

資料

心理学におけるExcel VBAの利用 その3
— 音声認識ソフトウェアの応用 —

久 本 博 行

Applications of Excel VBA in Psychology (3)
Application for Speech Recognition Software

Hiroyuki HISAMOTO

Abstract

When psychological experiments are run using computers, most reactions are input from the keyboard. However, there are a lot of experiments whose reactions are obtained by voice. Speech recognition software was applied to psychological experiments, and methods of measuring reactions by voices were shown.

Key words: Speech Recognition, Microsoft Speech API, Julius, Excel VBA, Programming

抄 録

コンピュータで心理学実験を行う場合、反応の多くがキーボードからの入力となる。しかし、声による反応が重要な要素である実験も多い。ここでは、心理学実験に音声認識ソフトウェアを応用し、声による反応をとる方法が示された。

キーワード：音声認識, Microsoft Speech API, Julius, Excel VBA, 心理学実験

1. はじめに

1.1 音声認識とは

音声認識とは、人間の話す音声をコンピュータに入力し、それを文字列に変換することです。コンピュータ上でこのような処理を行うソフトウェアを、音声認識エンジンと呼んでいます。ここでは音声認識エンジンについての詳細な解説は行ないませんが、音声認識エンジンを利用する上で必要な簡単な概説を述べておきます。音声認識は、声の音響的な特徴を表現した音響モデルと、話の言語的な性質を表現した言語モデルを基礎としており、まず言語モデルをもとに現れる単語の候補を挙げます。たとえば、文の初めであれば「まず」や「そして」といった言葉が多く使われ、鉄道の切符の電話予約であれば、駅名の後に「から」や「まで」といった言葉が多く使われるというようなことから単語の候補を挙げるのです。その単語の候補と音響モデルによってどのような音声が発せられたかという音声との照合を行い、認識結果を確定しているのです（河原，荒木（2006），荒木（2007））。

1.2 心理学実験における音声認識の必要性

現在、コンピュータによる心理学実験が普及してきており、それに関連した文献も多く出版されてきています（北村・坂本 2004，水野 2004）。そして、それらの実験プログラムでは被験者の反応は、全てキーボードからのキーの打鍵やマウスのクリックまたはドラッグで行われています。このような反応方法で問題のない課題も多いのですが、課題によってはひじょうに不自然な反応方法となるものもあります。例えば、久本（2007）ではストループ効果の実験プログラムを示しましたが、そのプログラムでは色を応えるのにキーボードから数字を入力するのです。赤を応えるときは1，黄は2，緑は3，青が4といった具合に入力するのです。水野（2004）でもストループ効果の実験が取り上げられていますが、そのプログラムでは、赤がJ，青がK，緑がLのキーを押すことによって反応をとっています。しかし、これは自然な反応のとらえ方ではないのです。本来ストループ効果は、文字そのものの色と文字が示している色との違いが、色の名前を声に出す際、脳の言語処理系で発生する干渉によっておこると考えられます。したがって、声に出して色の名前をいうことに非常に意味があるので、キーボードによる実験でも現象を起こすことはできますが、やはりあくまでも代替手段にしかすぎないのです。また、このような実験では被験者は色と数字や英字の対応を覚えなくてはならず、さらにキーボードの操作になれていない被験者は、キーボードのキーの位置も覚えなくてはならないのです。したがって、色と数字や英字との対応を記憶すること

やキーボードに慣れているかどうかによって反応時間が左右されることになります。

このように、音声による応答が不可欠な実験に対して、これまでは代替的な方法しかとられていなかったのですが、音声認識エンジンを実験用ソフトに組み込み音声で反応をとることができるようになることでよりよい実験が行なえるようになるのです。

ここでは、音声認識をExcel VBAで利用する方法を示します。

音声認識エンジンは、いろいろなコンピュータメーカーから出ていますが、フリーウェアとして手軽に利用できるものには、Microsoft社のSpeechSDK5.1と京都大学や名古屋工業大学、奈良先端技術大学院大学、情報処理振興事業協会が共同で開発しているJuliusがあります。最初にMicrosoft社のSpeechSDK5.1について述べ、つぎにJuliusについて説明します。

2. Microsoft Speech SDK5.1

ここでは、初心者でも使いやすいMicrosoft Speech SDK5.1の使い方を紹介します。なお、以下の解説はWindowsXPでの操作です。WindowsVistaでの操作方法については、最後に触れます。

2.1 ダウンロードとインストール

WindowsXPでは、標準で音声認識ソフトがインストールされていますが、アプリケーションを開発する場合は、Speech SDK5.1をダウンロードしインストールする必要があります。

下記のMicrosoft社のページからSpeech SDK5.1をダウンロードします。


<http://www.microsoft.com/downloads/details.aspx?FamilyId=5E86EC97-40A7-453F-B0EE-6583171B4530&displaylang=en>


