

基于微软 Speech API 的车站自动播音系统设计与实践

Design and Application of Station Automatic Broadcasting System Based on Microsoft Speech API

李忠延 (中国海洋大学 潍坊交运公司潍坊汽车总站 山东 潍坊 261041)

彭玉忠 (潍坊医学院计算机教研室 山东 潍坊 261041)

摘要: 本文阐述了一个自动播音系统的设计与实现过程。该系统基于微软 Speech API(简称 SAPI)接口设计,能够根据动态检索的信息实时播音;自动加入 XML 标记使播音精确化;播音与音乐合成,通过自动音量控制,产生具有背景音乐的播音效果。实践证明,基于 SAPI 接口开发自动播音系统是现实可行的。它能够满足一般情况下播音的需要。自投入使用以来,该系统运行稳定可靠,效果良好。

关键词: 自动播音 精确播音 Microsoft Speech API TTS XML 查询

1 引言

目前 TTS 文本语音合成技术已趋成熟,发音水平有了显著进步。微软 Speech API 由于实用化程度较高,得到了广泛应用。遵循微软 Speech API 接口规范的第三方产品给了我们更多的选择余地。

基于这样的原因,我们基于微软 Speech API 来实现车站的自动播音系统。本文给出了在开发自动播音系统过程,如何动态检索数据以形成播音文本^[1],如何通过 XML 标记使播音精确化,以及播音时产生背景音乐的原理。

2 系统要求、设计与开发环境

车站对自动播音系统的要求是:不需要过多的人工干预自动运行;每隔 10 分钟广播一次当前时间内即将发车的班次信息,提醒旅客及时检票乘车。空闲时间播放音乐,或播放其它文本信息。必要时可进行英文播音。

根据此要求,系统设计如图 1 所示。程序启动时分别创建两个线程:一个用于播放语音,该线程内有一个计时器用来检测距上次播音的间隔时间,如果已间隔 10 分钟,则播放班次信息,否则可以根据条件插播语音文件;另一个用于播放音乐。两个线程通过收发消息通讯。

2.1 系统设计

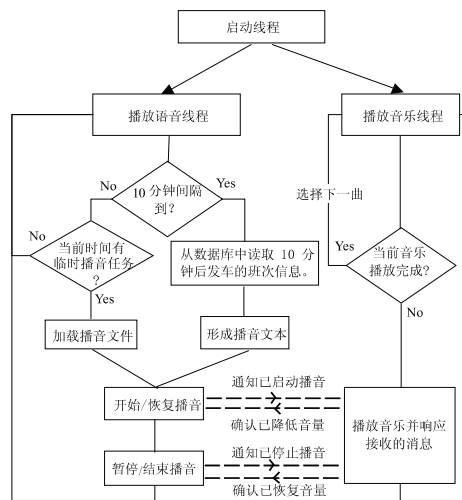


图 1 车站自动播音系统设计图

2.2 选择 TTS 产品

微软 Speech API 的 5.3 版比 5.1 版发音好,更成熟^[1]。但 5.3 版与 Vista 捆绑,不提供独立软件包。我们选择 NeoSpeech 公司的 TTS 语音引擎,该产品适用于多种 Windows 平台,发音效果很好,且支持 SAPI 编程接口。这样我们开发出的系统,就可以运行在多种 Windows 平台上。

2.3 建立开发环境的步骤

我们使用 Delphi2007 来开发自动播音系统。从微软网站下载并安装 SpeechSDK51.exe; 安装 NeoSpeech 的 VW Lily 和 VW Paul 中英文语音引擎; 向 Delphi2007 导入 Microsoft Speech Object Library 5.1 类型库。该类型库中的 TSpVoice 对象是实现语音 TTS 的核心对象^[2]。至此, 建立了完整的开发环境。

3 语音处理的主要流程^[2]

3.1 语音对象的创建和释放

3.1.1 创建语音对象

```
SpVoice := TSpVoice.Create(Self);
SpVoice.EventInterests := SVEAllEvents; // 监听语音事件
SpVoice.OnWord := SpVoiceWord;
SpVoice.OnStartStream := SpVoiceStartStream;
SpVoice.OnEndStream := SpVoiceEndStream;
// 获取所有语音, 随后可选择语音
SOTokens := SpVoice.GetVoices(, );
for I := 0 to SOTokens.Count - 1 do
begin
    SOToken := SOTokens.Item(I);
    cboVoices.Items.Add(SOToken.GetDescription(0));
end;
.....
```

