

語音表格產生器之研究

葉耀明、楊繡如

國立台灣師範大學資訊教育系

ymyeh@ice.ntnu.edu.tw、u87307@ice.ntnu.edu.tw

摘要

一直以來，人們都是以視覺在瀏覽網頁並接收資訊，例如在網路上查詢考試榜單、瀏覽每日新聞、看餐廳或電影介紹、查詢股票報價等等。但是人們並非無時無刻都在電腦旁邊，當我們不在辦公室、不在家裡，卻又要接收資訊時，電話就扮演了一個既方便又普及的溝通角色了。隨著 3G 手機時代的來臨，目前手機業者也已體認電話帶給大眾生活的便利性，所以現在手機可使用的服務也越來越多，電腦已經不是上網瀏覽網頁、存取網路服務的唯一工具了。

本研究從聽覺的角度來考量，希望在沒有視覺的幫助下，光是靠聽與說，亦可以獲得原本應該以視覺得到的網頁資訊。在 HTML 網頁裡，表格內容一向是要親眼所見，才能夠清楚了解的資訊。故本研究由 HTML 表格資訊切入，研究並分析歸類網頁上的六種表格類型，根據每個類型設計不同的對話模式，並開發了將表格轉成 VoiceXML 格式的「語音表格產生器」(Voice Table Generator, 簡稱 VTG)系統，讓網頁設計者透過簡單的操作，就可以將視覺化之 HTML 表格轉換成語音表格，讓電話使用者將可藉由電話與語音平台對話互動，使一般網站上能夠看到的表格資訊，也可以在語音瀏覽器上以語音網站的方式來呈現給使用者。

關鍵詞： World Wide Web、Text to Speech、VoiceXML、XML。

1. 概述

網際網路的服務有很多種，其中，全球資訊網(WWW)算是最熱門也最普遍的一種，因為全球資訊網結合了文字、聲音、影像、電腦動畫等多媒體顯示效果，讓使用者可以很容易的在網路上存取各種多媒體資訊。

網頁(Web Page)，就是就是全球資訊網的顯示文件，透過瀏覽器(Browser)，使用者就可以看到網頁要呈現的豐富資訊。在編輯網頁的語言裡最常用的，莫過於 HTML(Hyper Text Markup Language)了。HTML 具有跨平台特性，使用者可以清楚地看到資訊在其豐富多樣的視覺效果下呈現。隨著網際網路的各種應用不斷發展，單純的 HTML 已不能滿足網際網路所有需求。XML(eXtensible Markup Language, 可擴充文件語言)是 W3C 在西元 1998 年二月訂定的新一代網頁語言【6】，我們可以將其稱為第二代的網頁語言以別於第一代網頁語言 HTML。XML 比 HTML 更具彈性，更容易添加新功能，但相對地也比 HTML 複雜。XML 至今誕生五年，已經幾乎成了網路上的世界語

言，也為下一波網路化的服務奠定根基。

VoiceXML 是 XML 的一種延展，代表人與電腦對話的一種標記語言，可讓人們利用語音與電話存取網路上的資訊與服務。如同 HTML 定義網路文字、影像的顯示與傳遞一般，VoiceXML 定義了使用者與網頁間的語音互動行為，使得網頁資訊可以透過電話傳遞。

目前世界科技演進的趨勢，是朝向「全方位(Ubiquitous)網路應用服務」的走向來發展，這個發展趨勢的願景是「Anytime, Anywhere, Anydevice」，也就是希望在任何時間、任何地方、使用任何裝置，都可以順利的上網獲取資訊。如圖 1 所示，位於伺服器內的所有資訊，在經過「過濾」與「轉換」的機制之後，就可以傳達給任何裝置，例如個人電腦、個人數位助理(PDA)、電話【3】等等【8】。而本研究的定位就是圖中的「轉換」模組，希望將 HTML 網頁上的資訊經過轉換後，可以讓使用者經由電話輕鬆的獲得資訊。

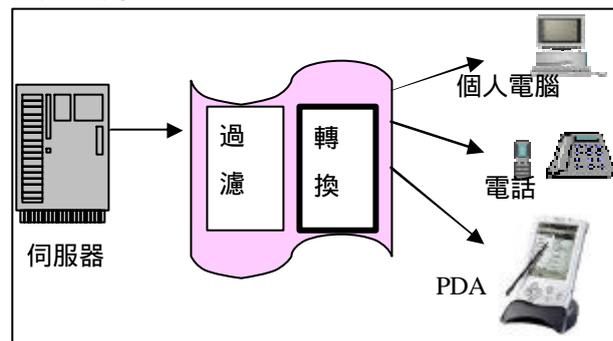


圖 1 「全方位應用軟體」概念圖

2. 文獻探討

2.1 XML 的語音技術：VoiceXML

在西元 1994 年，或者在更早之前，就有一些公司想研發出一套以純文字文件為基礎的標記語言，用以描述語音應用程式裡的對話流程。這些早期的語音標記語言的主要缺點是不夠完整，而且以前的語音應用程式是使用傳統的低層級(low-level)應用程式界面(Application Program Interface, API)，所以仍需精細的控制屬於應用程式底層的語音辨識系統【4】。

西元 1999 年三月，AT&T, IBM, Lucent, and Motorola 四個公司創立了一個名為「VoiceXML 論壇」【14】的業界組織，共同研究討論，在同年八月發表了 VoiceXML 0.9 規格書，之後又在 2000 年三月時發表了 VoiceXML 1.0 規格書，這份規格書後來在 2000 年五月經過 W3C 認可，公布成為 VoiceXML 1.0 標準