上記のページを開くと図2.1のようなダウンロードファイルの一覧が出てきますのでそのうち、SpeechSDK51.exe, SpeechSDK51LangPack.exe, sapi.chm, の3つをダウンロードします(図2.1の下線部)。SpeechSDK51.exeは本体のソフトで、SpeechSDK51LangPack.exeは多言語用パッケージ, sapi.chmはヘルプ・ファイルです。

Files in This Download		
The links in this section correspond to separate files available in this download. Download the files most appropriate for you.		
File Name:	File Size	
msttsc22L.exe	2.0 MB	Download
<u>sapi.chm</u>	<u>2.3 MB</u>	Download
Sp5T1IntXP.exe	3.5 MB	Download
<u>SpeechSDK51.exe</u>	<u>68.0 MB</u>	Download
<u>SpeechSDK51LangPack.exe</u>	<u>81.5 MB</u>	Download
speechsdk51msm.exe	131.5 MB	Download

図 2.1 ダウンロードファイル

ダウンロードしたSpeechSDK51.exeをダブルクリックし解凍します。そして、解凍されたフォルダの中のsetup.exeをダブルクリックしインストールします。次にSpeechSDK51LangPack.exeをダブルクリックし解凍し、setup.exeをダブルクリックしインストールします。これで、インストールは完了です。

次に環境設定を行います。コントロール・パネルを開き、「サウンド、音声、およびオーディオデバイス」 をクリックし開きます。

さらに「音声認識」 をクリックし音声認識のプロパティを開きます。

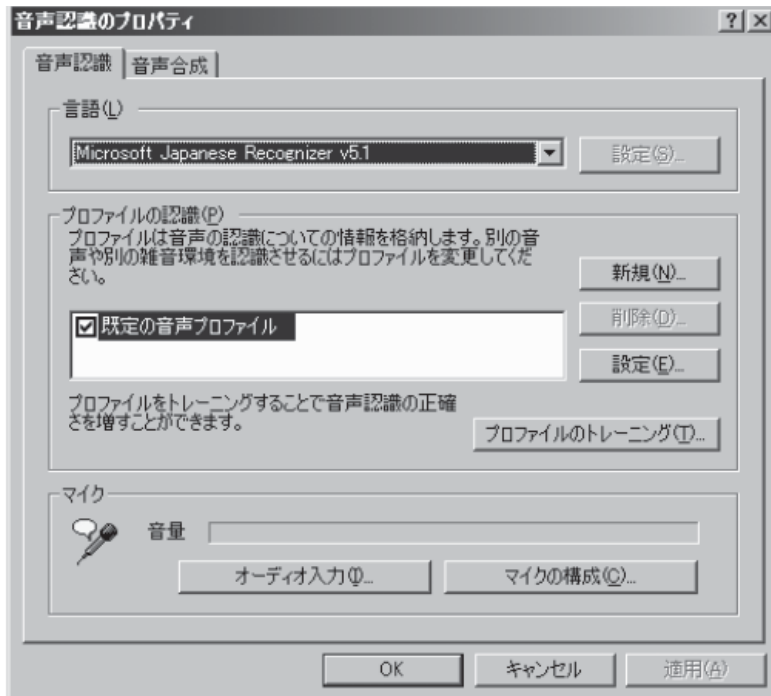


図 2.2 音声認識のプロパティ

「音声認識」タブの言語で「Microsoft Japanese Recognizer v5.1」を選択します。SpeechSDKのインストールが正しく行われている場合は、すでに「Microsoft Japanese Recognizer v5.1」が選択状態になっています。

2.2 プログラムの作成

SpeechSDKを利用したプログラムを作成する場合、認識する言葉や文を定義した文法ファイルを作成することと応用プログラムを作成することの2つの作業が必要になります。それでは、まず文法ファイルを作ってみましょう。

2.2.1 文法ファイルの作成

文法ファイルを作る場合は、ワードパッドやメモ帳などのテキストエディタを使って作成します。文法ファイルはXML形式のファイルですので、保存するときの拡張子は「.xml」です。XMLは、Webページを作成するときに使われているHTMLと同じマークアップ言語の一つです。HTMLとの最も大きな違いはタグが固定ではなく、ユーザーがタグを作成することができるという点です。

```
<GRAMMAR LANGID="411">
  <RULE NAME="Number" TOPLEVEL="ACTIVE">
    <L PROPNAME="Digit">
      <P VAL="1">/1/いち/イチ;</P>
      <P VAL="2">/2/に/ニ;</P>
      <P VAL="2">/2/に/ニー;</P>
      <P VAL="3">/3/さん/サン;</P>
      <P VAL="4">/4/よん/ヨン;</P>
      <P VAL="4">/4/し/シ;</P>
      <P VAL="5">/5/ご/ゴ;</P>
      <P VAL="5">/5/ご/ゴー;</P>
      <P VAL="6">/6/ろく/ロク;</P>
      <P VAL="7">/7/なな/ナナ;</P>
      <P VAL="7">/7/しち/ヒチ;</P>
      <P VAL="7">/7/しち/シチ;</P>
      <P VAL="8">/8/はち/ハチ;</P>
      <P VAL="9">/9/きゅう/キュー;</P>
      <P VAL="9">/9/く/ク;</P>
      <P VAL="10">/終り/おわり/オワリ;</P>
      <P VAL="11">/赤/あか/アカ;</P>
      <P VAL="12">/青/あお/アオ;</P>
      <P VAL="13">/黄色/きいろ/キイロ;</P>
      <P VAL="13">/黄/き/キ;</P>
    </L>
  </RULE>
</GRAMMAR>
```

