authorware 编制的程序，打包成可执行文件时，可以在其他机子上直接运行，而不需要手动去配置数据源。

引入外部函数 tMsDSN. u32（可在 Authorware 的安装目录中找到）中的 tMsDBRegister（）函数，该函数含有三个参数：dbreqtype 用来指定对数据源的操作方式，1 表示增加 ODBC 数据源，4 表示增加 ODBC 系统数据源；dbtype 指定 ODBC 数据源需要使用的驱动程序；dblist 指定 ODBC 数据源的名称、对数据源的描述以及与之相关联的数据库文件（注意它们之间必须用分号隔开）。打开计算图标" regDataSource"，输入以下内容完成 ODBC 数据源的自动配置[1,2,3]：

dbreqtype：=4 －－数据源类型

dbtype：=" Microsoft Access Driver（*. mdb）" －－数据库文件驱动类型

dblist：=" DSN = test_listening；" －－数据源名称

dblist：= dblist^" Description = Listening Test；" －－数据源描述

dblist：= dblist^" FIL = MS Access；" －－数据库类型

dblist：= dblist^" DBQ =" ^FileLocation^" test_listening. mdb；" －－数据库文件位置

tMsDBRegister（dbreqtype，dbtype，dblist） －－自动注册 ODBC 数据源

2.2 听力测试实现

双击框架图标" listeningTest"，删除里面的导航面板和导航链接交互图标里的所有分支，另外添加三个按钮方式的计算图标，分别命名为" Previous"、" Next" 和" Exit"，调整好按钮位置。计算图标" Previous" 用来返回上一题以作修改。为此，首先要考虑是否回到了题库顶部，是则予以提示，显示第一题；否则还要考虑返回前用户所作的选择是否正确，是则扣掉所得分数，否则得分不变。程序行如下所示：－－判断是否回到题库顶部

```
if getnum < =1 then
    SystemMessageBox ( WindowHandle, "
This is the first question!", "Warning", 48)
    getnum：=1
else
  getnum：=getnum -1
end if
－－设置返还数组，已得分扣除，否，不扣分
if pre[i -1] =1 then
  score：=score -1
  i：=i -1
else
  score：=score
  i：=i -1
end if
－－回到题库顶端，分数为0
if getnum =1 then
  score：=0
end if
－－返回前一题
GoTo(IconID@" showQuestions")
```

计算图标" Next" 用来进入下一题，程序行同" Previous" 类似，但需要考虑是否到了题库底部，是则予以提示，显示最后一题；否则进入下一题前对用户的选择做出判断，正确的予以加分。

计算图标" Exit" 用来强行退出听力考试系统，同时实现最后计分。在用户无意点击到该按钮时应予以警示。程序行如下所示：

SystemMessageBox (WindowHandle, " Are you

```
sure to exit the listening test? Click YES to exit and NO to continue." , "Warning" , 308)
    result: = score/30 * 100
    quit( )
```

2.2.1 初始化数据库

要想对数据库进行打开、执行、关闭操作必须导入外部函数 ODBC. u32(可在 Authorware 的安装目录中找到)中的三个函数 ODBCOpen( )、ODBCExecute( )和 ODBCClose( )。在 regDataSource 计算图标中接着输入以下内容(假定总题目数为 30,对话理解为 20 个,短文理解为两篇 10 个问题):

```
--初始化对话理解题目
if getnum < =20 then
    SQLString: = " select count( * ) from CONVERSATION;"
    max: = ODBCExecute(ODBCHandle,SQLString)
    timushu: =1
    repeat while timushu < =20
      i: = Random(1,max,1)
      SQLString: = " update conversation set flag =1 where sn = "^i" ;" --标志选中题目
      ODBCExecute(ODBCHandle,SQLString)
      timushu: = timushu +1
    end repeat
    SQLString: = " select SN from conversation where FLAG =1;"
    conSoundData: = ODBCExecute(ODBCHandle, SQLString) --返回对话听力文件名
    SQLString: = " select CHOICE_A, CHOICE_B, CHOICE_C, CHOICE_D, RIGHT_ANS from CONVERSATION where FLAG =1;"
    UserData: = ODBCExecute(ODBCHandle, SQLString) -返回所有选中的题目
    SQLString: = " update conversation set flag = 0 where flag =1;"
    ODBCExecute(ODBCHandle, SQLString) --重新初始化数据库
    ODBCClose(ODBCHandle)—关闭数据库
    direction: = " Directions: You will hear 10 short conversations. Each will be read only once and followed by one question. Listen carefully and make you choice."
--初始化短文理解题目
else
    SQLString: = " select count( * ) from PASSAGE;"
    max: = ODBCExecute(ODBCHandle,SQLString)
    timushu: =1
    repeat while timushu < =2
      i: = Random(1,max,1)
      SQLString: = " update PASSAGE set flag = 1 where PASSAGE_NO = "^i" ;" --标志选中题目
      ODBCExecute(ODBCHandle,SQLString)
      timushu: = timushu +1
    end repeat
    SQLString: = " select distinct( PASSAGE_NO) from PASSAGE where FLAG =1;"
    pasSoundData: = ODBCExecute(ODBCHandle, SQLString) --返回短文听力文件名
    SQLString: = " select CHOICE_A, CHOICE_B, CHOICE_C, CHOICE_D, RIGHT_ANS from PASSAGE where FLAG =1;"
    UserData: = ODBCExecute(ODBCHandle, SQLString) --返回所有选中的题目
    SQLString: = " update conversation set flag = 0 where flag =1;"
    ODBCExecute(ODBCHandle, SQLString) --重新初始化数据库
    ODBCClose(ODBCHandle) --关闭数据库
    direction: = " Directions: You will hear 2 short passages. Each will be read only once and followed by five questions. Listen carefully and make you choice."
end if
```