【15】。目前「VoiceXML論壇」的成員數已經成長到 300 餘家公司,並以開發、推廣 VoiceXML 為目的。後來 W3C 的 Voice Browser Working Group【18】繼續努力,制訂了目前的 VoiceXML 2.0【16】,以及目前仍是處於草案(Working Draft)階段的 VoiceXML 2.1【17】。

VoiceXML 的主要功能是用來設計語音瀏覽器上的使用者介面,以語音瀏覽器、聲音的輸出(語音合成或事先預錄的音檔)與聲音的輸入(使用者的聲音或按鍵音)來呈現,其中按鍵音即 DTMF (Dual Tone Multi-Frequency),就是我們使用一般電話的按鍵來輸入資訊。這些輸出與輸入的方式,需要語音辨識(Speech Recognition)與語音合成(Speech Synthesis)的技術來配合【9】。

一個完整的 VoiceXML 執行環境如圖 2 所示,電話使用者打電話到語音瀏覽器,等於是送出要瀏覽網頁的需求,於是語音瀏覽器會對網頁伺服器(Web server)請求某一個 VoiceXML 文件或者其他所需檔案,然後網頁伺服器會將這些檔案回應給語音伺服器,所以語音伺服器便可以取得所需的 VoiceXML 資訊、語音文法【11】、要播放的音檔等等,然後以此與使用者互動,讓使用者得到所需資訊;另外,在網頁伺服器端,可以與相連的資料庫系統互動,以存取需要的資料。【7】【10】

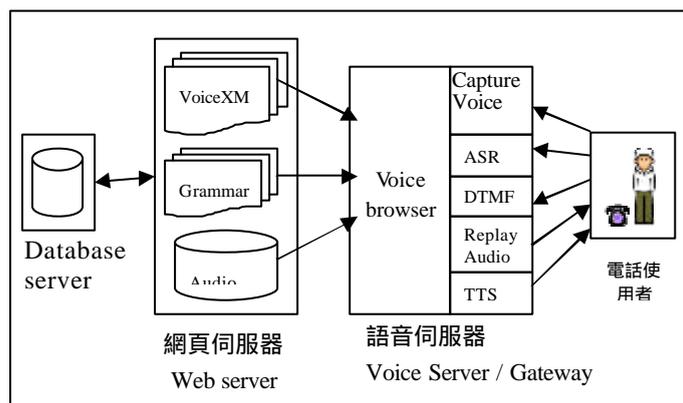


圖 2 VoiceXML 平台架構圖

2.2 網頁轉換成語音文件之實例探討

目前視障者若要聆聽一張網頁的內容,大都是使用螢幕閱讀機來閱讀網頁,螢幕閱讀機是屬於硬體的部分。若是就軟體的角度來看,與「語音瀏覽網頁」相關的軟體系統不多,在此我們探討 W3C 的資訊網可及性推動組織(Web Accessibility Initiative, WAI【20】)的「Tablin【12】【19】」。

Tablin 是由 WAI 組織的 Evaluation and Repair Tools Working Group【5】所發展出來的一個系統,主要是想要解決 HTML 表格資訊無法讓螢幕閱讀機正確閱讀的問題。站在視障者的角度來看,螢幕閱讀機是他們使用電腦的重要輔助工具,藉由聆聽螢幕閱讀機閱讀螢幕畫面內容,視障者就可以順利的使用電腦、上網瀏覽網頁。但是並不是每個網頁都很單純,

對於那些有表格內容的網頁,螢幕閱讀機由於無法判別閱讀的順序,所以還是會依照一般的順序,例如由左至右這樣的順序來閱讀,讓視障者很難了解網頁表格所要呈現的內容。Tablin 會根據表格的標題以及「行優先」或者「列優先」的選擇來生成一串描述表格全部內容的文字內容,螢幕閱讀機要閱讀表格時,就閱讀這一段文字,如此這個表格就比較容易被視障者了解,所以這個系統算是視障者的一大福音。但是 Tablin 只提供單方向播送表格的全部資訊給視障者的方式,視障者並不能選擇他們要聽的表格部分,只能把所有的表格內容聽完,雖然這樣比較容易了解表格的全貌,但若是表格太大、資料太多時,這樣的方式就顯得很沒效率,因為並非每個人都想聽全部的表格資訊內容。

以上介紹的這兩個系統各有其特色,也各有貢獻。但是我們可以看出來,對於 HTML 網頁表格資訊內容的無障礙化,要努力的地方還很多,還有很多的難題要挑戰。對於導讀網頁內容,目前已經有一些相關的研究【1】【13】,但是對於 HTML 網頁中的其他較複雜的部分,例如框架、表格資訊、表單資訊等部分,都還沒有比較理想的解決方式。

3. 語音表格產生器之設計

一個 HTML 網頁是由很多種類的標籤組成的,若要將網頁的資訊全部用語音輸出,而且要讓「聽」網頁的人都能將網頁的內容聽明白,實非易事。故我們在此從表格 <table></table> 標籤切入,試著開發的一套系統,稱為「語音表格產生器」(Voice Table Generator, VTG),探討如何將網頁的表格資訊用語音網頁的方式來輸出。

3.1 表格種類與對話模式

HTML 表格的面貌變化萬千,所以要研究如何用語音來呈現表格,就要先將目前在網頁上一般可以看到的表格分析與歸類。特別要附註的一點是,此處我們要探討的表格為用來呈現資料內容的表格,而不是用來排版網頁內容的表格。