図 2.3 文法ファイルのサンプル

図 2.3 に文法ファイルのサンプルを示しています。これをもとに文法ファイルを説明していきます。

1 行目の GRAMMAR タグは、文法ファイルの始まりを示します。そして、LANGID は、認識する言語の ID で日本語の場合は 411 で、アメリカ英語では 409 になります。認識する言語によって値は変わります。

この音声認識システムは、複雑な連続音声認識のために作られたもので、そのため複数の文法規則を記述でき、それぞれの規則に名前をつけて識別できるようになっています。ここでは、RULE NAME タグで名前をつけておき、その規則を ACTIVE にしておき

ます。RULE NAME=の後に“ ”で括って名前を付けます。

タグLはLISTの略で、認識される語句のリストの始まりを示します。PROPNAMEはPropertyの名前です。これも=の後に“ ”で括って名前を付けます。

タグPはPhraseの略で、ここで認識すべき言葉を一つ一つ定義していきます。VALは認識が成功した場合、プログラムに返される値を示します。例えば、<P VAL="4">/4/よん/ヨン;</P>と次の<P VAL="4">/4/し/シ;</P>ではVAL=“4”で同じ値がプログラムに返されます。つまりこれは、4を「ヨン」と発音しても「シ」と発音しても同じ4を返すということを示しています。もし、数値ではなく認識結果を文字で返してほしい場合は、VALではなくVALSTRタグを使います。例えば、「黄」と言ったときにも「黄色」と言ったときにも「黄色」という文字を返してほしい場合は、<P VALSTR="黄色">/黄/き/キ;</P><P VALSTR="黄色">/黄色/きいろ/キイロ;</P>とすれば、どちらでも「黄色」という文字列が返ってきます。

<P VAL="1">/1/いち/イチ;</P>の中で「/1/いち/イチ;」の部分は、「/画面表示形/辞書登録形/発音形;」となっています。画面表示形は通常画面に表示されているその文字の形です。辞書登録形は辞書に登録されている読みです。そして、発音形はそれが発音される読みです。簡単な数字や文字は、画面表示形のみを指定しても認識してくれますが、認識精度をあげるには発音形まで指定した方がよいでしょう。「/9/きゅう/キュー」や「/水泳/すいえい/スイエー」のように発音形を記述する時は、実際の発音にできるだけ近い形で記述した方が認識精度をあげることができるようです。

このようにして文法ファイルを作成します。一通りできあがると文法ファイルに間違いがないかどうかチェックします。プログラムを作成し、うまく動かない場合、プログラムに問題があるのか、文法ファイルが間違っているのか分からないと問題の解決に時間をとってしまいます。ですから、文法ファイルはきちんとチェックしてから使いましょう。そこで、文法ファイルをチェックするためにGrammar Compilerというツールを使います。このツールは、SpeechSDK5.1をインストールすると一緒にインストールされているので、特に新たに用意する必要はありません。

Grammar Compilerで文法ファイルをチェックするのですが、その前にマイクを接続し、マイクの音量もチェックしておいてください。マイクの音量は、図2.2の音声認識のプロパティでチェックできますので、コントロール・パネルからもう一度音声認識のプロパティを開いて、マイクの音量をチェックしておいてください。

Grammar Compilerの起動はWindowsXPでは、[スタート] ボタンをクリックし、[すべ

でのプログラム] → [Microsoft Speech SDK 5.1] → [Tools] → [Grammar Compiler] の順にポイントし、クリックします。

そして、Grammar Compilerが起動したら、[File] メニューから [Open] をクリックし先ほど作成した文法ファイルを指定します。

文法ファイルの誤りを見つける場合は、[Build] メニューから [Compile] を選びます。

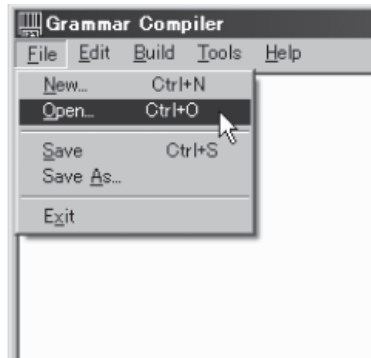


図 2.4 文法ファイルを開く

そうすると、文法のチェックが行われ、正しい場合は「Compile Successful」というメッセージが出てきます。エラーが出た場合は、エラーのある行の番号が表示されますので、それを修正します。

文法のチェックだけでなく、実際に音声認識をしてどのような値が返ってくるか、といったことまでテストしたい場合は、[Tools] メニューから [Compile and Test Grammar] をクリックします。