2.2.2 播放声音文件

为实现随即播放 mp3 文件,需要导入外部函数 MP3Player. u32,首先先调用 tMsMP3CreatePlayer( )函数进行播放初始化,其次使用 tMsMP3LoadFile ( )函数打开要播放的 mp3 文件,然后使用 tMsMP3Play ( )函数进行播放。

```
if getnum < =20 then
```

```
    j:=1
    repeat while j<=20
      sounfile:=getline(conSoundData,j)
      (WindowHandle,"Listening Test")
      tMsMP3LoadFile(FileLocation^"\sound\conversation\"^soundfile^".mp3")
      tMsMP3Play(0,0)
      j:=j+1
    end repeat
  else
    j:=1
    repeat while j<=2
      soundifle:=getline(pasSoundData,j)
      tMsMP3CreatePlayer(WindowHandle,"Listening Test")
      tMsMP3LoadFile(FileLocation^"\sound\passage\"^soundfile^".mp3")
      tMsMP3Play(0,0)
      j:=j+1
    end repeat
  end if
```

2.2.3 显示问题

在计算图标 A 中输入以下内容：

```
Checked@"A":=1
Checked@"B":=0
Checked@"C":=0
Checked@"D":=0
if right="A" &Checked@"A"=1 then
  scoreflag:=1
else
  scoreflag:=0
end if
```

则当用户点击单选按钮 A 时，用来判断对错。计算图标 B、C、D 的内容依次类推。

打开 showQuestions 计算图标，输入以下内容，显示问题，供用户选择：

```
--使加分标志复置为0
scoreflag:=0
--取下一题
question:=GetLine(UserData,getnum)
question:=Replace(Tab,"\r",question)
display:=Array(0,4)
display[1]:=GetLine(question,1)
display[2]:=GetLine(question,2)
display[3]:=GetLine(question,3)
display[4]:=GetLine(question,4)
right:=GetLine(question,5)
```

打开交互图标 choice，使用文字工具按图 3 的格式输入内容：

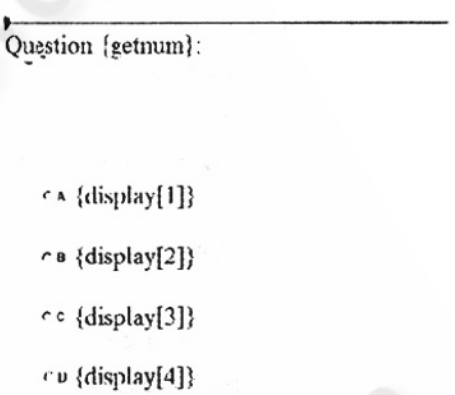

图 3

2.3 成绩显示

在显示图标 end 中用文字工具在合适位置输入 {ending} 用来显示结束语和最终得分。

3 结束语

本文提出的实现方法秉承 Authorware 的强大功能，克服了困扰了听力自测题软件开发中听、做不同步的缺点，真正实现了随机出题，为实现无纸化考试、无监考考试提供了技术支持。

参考文献

1 姜永生，基于数据库与 Authorware 的练习型多媒体课件制作[J]，福建电脑，2007，(2)：153－154.

2 李敏、尹先文，基于 B/S 的在线考试系统设计[J]，兵工自动化，2006，(10)：47－48.

3 周国强、吴新玲，在 Authorware 中实现多媒体数据库的构造与查询[J]，多媒体技术，2004，(1)86－87.