經過我們的研究與分析,HTML 表格可以分為六種類型,分別是:以行為主型(Column-wise)、以列為主型(Row-wise)、行列型(Column & row)、跨行型(Column-span)、跨列型(Row-span)、跨行跨列型(Column-span & Row-span)等六種。以下將分別介紹這六種類型的表格。

1. 以行為主型(Column-wise):

如圖 3 所示,以行為主型的表格,標題在第一列,資料內容在其他列,經由標題與資料內容的配合,我們用眼睛就可以看出這樣的表格要傳達的資訊。

標題A	標題B
資料1	資料2
資料3	資料4

圖3 以行為主型表格示意圖

2. 以列為主型(Row-wise) :

如圖 4 所示，以列為主型的表格，標題在第一行，資料內容在其他行，經由標題與資料內容的配合，我們也是很容易就可以看出這樣的表格要傳達的資訊。以列為主型的表格與上述之以行為主型的表格都是屬於格式比較單純的網頁表格。

標題A	資料1	資料2
標題B	資料3	資料4

圖4 以列為主型表格示意圖

3. 行列型(Column & Row) :

如圖 5 所示，行列型的表格有兩類標題，分別是行標題與列標題，行標題位於第一列，列標題位於第一行，但是表格的最左上角那一格，並非行標題或列標題，在這種表格裡，會自動忽略最左上角那個欄位。其他的部分為資料欄位，藉由行標題與列標題的對應，我們可以很明白的看出這樣的表格要傳達的資訊。行列型的表格是一種在網路上最常見的一種表格。

	標題A	標題B
標題C	資料1	資料2
標題D	資料3	資料4

圖5 行列型表格示意圖

4. 跨行型(Column-span) :

如圖 6 所示，跨行型的表格，標題在第一列到第 N 列，資料內容在其他列，而 N 值是多少則因表格而異。跨行型表格的標題不只一列，而且某一列有跨行的現象，故名為跨行型表格。跨行型表格其實與「以行為主型」的表格在結構上相類似，差別僅在於是否跨行而已。

標題A	
標題B	標題C
資料1	資料2
資料3	資料4

圖6 跨行型表格示

5. 跨列型(Row-span) :

如圖 7 所示，跨列型的表格，標題在第一行到第 N 行，資料內容在其他行，而 N 值是多少則因表格而異。跨列型表格的標題不只一行，而且某一行有跨列的現象，故名為跨列型表格。跨列型表格其實與「以列為主型」的表格在結構上相類似，差別僅在於是否跨列而已。

標題A	標題B	資料1	資料2
	標題C	資料3	資料4

圖7 跨列型表格示意圖

6. 跨行跨列型(Column-span & Row-span) :

如圖 8 所示，跨行跨列型的表格，有兩類標題，分別是行標題與列標題，行標題位於第一列到第 N 列，列標題位於第一行到第 M 行，M, N 值是多少則因表格而異。但是在這兩種標題在表格左上角交錯的部分，則不屬於行標題或列標題。跨行跨列型表格的行標題不只一行，列標題不只一列，且某一行有跨列的現象，某一列有跨行的現象，故名為跨行跨列型表格。跨行跨列型表格其實與「行列型」的表格在結構上相類似，差別僅在於是否跨行跨列而已。

		標題A	
		標題B	標題C
標題D	標題E	資料1	資料2
	標題F	資料3	資料4

圖8 跨行跨列型表格示意圖

3.2 HTML 表格模式

當 VTG 系統讀進來一個 HTML 網頁表格時，會對這個表格作分析，取出表格資訊的部分，將表格內容以一個 XML 文件來記錄，在此我們將此文件稱為「表格索引 XML 文件」，在這個 XML 文件裡，除了記錄原來的 HTML 網頁表格的資訊外，還標明了某一項資料是位於表格的哪一個儲存格，所謂的儲存格，就是表格中每一個記錄資訊的單位格子。為了要清楚的標明每一個儲存格在表格裡的所在位置，我們將每一個儲存格都冠上一個「儲存格座標名稱」。直排儲存格，也就是行儲存格的名稱定為「\$c」，所以表示第一行的儲存格名稱為「\$c1」；而橫排儲存格，也就是列儲存格的名稱定為「\$r」，所以第一列的儲存格名稱為「\$r1」。舉例來說，例如有一個儲存格位於第三列第二行，那麼此儲存格的名稱為：「\$r3\$c2」。

無論是六種表格類型中的哪一種，在 VTG 系統讀進 HTML 文件之後，都會將 HTML 表格轉成「表格索引 XML 文件」，轉換的方法，大致上來說，都是按照 HTML 表格的結構為主，但是也會因為表格類型的不同有些微的差異。在此為了方便說明，我們

將第一、二、三種類型的表格，也就是以行為主型、以列為主型、行列型這三種的表格，歸類為第一大類，並且將第四、五、六種類型的表格，也就是跨行型、跨列型、跨行跨列型這三種的表格，歸類為第二大類。

第一大類的表格，會按照原本 HTML 語法內容的 <tr>與<td>標籤，直接對應到表格索引 XML 文件的 <tr>與<td>標籤，差別僅在於表格索引 XML 文件有<info>根元素及其 model、row_num、column_num 三種屬性，並且在<tr>與<td>標籤加入了 name 屬性。表 1 是第一大類的表格形成表格索引 XML 文件的對照表，本表格是一個「行列型」的網頁表格，資料內容為國文、數學、社會、自然的成績單，從對照表可以很明顯的看出左方的 HTML 文件內容與右方的表格索引 XML 文件的對比。另外同屬於這一大類的以行為主型、以列為主型網頁表格轉換成 HTML 文件內容時也是類似於這樣的方式。