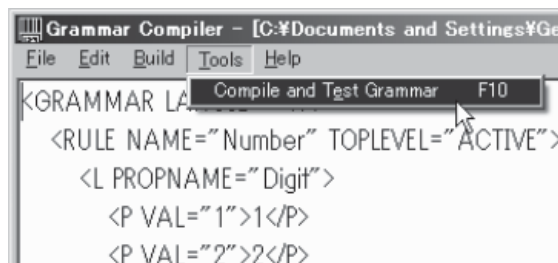


図 2.5 文法ファイルの誤りの検査と認識テスト

Test Grammarを使うと、認識した文字やそのときに返される数値なども分かります。

[Compile and Test Grammar] を選ぶと下図の画面が表示されます。[Begin Recognition] にチェックを入れると音声認識が開始され、何が認識されたかが表示されます。

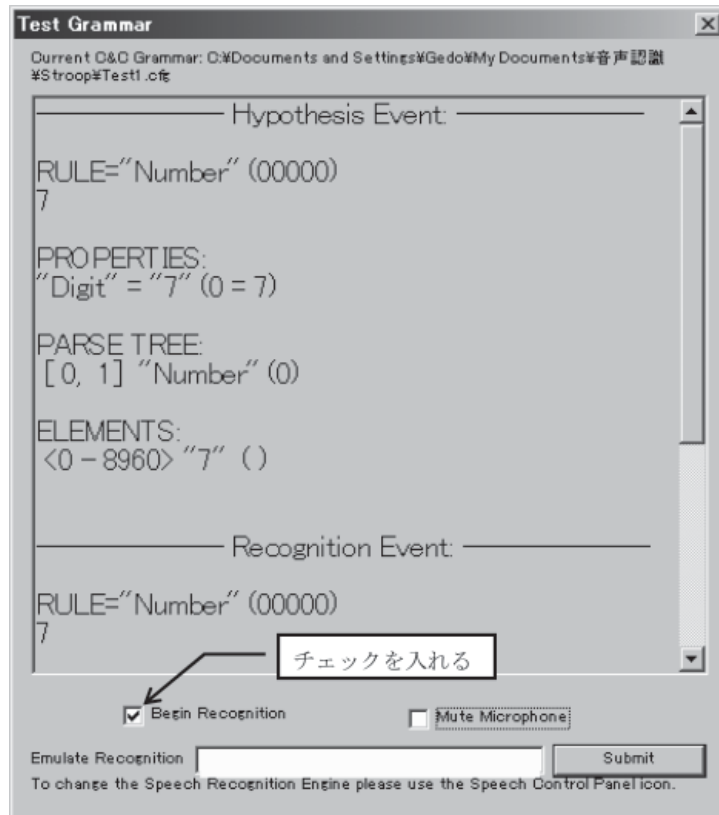


図 2.6 認識テスト

2.2.2 Excel VBA サンプルプログラムの作成

ここでは、簡単なExcel VBAによるサンプルプログラムを作ってみましょう。このプログラムは、1～9までの数字を表示し、その数字をユーザーが読み上げ、その音声を認識した結果と読み上げ時間を表示するというものです。

