

User Interface Requirements

In order to use the Mp3-4-Me service, the user makes a phone call to an automated attendant similar to a telephone banking service. For this product, the interaction is minimal since only one piece of information is required: the radio station in question. The user dials mobile phone a predefined number for the Mp3-4-Me service, say MP3 (*673). The service immediately greets the user and asks for the radio station, providing a sample of what the user should say. The user's utterance is recorded, and the transaction is complete. This product has no learning curve, but there are slight restrictions to what it can perform. For example, radio station aliases such as "Energy" for 107.9 FM in Toronto will not be recognized if the user says "Energy". Also, the grammar file created to recognize station frequency phrases consider only conventional phrase types such as "ten fifty", "ninety eight point one", "one oh two point one", in addition to single digit utterances. Unconventional phrasing such as "ten four point five" are excluded from the grammar set.

Demonstration UI Development

The demonstration UI was developed to convey the flexibility of our product's architecture. Although it was not implemented, the UI was made to accommodate multiple phone ownerships where the demo would be run multiple times with separate phone numbers. This would illustrate how the intelligent network components work together to access crucial personal profile information.

The demonstration interface was implemented in Visual Basic for Applications (VBA). This was chosen because it was easy to code and to maintain. As a front-end to our core engine, its main interaction with the Mp34me service involves executing two series of batch files.

The first is executed when the phone is 'powered on', and activates the Nuance license manager service, the Nuance grammar recognition server, and the Nuance resource manager. The license manager activates the voice-recognition services and the recognition server loads our specific Mp34me grammar set. Finally, the resource manager is a necessary Nuance service running in the background.

The second set of batch files is executed when the user successfully makes a call to '*673', the Mp34me service number. Here, the command to start the Mp34me system processor is executed, the application that performs the speech recognition is started, and an SQL command refreshes a database for the simulated radio database. The database is updated with the latest timestamp with which to generate a radio song playlist that contains song data for the approximate timeframe of the call.

Nokia WAP Toolkit was first investigated for developing the UI, but further research revealed that system function calls were not permitted using WML (wireless markup-up language), nor with WML Script. So, the graphic from the toolkit was exported for our phone graphic, and the phone functionality was implemented using

drawing objects in VBA. The code for the UI was kept simple and usable to cater to product demonstration requirements.

The following figures illustrate a sample user test-case and the phone interface used in our product demonstration.