HTML 文件	表格索引 XML 文件
<pre><table> <tr> <td></td> <td>國文</td> <td>數學</td> <td>社會</td> <td>自然</td> </tr> <tr> <td>王大頭</td> <td>80</td> <td>56</td> <td>87</td> <td>75</td> </tr> <tr> <td>陳美美</td> <td>74</td> <td>85</td> <td>76</td> <td>37</td> </tr> <tr> <td>張小猴</td> <td>54</td> <td>56</td> <td>100</td> <td>83</td> </tr> </table></pre>	<pre><?xml version="1.0" encoding="UTF-8" ?> <info model="column&row" row_num="8" column_num="5"> <tr name="\$r1"> <td name="\$c1" /> <td name="\$c2">國文</td> <td name="\$c3">數學</td> <td name="\$c4">社會</td> <td name="\$c5">自然</td> </tr> <tr name="\$r2"> <td name="\$c1">王大頭</td> <td name="\$c2">80</td> <td name="\$c3">56</td> <td name="\$c4">87</td> <td name="\$c5">75</td> </tr> <tr name="\$r3"> <td name="\$c1">陳美美</td> <td name="\$c2">74</td> <td name="\$c3">85</td> <td name="\$c4">76</td> <td name="\$c5">37</td> </tr> <tr name="\$r4"> <td name="\$c1">張小猴</td> <td name="\$c2">54</td> <td name="\$c3">56</td> <td name="\$c4">100</td> <td name="\$c5">83</td> </tr> </info></pre>

表 1 第一大類的表格索引 XML 文件的對照表

第二大類的表格，由於都有跨行或者跨列，所以和第一大類的表格有顯著的不同。因為跨行，所以每一列的儲存格數目就會不一樣，因為跨列，所以每一行的儲存格數目，也會不一樣。如此一來，要怎麼形成表格索引 XML 文件，且對每個儲存格都冠上一個儲存格座標名稱呢？解決的方式就是當跨行或者跨列的情形發生時，仍然將這個表格想像成每一列的儲存格數目一樣，每一行的儲存格一樣，原本跨行或者跨列的儲存格沒有跨行跨列，只是彼此相鄰的儲存格出現相同的標題而已，例如圖 9 所示，左方的標題 A

原本是跨行的標題，但是我們將這樣的表格想像成右邊表格一般，標題 A 在兩行都各出現一次。如此一來，對於第二大類的表格，我們可以對每個儲存格都冠上一個儲存格座標名稱，也就可以和第一大類表格一樣順利的產生表格索引 XML 文件了。

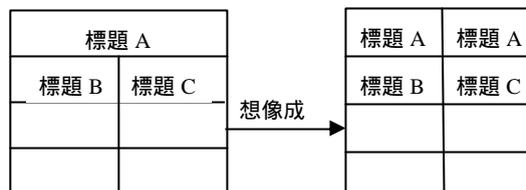


圖 9 跨行或跨列的表格處理方式示意圖

3.3 XML 式索引表格的閱讀模式

當 VTG 系統讀進來一個 HTML 網頁表格時，會對這個表格作分析，並生成「表格索引 XML 文件」，在這之後，VTG 系統會接著產生另一個 XML 文件，稱為「閱讀次序 XML 文件」，此 XML 文件的內容是記錄當生成 VoiceXML 檔案來閱讀該表格時，該如何來念這個表格，讓電話使用者能夠瞭解表格內的資訊。在這個文件中，是使用前面提過的「儲存格座標名稱」來記錄儲存格的閱讀順序，以及表格標題的所在位置。

圖 10 則是一個「行列型表格」的閱讀次序 XML 文件的一部份，由此畫面我們可以看出如果沒有表格索引 XML 文件的配合，單是看閱讀次序 XML 文件，我們是很難看出表格資訊的內容是怎麼被閱讀的。

```
<title name="row">$r2$c1</title>
<title name="row">$r3$c1</title>
<title name="row">$r4$c1</title>
<title name="row">$r5$c1</title>
<title name="row">$r6$c1</title>
<title name="row">$r7$c1</title>
<title name="row">$r8$c1</title>
</titles>
<seq>
<all>$r2$c1,$r1$c2,$r2$c2,$r1$c3,$r2$c3,$r1$c4,$r2$c4,$r1$c5,$r2$c5,$r3$c5,$r4$c1,$r1$c2,$r4$c2,$r1$c3,$r4$c3,$r1$c4,$r3,$r1$c4,$r5$c4,$r1$c5,$r5$c5,$r6$c1,$r1$c2,$r6$c2,$r1$c3,$r2,$r1$c3,$r7$c3,$r1$c4,$r7$c4,$r1$c5,$r7$c5,$r8$c1,$r1$c2,$r5,$</all>
<onecol>
<col name="$r1$c2">$r1$c2,$r2$c1,$r2$c2,$r3$c1,$r3$c2,$r4$c1,$r2,$r8$c1,$r8$c2,</col>
<col name="$r1$c3">$r1$c3,$r2$c1,$r2$c3,$r3$c1,$r3$c3,$r4$c1,$r3,$r8$c1,$r8$c3,</col>
<col name="$r1$c4">$r1$c4,$r2$c1,$r2$c4,$r3$c1,$r3$c4,$r4$c1,$r4,$r8$c1,$r8$c4,</col>
<col name="$r1$c5">$r1$c5,$r2$c1,$r2$c5,$r3$c1,$r3$c5,$r4$c1,$r5,$r8$c1,$r8$c5,</col>
</onecol>
<onerow>
<row name="$r2$c1">$r2$c1,$r1$c2,$r2$c2,$r1$c3,$r2$c3,$r1$c4,$r2,$r3$c1,$r1$c2,$r3$c2,$r1$c3,$r3$c3,$r1$c4,$r4$c1,$r4$c2,$r1$c3,$r4$c3,$r1$c4,$r5$c1,$r5$c2,$r1$c3,$r5$c3,$r1$c4,$r6$c1,$r1$c2,$r6$c2,$r1$c3,$r6$c3,$r1$c4,$r7$c1,$r7$c2,$r7$c3,$r7$c4,$r7$c5,</row>
```