制御の大まかな流れは、次の3つの段階になっています。

- (1) 文法ファイルを読み込む
- (2) 音声認識を開始する
- (3) 音声認識が成功した場合のイベント・プロシージャで、音声入力データを受け取る。

はじめにExcel VBAでSpeech SDK 5.1を利用するための設定をおこないます。

ExcelVBAのVisual Basic Editorを開き、[ツール] メニューから [参照設定] をクリックします。

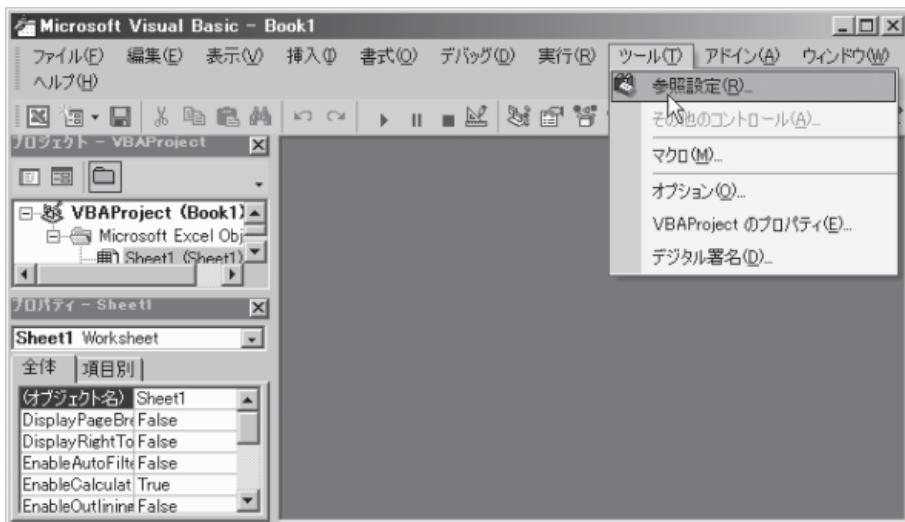


図 2.7 参照設定

参照設定のダイアログ・ボックスで [Microsoft Speech Object Library] にチェックを入れ、[OK] ボタンを押します。

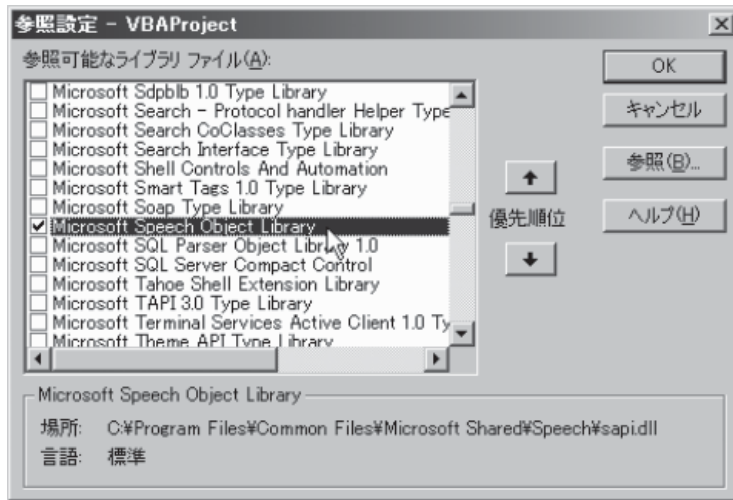


図 2.8 参照設定ダイアログ・ボックス

次に下図のようなFormを作ります。このFormはLabelが9つ貼ってあり、Label 1 には読み上げる数字を表示します。Label 2 には音声認識が成功したか、エラーであったかを表示します。Label 3 は認識に成功した場合、文法ファイルで定義されている値が返ってきますので、それを表示します。Label 4 は、実際に認識できた言葉を、Label 5 は、数字を表示してから、言葉を認識し始めるまでの時間を表示します。



図 2.9 サンプルのForm

```

1: Option Explicit
2: ' 音声認識のイベントと文法のための宣言
3: Public WithEvents Speech As SpeechLib.SpSharedRecoContext
4: Public RcgGrammar As SpeechLib.ISpeechRecoGrammar
5: ' 時間計測用関数の宣言
6: Private Declare Function GetTickCount Lib "Kernel32" As Long
7: Dim StartTime As Long ' 測定開始時刻
8: Private Sub UserForm_Initialize()
9:     Dim GrammarFile As String
10:    GrammarFile = ActiveWorkbook.Path & "\Test1.xml"
11:    ' 音声認識準備
12:    Set Speech = New SpeechLib.SpSharedRecoContext
13:    Set RcgGrammar = UserForm1.Speech.CreateGrammar
14:    ' 文法ファイルの読み込み
15:    RcgGrammar.CmdLoadFromFile GrammarFile, SpeechLib.
        SpeechLoadOption.SLOStatic
16:    ' 音声認識開始
17:    RcgGrammar.CmdSetRuleIdState 0, SpeechLib.SpeechRuleState.
        SGDSActive
18:    ' 刺激文字の表示
19:    Label1.Caption = Int( Rnd() * 9 ) + 1
20:    ' 反応時間の測定開始
21:    StartTime = GetTickCount()
22: End Sub

```

図 2.10 UserForm_Initializeのコード

プログラムの3行目は、Speechというオブジェクト型変数をイベント発生が検知できるオブジェクトとして宣言するためWithEventsキーワードを使っています。イベントの処理を行うイベント・プロシージャは「Speech_イベント名」という名前になります。

また、このプログラムでは時間計測用の関数としてWindows API のGetTickCount関数を使用しています。これは、Speech SDKから返される音声認識の開始時間がTickCountですので、それに合わせて刺激提示時刻をTickCountで得るためです。この関数はシステムが起動されてから、現在までの経過時間をミリ秒単位で取得することができます。GetTickCount関数は、VBAの標準の関数ではありませんので、使う前にどのライブラリ

にあり、引数や戻り値の型は何かを宣言しておく必要があります。プログラムの6行目はGetTickCount関数を宣言しているところです。

10行目は文法ファイルのパスを定義しているところです。文法ファイルのパスは、絶対パスでないとエラーになります。ここでは、このExcelファイルと同じフォルダにあるという形で定義しています。ActiveWorkbook.Pathは、Excelファイルのパスです。それと、Test1.xmlというファイル名を結合し、文法ファイルのパスを作っています。

12行目は、3行目で宣言されたSpeechという変数の実体を作っているところです。

13行目では、SpeechオブジェクトのCreateGrammarメソッドを使って、ISpeechRecoGrammarオブジェクトを生成しています。このオブジェクトは、文法を生成したり、文法を起動したりする役割をしています。