3.1.2 释放语音对象

```
// 清除未完成的播音
SpVoice.Speak(, SVSFPurgeBeforeSpeak);
SpVoice.Free;
```

3.2 播音处理

3.2.1 定义成员变量和成员过程

成员变量 IsVoiceSpeaking 和 IsVoicePaused 用于指示播音状态; 成员过程 SetSpeakingState(bSpeaking, bPaused: Boolean); 根据传入的播音状态控制播音按钮, 并更新 IsVoiceSpeaking 和 IsVoicePaused, 该过程主要代码:

```
cmdVoiceStart.Enabled := True; // 开始按钮总是有效
cmdVoiceStop.Enabled := bSpeaking;
```

```
// 若正在播音, 未暂停, 则暂停按钮有效
cmdVoicePause.Enabled := (bSpeaking and not bPaused);
```

```
IsVoiceSpeaking := bSpeaking;
```

```
IsVoicePaused := bPaused;
```

3.2.2 cmdVoiceStart 按钮

```
// 若未播音、未暂停, 则从头播音
```

```
if not (IsVoiceSpeaking and IsVoicePaused) then
```

```
    SpVoice.Speak(text, SVSFlagsAsync or
```

```
        SVSFlsXML or SVSFPurgeBeforeSpeak);
```

```
if IsVoicePaused then SpVoice.Resume;
```

```
SetSpeakingState(True, False); // 设为正在播音, 未暂停
```

3.2.3 cmdVoicePause 按钮

```
SpVoice.Pause; // 暂停, 按 cmdVoiceStart 将恢复播音
```

```
SetSpeakingState(IsVoiceSpeaking, True); // 设置已暂停
```

3.2.4 cmdVoiceStop 按钮

```
SpVoice.Speak(, SVSFPurgeBeforeSpeak); // 清除播音
```

```
// 如果当前暂停, 则必须恢复。由于播音已被清除,
```

```
// 所以恢复后, SAPI 会立即停止播音。
```

```
if IsVoicePaused then SpVoice.Resume;
```

```
SetSpeakingState(False, False); // 设为未播音, 未暂停
```

4 语音与音乐合成

在车站广播中, 除了播音外, 还需要广播音乐。播音和音乐对象均采用异步播放方式^[3], 工作在各自的线程中, 互不影响。它们的音量可以独立设置。播音对象的音量使用 SpVoice.Volume 属性设置; 音乐对象的音量使用 mciSendCommand 函数设置。播音时将 SpVoice 对象音量设为最大, 而音乐对象的音量适当调小, 合成输出的音频就会产生带有背景音乐的播音效果。

由于两个对象分别要控制各自的音量属性, 只能通过该对象自身设置, 这就产生了线程间的通讯问题。程序采用 Windows 消息机制来进行两者的请求和确认, 从而使它们的工作状态同步。具体的处理过程不

再赘述。

5 语音精确化处理

在播音过程中,如果直接将文本发送给 SAPI,播音可能达不到理想的效果。为了使播音更精确,我们在以下几方面做了处理:

5.1 逐字朗读^[4]

车站的班次一般是数字或字母组合。SAPI 总是试图将其整体读出,而我们需要逐字符地朗读。为了能够控制每个字符的语速,采用 < SILENCE > 标记。比如:A7 次转换成:

```
< SILENCE MSEC = " 100" / > A
< SILENCE MSEC = " 100" / > 7
< SILENCE MSEC = " 100" / > 次
就能达到较好效果。
```

5.2 指定语音^[4]

播音时偶尔要插播一段文本,文本中可能既有中文也有英文。而播音开始前只能选择一种播音语言,这就不能很好的处理中、英文混合文本。为此在文本中加入 XML 标记强制一段文本使用指定语言播音:

```
< voice required = " name = VW Lily; Language = 804" > ... </voice > 包含的文本使用中文播音
< voice required = " name = VW Paul; Language = 409" > ... </voice > 包含的文本使用英文播音
```