圖 10 「閱讀次序 XML 文件」的範例

無論是六種表格類型中的哪一種，在 VTG 系統讀進 HTML 文件之後，都會將 HTML 表格轉成「表格索引 XML 文件」，以及「閱讀次序 XML 文件」，轉換成閱讀次序 XML 文件的方式，會依據表格類型而有些微的差異。接下來我們就針對各個表格類型說明其閱讀模式。

1. 以行為主型(Column-wise)：

圖 3 是一個以行為主型的表格示意圖，此表格的

兩個標題是「標題 A、標題 B」；唸某一欄(行)時，當電話使用者選擇念「標題 A」這一行，VTG系統會唸出「資料 1、資料 3」，選擇唸「標題 B」這一行，VTG系統會唸出「資料 2、資料 4」；若是要唸出表格的全部資訊，VTG系統則會唸出「標題 A、資料 1、標題 B、資料 2、標題 A、資料 3、標題 B、資料 4」。

2. 以列為主型(Row-wise)：

圖 4 是一個以列為主型的表格示意圖，此表格的兩個標題是「標題 A、標題 B」；唸某一列時，當電話使用者選擇念「標題 A」這一行，VTG系統會唸出「資料 1、資料 2」，當選擇唸「標題 B」這一行，VTG系統會唸出「資料 3、資料 4」；若是要唸出表格的全部資訊，VTG系統則會唸出「標題 A、資料 1、標題 B、資料 3、標題 A、資料 2、標題 B、資料 4」。

3. 行列型(Column & Row)：

圖 5 是一個行列型的表格示意圖，此表格的兩個行標題是「標題 A、標題 B」，兩個列標題是「標題 C、標題 D」；唸某一欄(行)時，當電話使用者選擇念「標題 A」這一行，VTG系統會唸出「資料 1、資料 3」，而選擇唸「標題 B」這一行，VTG系統會唸出「資料 2、資料 4」；唸某一列時，當電話使用者選擇念「標題 C」這一行，VTG系統會唸出「資料 1、資料 2」，當選擇唸「標題 D」這一行，VTG系統會唸出「資料 3、資料 4」；若是要唸出表格的全部資訊，VTG系統則會唸出「標題 C、標題 A、資料 1、標題 B、資料 2、標題 D、標題 A、資料 3、標題 B、資料 4」。

另外，這一類型的表格有一種有別於其他種類表格的查詢方式—「查某一筆資訊」，只要電話使用者說出行標題與列標題，就可以查詢到行標題與列標題對應的資訊，例如當電話使用者說出「標題 A 標題 C」或者「標題 C 標題 A」的時候，VTG系統會唸出「資料 1」，以此類推。

4. 跨行型(Column-span)：

圖 6 是一個跨行型的表格示意圖，此表格的兩個標題是「標題 A 標題 B、標題 A 標題 C」；唸某一欄(行)時，當電話使用者選擇念「標題 A 標題 B」這一行，VTG系統會唸出「資料 1、資料 3」，而選擇唸「標題 A 標題 C」這一行，VTG系統會唸出「資料 2、資料 4」；若是要唸出表格的全部資訊，VTG系統則會唸出「標題 A 標題 B、資料 1、標題 A 標題 C、資料 2、標題 A 標題 B、資料 3、標題 A 標題 C、資料 4」。

5. 跨列型(Row-span)：

圖 7 是一個跨列型的表格示意圖，此表格的兩個標題是「標題 A 標題 B、標題 A 標題 C」；唸某一列時，當電話使用者選擇念「標題 A 標題 B」這一行，VTG系統會唸出「資料 1、資料 2」，選擇唸「標題 A 標題 C」這一行，VTG系統會唸出「資料 3、資料 4」；若是要唸出表格的全部資訊，VTG系統則會唸出「標題 A 標題 B、資料 1、標題 A 標題 C、資料 3、標題 A 標題 B、資料 2、標題 A 標題 C、資料 4」。

6. 跨行跨列型(Column-span & Row-span)：

圖 8 是一個跨行跨列型的表格示意圖，此表格的兩個行標題是「標題 A 標題 B、標題 A 標題 C」，兩個列標題是「標題 D 標題 E、標題 D 標題 F」；唸某一欄(行)時，當電話使用者選擇念「標題 A 標題 B」這一行，VTG系統會唸出「資料 1、資料 3」，當選擇唸「標題 A 標題 C」這一行，VTG系統會唸出「資料 2、資料 4」；唸某一列時，當電話使用者選擇念「標題 D 標題 E」這一行，VTG系統會唸出「資料 1、資料 2」，選擇唸「標題 D 標題 F」這一行，VTG系統會唸出「資料 3、資料 4」；若是要唸出表格的全部資訊，VTG系統則會唸出「標題 D 標題 E、標題 A 標題 B、資料 1、標題 A 標題 C、資料 2、標題 D 標題 F、標題 A 標題 B、資料 3、標題 A 標題 C、資料 4」。