15行目は文法ファイルを読み込んでいます。CmdLoadFromFileメソッドは、文法ファイルを指定されたファイルから読み込むものです。パラメータはファイル名とオプションとしてSLOStaticかまたは、SLODynamicを指定することができます。実行時に文法規則を修正したり変更したりする場合はSLODynamicを指定し、それ以外の場合はSLOStaticを指定します。省略した場合は、SLOStaticが指定されたものとみなされます。ですから、このパラメータは省略しても構いません。

```
ISpeechRecoGrammar.CmdLoadFromFile(
    FileName As String,
    [LoadOption As SpeechLoadOption = SLOStatic]
)
```

図2.11 CmdLoadFromFileメソッドの書式

17行目はCmdSetRuleIdStateメソッドを使って文法規則を起動し、音声認識を開始させています。CmdSetRuleIdStateメソッドは文法規則を起動するだけでなく、終了させることもできます。これにはパラメータが2つあり、最初はRuleIdでこれは文法規則の中のどの規則を起動するかをIdで指定します。ここでは0としています。RuleIdが0の場合は、全てのTopLevelでかつActiveな規則が対象となります。2番目のパラメータはStateで文法規則を起動するのか、あるいは終了させるのかといった状態を指定します。このパラメータで指定できる状態にはSGDSInactive、SGDSActive、SGDSActiveWithAutoPause、SGDSActiveUserDelimitedの4つがあります。

SGDSInactive：起動している文法規則を終了し、音声認識を終わらせる。

SGDSActive：文法規則を起動し、音声認識を開始させる。

SGDSActiveWithAutoPause：文法規則を起動し、音声認識を開始させるのですが、音声認識が成功すると、音声認識エンジンを一時停止状態にする。

SGDSActiveUserDelimited：ユーザーの発話によって音響部を開始し、発話の終了によって音響部を終了させる。

```
ISpeechRecoGrammar.CmdSetRuleIdState(  
    RuleId As Long,  
    State As SpeechRuleState  
)
```

図 2.12 CmdSetRuleIdState メソッドの書式

19行目は乱数によってLabel 1 に 1 から 9 の文字を表示しているところです。21行目は GetTickCount () によって刺激提示時間を測定しています。

次に音声を知識できた場合や、知識できなかった場合の処理を記述します。Speech SDKでは、こうしたイベントが18個用意されています。ここでは、音声知識エンジンが知識できた場合に発生する Recognition イベントと、知識に失敗した場合に発生する FalseRecognition イベントを利用します。

Recognition イベントは、文法ファイルに従った音声知識に成功した場合に発生するイベントで、そのイベントのパラメータは図 2.13に示すものです。

```
SpeechRecoContext.Recognition(  
    StreamNumber As Long,  
    StreamPosition As Variant,  
    RecognitionType As SpeechRecognitionType,  
    Result As ISpeechRecoResult  
)
```

図 2.13 Recognition イベントのパラメータ

StreamNumberは認識できた音声の番号を示します。StreamPositionはその音声の中の位置を示します。RecognitionTypeは認識結果がどのような方法で得られたものであるのか、あるいは音声認識エンジンの状態を示すものです。RecognitionTypeにはSRTStandard, SRTAutopause, SRTEmulatedの3つがあり、SRTStandardは認識結果が標準の音声認識で得られたものであることを示しており、SRTAutopauseはSRTStandardと同様標準の音声認識ですが認識した後、音声認識エンジンが一時停止状態になっていることを示しており、SRTEmulatedはエミュレーションによる音声認識を行った場合を示すものです。

Resultは、音声認識の結果を返すオブジェクトです。このオブジェクトには、4つのプロパティと5つのメソッドがありますが、ここではPhraseInfoとTimesの2つのプロパティを使っており、この2つについて説明しておきます。

PhraseInfoプロパティは、最後に認識した語句についての詳細な情報を持っているプロパティです。PhraseInfoプロパティのPropertiesのItemのValueには、認識できた語句の文法ファイルでVALあるいはVALSTRで割り当てられている値が入ってきます。また、PhraseInfoのGetTextメソッドで実際に認識できた文字列を取得できます。

Timesプロパティは4つのプロパティを持っていますが、そのうちのTickCountプロパティにシステムが起動してから、今の認識結果の始まりまでの経過時間があります。

FalseRecognitionは、音声認識が失敗した場合に発生するイベントで、そのイベントのパラメータは図2.14に示すものです。Recognitionイベントとほぼ同じで、違いはRecognitionTypeがないことです。

```
SpeechRecoContext.FalseRecognition(
    StreamNumber As Long,
    StreamPosition As Variant,
    Result As ISpeechRecoResult
)
```