5.3 多音字的处理^[5]

播音文本中有一些多音字地名,SAPI 有时发音会不准确,需要加以修正。为此,创建一个文本文件 VoiceCorrect.list 作为多音字替换发音规则列表,格式如下:

```
#源文本:替换文本:语言编码
单县:善县:ALL
.....
```

如果文本中出现“源文本”,则以“替换文本”代替;每行结尾的语言编码,可以填入 CN 或 EN 或 ALL。CN 表示中文,EN 表示英文,ALL 表示替换规则对所有语言的文本均有效。必要时,甚至可以在“替换文本”处填入文件名,使用预先录制的 wav 语音文件替代“源文本”发音。

5.4 中、英文播音^[6]

从数据库中读取的信息格式固定为:班次,终点

站,发车时间,里程,检票口,途径站点。根据这些信息很容易形成中文和英文播音文本。在用户选择播音语言时,自动切换播音文本。英文播音文本中的地名以中文地名的拼音字母替换。英文地名推荐使用小写字母,它比大写发音更准确些。

6 数据库访问设计

为从数据库中获取班次信息,需要查询几个相关的表,这些表及它们的关系如下图 2 所示。

一个班次运行在一条线路上,一条线路经过 N 个站点。为获取一个班次的途径站点,可以在存储过程中使用游标遍历的方式获取站点列表,但更好的办法是使用 SQL Server 的 XML 查询^[7],用一条 SQL 语句获取全部信息^[8]。

```
select a.run_no, c.name, a.start_time,
a.check_no,b.stations
from timetable a join
(select road_id, stations = (
stuff( ( select ',' + b.name
from road_station a join station b on a.station_id = b.id
where a.road_id = t.road_id for xml path(''),1,1,')
) b on a.road_id = b.road_id
join station c on a.station_id = c.id
from road_station as t
group by road_id
```

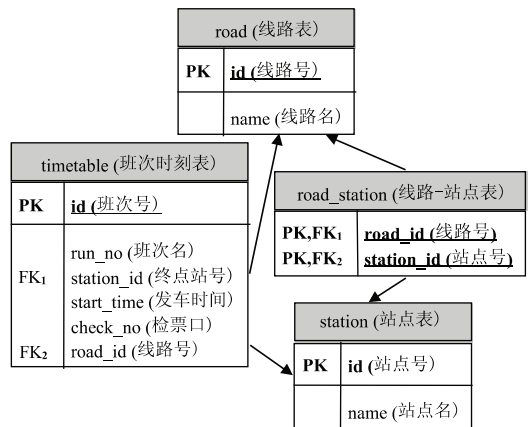


图 2 业务信息表及其关联关系

(下转第 179 页)

(上接第 79 页)

7 结束语

本文提出的基于微软 Speech API 的语音广播系统模型适用于车站、港口、机场等公共场所。这个系统模型具有一定的灵活性,既可以自动广播也可以人工控制。在一个相对特定的词语集合内,系统的语音表达可以很准确,从而达到使用要求。

参考文献

- 1 Shi H , Maier A. Speech - enabled windows application using Microsoft SAPI. IJCSNS International Journal of Computer Science and Network Security, 2006,6(9A): 33 -37.
- 2 Long B. Speech Synthesis & Speech Recognition Using SAPI 5.1. Borland UK, 2003.
- 3 易定. 用 Microsoft Speech SDK5.1 实现中文语音交互的方法. 电脑开发与应用,2005,18(4):62 -63.
- 4 Microsoft Corporation. Microsoft Speech SDK 5.1 Document. Microsoft Corporation,2001.
- 5 张力,薛惠锋,吴晓军,李憺. 中文 TTS 系统中多音字的一种解决方案. 计算机应用与软件,2008,25(2):143 -145.
- 6 吴兵,王艳萍. TTS 技术及其在英语多媒体教学中的应用. 中国现代教育装备,2006(3):28 -29.
- 7 李婷,吴业福,涂平晖,张恒喜. 关系数据库与 XML 数据融合的应用研究. 计算机与数字工程,2008,36(2):52 -54,94.
- 8 Microsoft Corporation. SQL Server 2005 联机丛书. Microsoft Corporation,2005.