注意以上各種類型表格的閱讀模式中，標題會穿插在語句中重複出現，避免電話使用者不知道某個儲存格資料是位於哪一個標題之下，而讓資料內容難以被了解，如此就失去唸表格的意義了。

3.4 視覺呈現模式轉換聽覺呈現模式之機制

在 VTG 系統對一個 HTML 網頁表格生成了「表格索引 XML 文件」以及「閱讀次序 XML 文件」之後，VTG 系統會依據這兩個 XML 文件的內容，生成此表格的 VoiceXML 格式。

在一格表格的閱讀次序 XML 文件中，詳細記載了該表格的閱讀次序，所以只要將閱讀次序 XML 文件中每個「儲存格座標名稱」抓取出來，對應到表格索引 XML 文件裡某個儲存格的內容，就可以將閱讀次序 XML 文件中的儲存格座標名稱都轉換成表格資訊內容了。依據這些表格資訊，VTG 系統會生成這個 HTML 網頁表格的 VoiceXML 版本，藉以讓使用者打電話進來查詢表格內容資訊。

每種類型的表格轉成的 VoiceXML 版本對話模式都不盡相同，如圖 3-26 所示為「行列型表格」與「跨行跨列型表格」的對話流程。一進入表格之 VoiceXML 文件之後，系統會先播放系統歡迎訊息與提示使用者有哪些選項可以選，使用者選擇「念行標題」時，便會聽到行標題內容；使用者選擇「念列標題」時，便會聽到列標題內容；使用者選擇「念全部資訊」時，便會聽到表格的全部資訊；使用者選擇「念某一行時」，系統便會告訴使用者有哪些行標題可以選擇，當使用者選擇某一個行標題時，便會聽到該行的資訊；使用者選擇「念某一列」時，系統便會告訴使用者有哪些列標題可以選擇，當使用者選擇某一個列標題時，便會聽到該列的資訊。

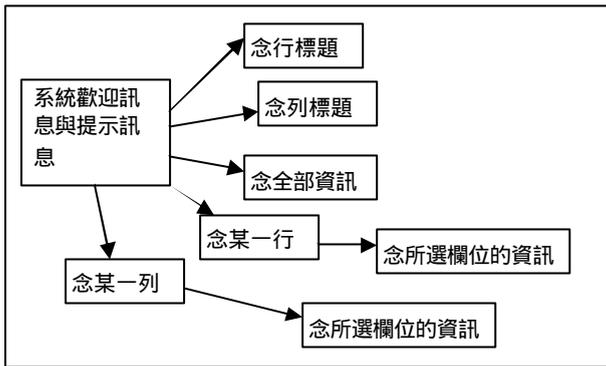


圖 11 「行列型表格」與「跨行跨列型表格」對話流程

4.系統架構與實作

4.1 VTG 系統架構

從 UML 的「使用案例觀點」來看，VTG 系統的使用者有兩個，一個是「網站設計者」，一個是「語音平台」。網站設計者可以上傳網頁表格檔、分析表格、將表格轉換成 VoiceXML 格式，而另外一個使用者，語音平台，則是可以使用 VTG 系統生成的 VoiceXML 語音網頁表格檔案，這些語音網頁表格檔案跟語音平台互相配合，就可以讓電話使用者打電話到語音網站了。圖 12 的使用案例圖就是描述了上述的情景。

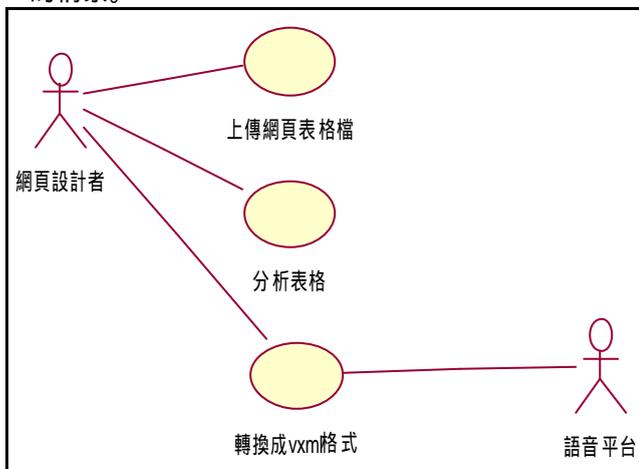


圖 12 VTG 使用案例圖

在 VTG 系統的內部結構中，有三個類別在相互合作，這三個類別分別是 MainClass，ParseTable，GenVxml 等等。MainClass 類別在 VTG 中，是扮演著指揮者的主程式角色，另外兩個類別 ParseTable 與 GenVxml，都要聽令於 MainClass。ParseTable 類別在意義上，是處理表格網頁的第一步：分析表格。一個 HTML 網頁表格根據使用者填入的類別經過 ParseTable 類別的分析後，就會生成該種類型表格的「表格索引 XML 文件」與「閱讀次序 XML 文件」，這兩個文件對於之後產生 VoiceXML 格式的網頁表格，可是非常重要的關鍵。GenVxml 類別在 VTG 系統裡的主要功能是根據「表格索引 XML 文件」與「閱讀次序 XML 文件」，將 VoiceXML 格式的網頁表格