図2.14 FalseRecognition イベントのパラメータ

イベント・プロシージャ名は、「オブジェクト名_イベント名」となるので、Recognition イベントと FalseRecognition イベントを受けるイベント・プロシージャ名は、図 2.10 の 3 行目で宣言した Speech というオブジェクト変数とイベント名を_ でつないだ形になり、それぞれ Speech_Recognition と Speech_FalseRecognition になります。

```

1: ' 音声認識による応答の処理
2: Public Sub Speech_Recognition(ByVal StreamNumber As Long, ByVal
   StreamPosition As Variant, _
3:     ByVal RecognitionType As SpeechLib. SpeechRecognitionType, _
4:     ByVal RecoResult As SpeechLib.ISpeechRecoResult)
5:     Label2.Caption = "成功" ' ステータスを表示
6:     Label3.Caption = RecoResult.PhraseInfo.Properties.Item(0).Value ' 文法ファ
   イルで定義された値を表示
7:     Label4.Caption = RecoResult.PhraseInfo.GetText ' 認識された言葉を表示
8:     Label5.Caption = RecoResult.Times.TickCount - StartTime ' 反応時間を表示
9:     If RecoResult.PhraseInfo.Properties.Item(0).Value = 10 Then End ' 「終り」
   という言葉であれば終了する
10: ' 刺激文字の表示
11: Label1.Caption = Int(Rnd() * 9) + 1
12: ' 反応時間の測定開始
13: StartTime = GetTickCount()
14: End Sub
15: ' 音声認識の失敗
16: Private Sub Speech_FalseRecognition(ByVal StreamNumber As Long, ByVal
   StreamPosition As Variant, _
17:     ByVal RecoResult As SpeechLib.ISpeechRecoResult)
18:     Label2.Caption = "エラー"
19:     Label3.Caption = ""
20:     Label4.Caption = RecoResult.PhraseInfo.GetText
21:     Label5.Caption = RecoResult.Times.TickCount - StartTime ' 反応時間を表示
22: End Sub

```

図 2.15 UserForm_Initialize のコード

Recognition と FalseRecognition のイベント・プロシージャは、図 2.15 に示すものです。プログラム 6 行目の「RecoResult.PhraseInfo.Properties.Item(0).Value」は文法ファイ

ルで「VAL=」という形で定義した値が返され、それをLabel 3に表示しています。

プログラム7行目の「RecoResult.PhraseInfo.GetText」は実際に認識できた言葉を取得し、それをLabel 4に表示しています。

プログラム8行目の「RecoResult.Times.TickCount」はミリ秒単位でシステムが起動された時から、現在認識できた言葉の始まりまでの経過時間を示しています。それから、StartTimeを引いたものが反応時間となり、それをLabel 5に表示しています。

プログラム9行目は、図2.3の文法ファイルで「終わり」という言葉の値を「VAL=“10”」としていますので、10という値が返ってきた場合は「終わり」と言われたので、プログラムを終了させています。

プログラム11行目と13行目は、次の刺激の提示と反応時間の測定を開始しているところです。

プログラム16行目以降のFalseRecognitionのイベント・プロシージャはRecognitionのイベント・プロシージャとほとんど同様の形になっています。

3. WindowsVistaでの音声認識

WindowsVistaにはMicrosoft音声認識エンジン8.0が組み込まれています。これは、Speech API5.3というライブラリで利用できます。5.1にいくつかの機能が追加されていますが、5.1のプログラムもほとんどそのまま動きます。ですから、上述のプログラムもSpeech API 5.3でも修正せずに動かすことができます。

しかし、ライブラリがインストールされていないので、それをインストールしてやる必要があります。Speech API5.3はWindows SDKの中に入っているので、Windows SDKをインストールします。

Windows SDKのインストール方法には二通りあり、1つはWebからインストールする方法で、1つはDVDイメージをダウンロードし、DVDにイメージを焼いた上でDVDからインストールする方法です。それぞれの方法についてURLを下記に示しました。

Webからのインストール

<http://www.microsoft.com/downloads/details.aspx?familyid=E6E1C3DF-A74F-4207-8586-711EBE331CDC&displaylang=en>

DVDイメージのダウンロード

<http://www.microsoft.com/downloads/details.aspx?FamilyId=F26B1AA4-741A-433A-9BE5-FA919850BDBF&displaylang=en>

Windows SDKのインストールが終われば、後の手順は5.1の場合と同じです。コントロール・パネルを開いて音声認識の設定をする必要はありません。また、Grammar Compilerは、5.1のようなGraphical User Interfaceのものは用意されていません。Grammar Compilerを使う場合は、コマンドラインからgc.exeを起動してコンパイルする必要があります。

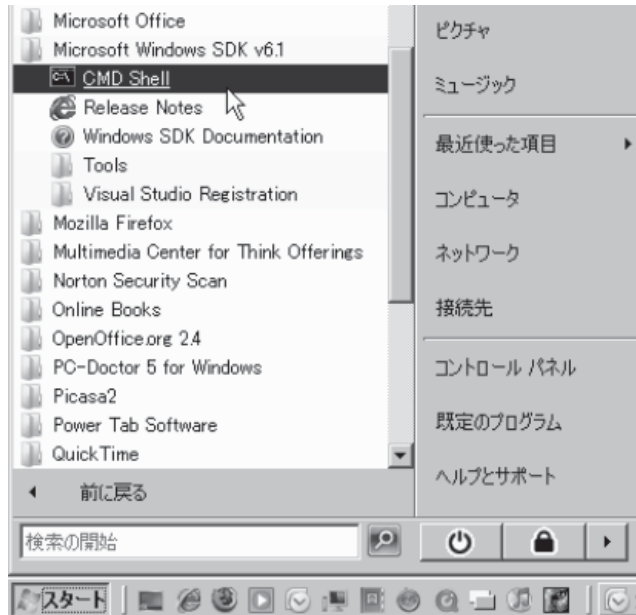


図 2.16 コマンド・ウィンドウの表示

gc.exeを起動するには、[スタート] ボタン → [すべてのプログラム] → [Microsoft Windows SDK v6.1] → [CMD Shell] の順にポイントし、クリックします (図 2.16)。