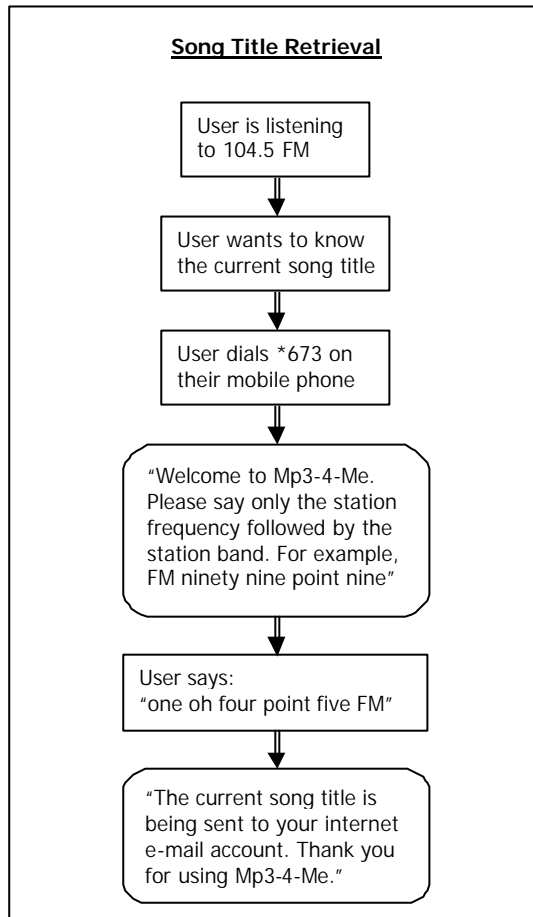


Figure 1 - Sample Use-Case



Figure 2 - Phone UI in VBA

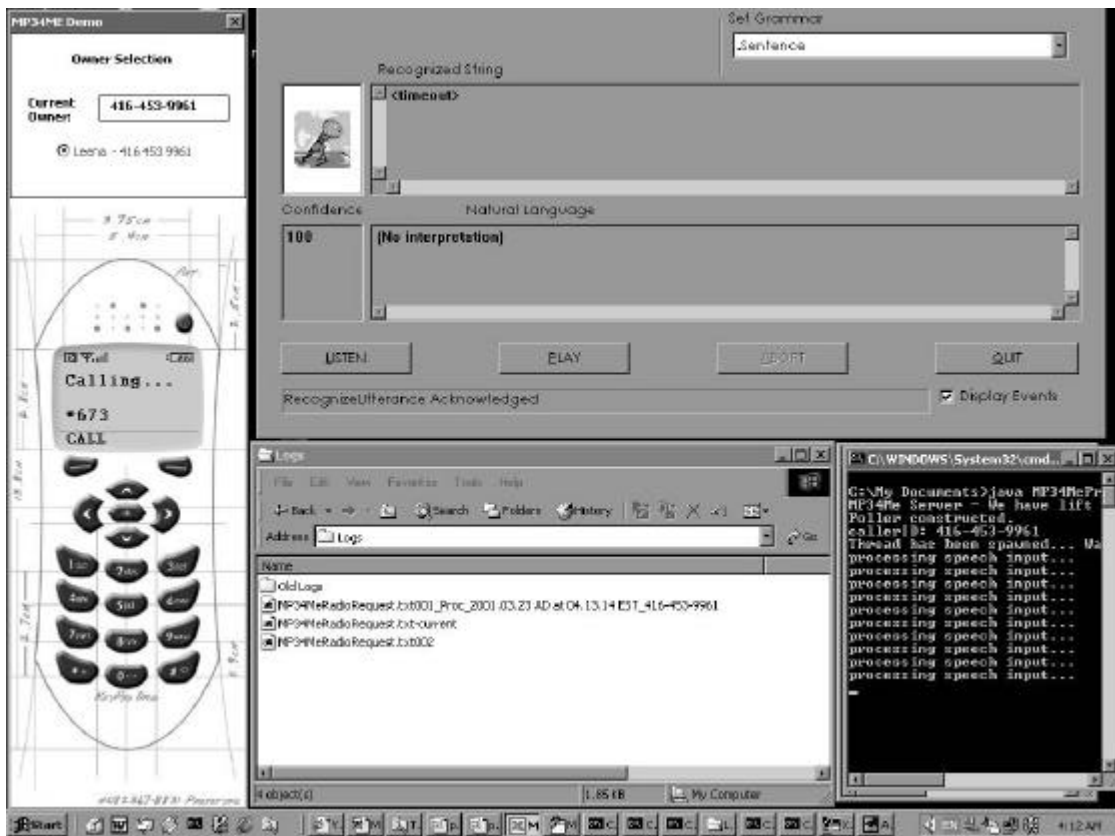


Figure 3 - Demonstration Screenshot

VBA Code Sample from mp34medemo.xls

COMMENT: Rewrites the batch executables for the MP34me process

```
Sub CreateAfile()
    Set fs = CreateObject("Scripting.FileSystemObject")
    If fs.fileexists("c:\mp34me\ui\go1.bat") Then
        FileSystem.Kill ("c:\mp34me\ui\go1.bat")
    Else
    End If
    Set a = fs.CreateTextFile("c:\mp34me\ui\go1.bat", True)
    a.WriteLine ("java MP34MeProcessor " & txtPhoneID.Text)
    a.Close

    Set fs = CreateObject("Scripting.FileSystemObject")
    If fs.fileexists("c:\mp34me\ui\go2.bat") Then
        FileSystem.Kill ("c:\mp34me\ui\go2.bat")
    Else
    End If
    Set a = fs.CreateTextFile("c:\mp34me\ui\go2.bat", True)
    a.WriteLine ("xapp -package c:\mp34me\test\mp34me")
    a.Close

```

End Sub

COMMENT: Clears the main screen for text entry to place a call

```
Private Sub Menucheck()
    If left(txtmain.Text, 1) <> "*" Or left(txtmain.Text, 1) = "E" Then
        txtmain.Text = ""
        txtaccept.Text = "CALL"
    End If

```

End Sub

COMMENT: Draw title screen of the phone UI

```
Private Sub Drawmain()
    txtmain.Text = txtPhoneID.Text & Chr(13) & " " & Date & Chr(13) & " " &
    Hour(Time()) & ":" & Minute(Time())

```

End Sub

COMMENT: Activates MP34me process when *673 is dialled

COMMENT: The process is executed with the batch files

```
Private Sub accept_Click()
    If Onstatus.Visible = False Then Exit Sub
    If txtaccept.Text = "CALL" And txtmain.Text = "*673" Then
        txtmain.Text = "Calling..." & Chr(13) & Chr(13) & "*673"
        Call CreateAfile
        MsgBox ("put batch file command code here.")
        Retval = Shell("c:\mp34me\UI\go1.bat", vbNormalFocus)
        Retval = Shell("c:\mp34me\UI\go2.bat", vbMinimizedNoFocus)
        Retval = Shell("c:\mp34me\UI\go3.bat", vbMinimizedNoFocus)
    Else
        txtmain.Text = "Enter *673 to call MP34ME..."
    End If

```

End Sub

COMMENT: The left arrow button on the phone UI

```
Private Sub left_Click()
    If (txtmain.TextLength > 0) And (left(txtmain.Text, 1) <> "-") Then

```

```
        txtmain.Text = left(txtmain.Text, txtmain.TextLength - 1)
    End If
End Sub
```

COMMENT: The phone UI's power button, turns on the required Nuance services

```
Private Sub power_Click()
If Onstatus.Visible Then
    Onstatus.Visible = False
    txtmain.Text = ""
    txtaccept.Text = ""
    txtsoft1.Text = ""
    Call Park_Cursor
    Retval = MsgBox("CALL ENDED", vbOKOnly, "MP34ME Service")
Else
    Onstatus.Visible = True
    Drawmain
    Retval = Shell("c:\mp34me\LicenseManager.bat", vbMinimizedNoFocus)
    Retval = Shell("c:\mp34me\ResourceManager.bat", vbMinimizedNoFocus)
    Retval = Shell("c:\mp34me\Recserver.bat", vbMinimizedNoFocus)
    Call Park_Cursor
End If
End Sub
```

COMMENT: take the cursor position away from the main textbox

```
Private Sub Park_Cursor()
    txtPhoneID.SetFocus
End Sub
```

COMMENT: Fills in phone number in the case of multiple numbers

```
Private Sub step3_Click()
    txtPhoneID.Text = "416-555-9961"
End Sub
```

COMMENT: UI activation procedures to set the phone's number and Park_Cursor

```
Private Sub UserForm_Activate()
    step3.Value = True
    Call Park_Cursor
End Sub
```

COMMENT: Mappings for all keys on the keypad are similar to zero_Click()

```
Private Sub zero_Click()
    If Onstatus.Visible = False Then Exit Sub
    Call Menucheck
    txtmain.Text = txtmain.Text & "0"
End Sub
```