產生出來。圖 13 是 VTG 系統的類別圖，在這個類別圖裡，我們可以看到這三個類別息息相關。

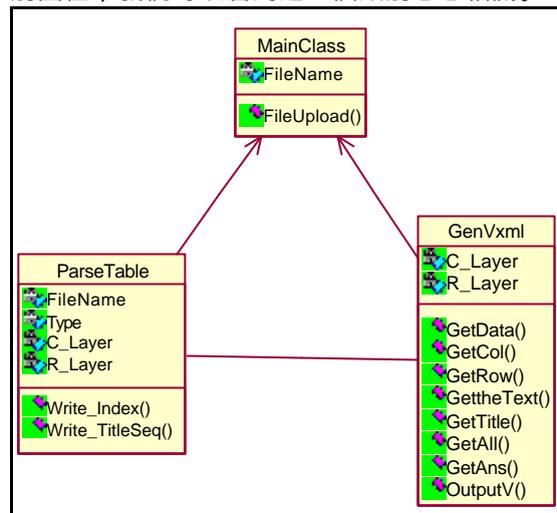


圖 13 VTG 類別圖

4.2 VTG 實作畫面

在 VTG 系統將一個 HTML 表格網頁轉換成 VoiceXML 格式時，第一個步驟，要填入 HTML 表格檔案本地端位址，並且選取這個表格是屬於六大類型中的哪一個類型，如圖 14 的範例所示是選取「行列型」表格。若表格是屬於跨行型、跨列型、跨行跨列型這三個類型時，還需填入標題的層數，這是因為表格標題有跨行或跨列的情形，標題就不會只有一層，也就是不會只有一行或一列。當步驟 1 中的資訊填妥後，按下『確定』按鈕。

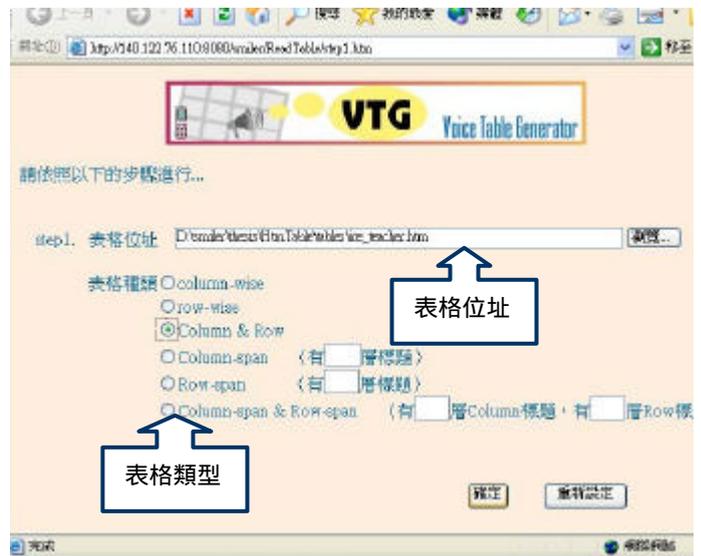


圖 14 VTG 系統步驟 1 畫面

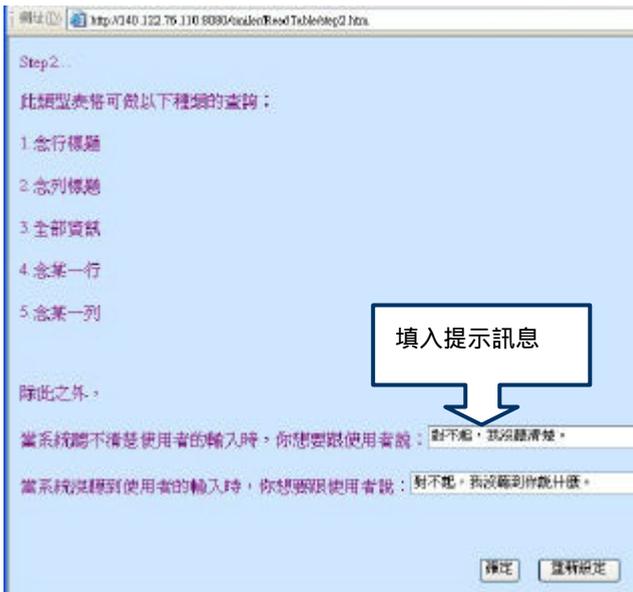


圖 15 VTG系統步驟 2 畫面

接下來來到步驟 2，如圖 15 所示，在這裡系統會告訴網頁設計者在步驟 1 選取的表格類型表格會有哪些查詢方式，並要求填入當系統無法辨識電話使用者輸入的語音時，要有什麼提示訊息。然後可以按下『確定』按鈕。

然後，VTG系統經過內部運作，就可以將原本的 HTML 格式的網頁表格轉換成一個 VoiceXML 檔案了。這個由 VTG系統生成的 VoiceXML 檔案，可以讓網頁設計者下載回去，並放置在語音平台中，這樣就可以給電話使用者使用了。VTG系統運作完成的畫面如圖 16 左圖所示，但若是表格類型為「行列型」表格的，完成畫面則如圖 16 右圖所示，這兩個畫面有些不同，是因為行列型的表格比別類型的表格多出一種特殊的查詢方式，所以要讓網頁設計者下載的檔案也比較多，共有三個檔案。而 VTG系統所生成的 VoiceXML 檔案，如圖 17 所示。



圖 16 VTG系統完成畫面



圖 17 VTG系統生成之語音網頁

5. 結論與未來發展

5.1 VTG 系統與傳統 IVR 之比較

傳統的語音按鍵系統(Interactive Voice Response, 簡稱 IVR 系統)亦有查詢資訊之功能，可以讓電話使用者打電話進去系統查詢資訊，但是設計一般 VoiceXML 系統的語音網頁檔，與使用 VTG 系統來生成語音網頁檔，和設計傳統 IVR 的語音查詢系統有很多不同的特點，如表 2 所示。