次に、コマンドラインのプロンプトの後に

gc 文法ファイル名

と入力します。

エラーがなければ、何も表示されませんが、エラーの場合は、

Compilation failed : エラーコード

という形で表示されます (図 2.17)。

図 2.17の①行目と③行目は、gc.exeを起動しているところです。②行目は正常終了したところです。④行目は、エラーが出たところです。

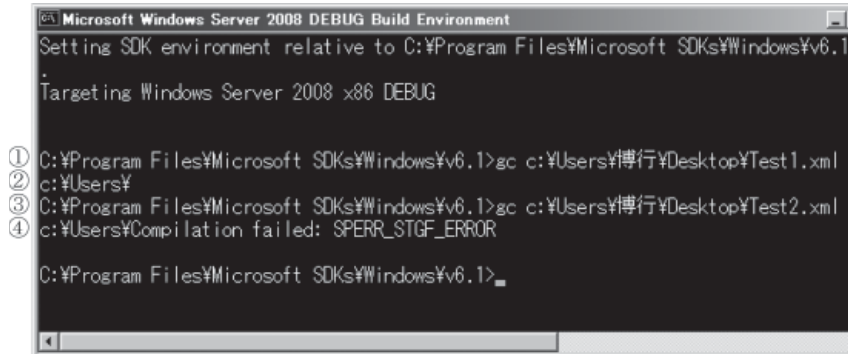


図 2.17 コマンド・ウィンドウ

Excel VBAの参照設定で「Microsoft Speech Object Library」を参照可能にしておくのは、5.3の場合と同じです。

4. 音声認識ソフトウェア Julius

これは先述のように、京都大学や名古屋工業大学、奈良先端技術大学院大学、情報処理振興事業協会が共同で開発しているフリーソフトウェアです。JuliusにはUnix版、Windows版、クロスプラットフォーム版などいろいろなOSに対応したものがありますが、ここではJulius for SAPIについて簡単に解説します。

Julius for SAPIは、Microsoft Speech APIを利用する音声認識エンジンでMicrosoft Speech APIを利用して作られたソフトウェアはそのまま動作します。Microsoft Speech APIを利用するので、最初にSpeech SDK5.1をインストールしておく必要があります。インストール手順の詳細は、下記のJuliusのサイトに詳しく解説されていますのでそれを参照してください。

<http://julius.sourceforge.jp/index.php?q=sapi/index.html>

音声認識の精度ではJuliusの方がMicrosoft Speech APIよりもよいように感じますが、認識に時間がかかるので、反応時間の測定に非常に高い精度が要求されるような実験には不向きだと思います。

5. その他

実際に音声認識を行う場合は、入力用のマイクが必要になります。マイクは1,000円程

度のもので十分に利用できます。マイクをコンピュータに接続した時は、かならずコントロール・パネルの音声認識プロパティで適切な音量で収録できているかどうかチェックしてから実験を行うようにしてください。

また、実際に音声認識をしてみると分かりますが、ちょっとしたノイズもマイクが拾って認識エラーを起こすことがありますので、静かな場所で実験する必要があります。うまく認識できない場合は、マイクを指向性が高く、かつ良いマイクにしてみてください。

「えー」、「あっ」、「うーん」といった間投詞などが多い場合、そうした言葉を文法ファイルに登録しておいて、それらを認識した場合は無視するようなことも一つの方法と考えられます。

参考文献

- 荒木雅弘 (2007) フリーソフトでつくる音声認識システム 東京 森北出版
- 久本博行 (2006) 心理学におけるExcel VBAの利用 その1 —VBAの基本文法— 関西大学社会学部紀要 38 (1) 191-221
- 久本博行 (2007) 心理学におけるExcel VBAの利用 その2 —ストループ効果の実験プログラム— 関西大学社会学部紀要 39 (1) 61-96
- 河原達也, 李晃伸 (2005) 連続音声認識ソフトウェアJulius 人工知能学会誌 20 (1) 41-49
- 河原達也, 荒木雅弘 (2006) 音声対話システム 東京 オーム社
- 北村英哉, 坂本正浩編 (2004) パーソナル・コンピュータによる心理学実験入門: 誰でもすぐのできるコンピュータ実験 京都 ナカニシヤ出版
- 水野りか (2004) Webを介してできる基礎・認知心理学実験演習 39-42 京都 ナカニシヤ出版

—2008. 8 .20受稿—