就語音播放來說，傳統 IVR 系統是使用預先錄製好的錄音檔，而 VoiceXML 系統與 VTG 系統不需事先預錄，使用 TTS 文字轉語音模組，可以即時將文字內容以語音播出。

就編寫程式的角度來說，傳統 IVR 系統需要為了每個 IVR 系統分別去編寫程式，VoiceXML 系統裡的每個 VoiceXML 檔案也都是需要去編寫的，而使用 VTG 系統來產生一個語音網站，就不需要編寫程式了，網站設計者只要遵循 VTG 系統的指示與流程，就可以順利產生一個語音網站。

就語音網站的開發速度來說，傳統 IVR 系統不但要自行編寫程式，而且程式裡面對於電話語音信號的處理等較低階的部分，全部都要一一寫程式來完成，所以開發的速度慢。VoiceXML 系統裡的 VoiceXML 網頁雖然也是要自行編寫，但是 VoiceXML 語言隱藏了低階的電話信號處理的部分，VoiceXML 設計者只要專心的就應用程式的動作、流程與語音介面來考量就好，所以開發的速度為中。而如果網站設計者使用 VTG 系統來開發一個語音網站，開發速度就算是快了，因為使用 VTG 系統不用

管低階信號處理的部分，也不需要編寫程式，所以網站設計者可以很快的將語音網站開發完成。

就通用性來說，傳統 IVR 系統是各家廠商獨立開發，系統不共享也不相通。而 VoiceXML系統與 VTG系統使用 W3C 所制訂之 VoiceXML標準，所以 VoiceXML格式的檔案可以跨平台，通用於所有的 VoiceXML語音平台上。

在與客戶系統的連通性上，傳統 IVR 系統通常是獨立的系統，不會與企業內部的系統做整合，而 VoiceXML系統與 VTG系統裡皆是使用 XML技術，若要與企業內部的系統整合，就相對的方便許多。

至於開發這些系統是否需要學會 VoiceXML呢？開發傳統 IVR 系統與使用 VXPB系統開發語音網站，都不需要學會 VoiceXML，而開發一般 VoiceXML系統裡的 VoiceXML檔案，需要親自編寫其流程與介面，所以一定要學會 VoiceXML語言才行。

	傳統 IVR 系統	VoiceXML系統	VTG 系統
語音播放	預先錄好的錄音檔	不需預錄。TTS 會自動文字轉語音播出	不需預錄。TTS 會自動文字轉語音播出
編寫程式	需編寫程式	需編寫 VoiceXML 檔案	不用編寫程式
語音網頁開發速度	慢	中	快
通用性	各家獨立開發，不共享	標準 VoiceXML 格式，跨平台。	標準 VoiceXML 格式，跨平台。
與客戶系統的連通性	獨立，難以連通。	使用 XML 技術，便於和客戶系統互連	使用 XML 技術，便於和客戶系統互連
是否需會 VoiceXML	不需要	需要	不需要

表 2 開發流程特性比較表

5.2 未來發展

在本研究中，我們對於 HTML 網頁表格轉換成 VoiceXML格式，提出方法來分析並實作出 VTG系統，讓網站設計者只要有 HTML 格式的網頁表格，就可以使用 VTG系統來得到一個語音網頁。

但是，HTML 網頁，並不是只有表格而已，表格只是其中一個比較複雜的標籤，除了表格之外，HTML 語言還有許多標籤，例如表單資訊內容的呈現就是另一個難題。如果所有的網頁內容都能夠成功的轉換成 VoiceXML格式來呈現，那麼未來的語音網頁會更加的多姿多彩，但是，這並非一件容易的事情，因為當我們在“看”一個網頁，與“聽”一個網頁時的感覺差很多，如何讓本來需要用“看”的資訊變成用“聽”的就可以了解，是一項很大的挑戰。

目前國內正在推動無障礙網頁的標準【2】，並且規定政府單位的網頁慢慢的都要改成符合無障礙網頁標準的版本，如此看來，各界對於無障礙網頁的必

行趨勢與重視程度是有目共睹的。但是無障礙的網頁，仍是視覺所見的網頁，未來可以利用 VTG系統讓這些視覺的網頁，產生「語音網頁」的版本，那麼對於視覺障礙者來說，就不用一定得消極的靠螢幕閱讀機來閱讀視覺網頁，而是可以自己積極的打電話到某個語音網站，靠自己與語音網站互動來獲取資訊，享受自己徜徉於網際網路豐富資訊中的快感。

6. 參考文獻

- 【1】 黃英旗，以語音呈現模式導讀網頁文件之研究，國立台灣師範大學資訊教育系碩士論文，中華民國九十一年六月。
- 【2】 國內無障礙網頁標準，<http://enable.nat.gov.tw/>
- 【3】 C. M. Huang, M.Y. Jang, " Surfing the Web using a telephone set ", *IEEE Euromicro Conference, 2000* . Page(s): 126 -133 vol.2
- 【4】 C. Sharma, J. Kunins; "VoiceXML: Professional Developer's Guide with CDROM", December 7, 2001
- 【5】 Evaluation and Repair Tools Working Group, <http://www.w3.org/WAI/ER/>
- 【6】 Extensible Markup Language (XML) 1.0 (Second Edition), <http://www.w3.org/TR/REC-xml> , October 6, 2000.
- 【7】 J. A. Larson, "VoiceXML and the W3C speech interface framework", *IEEE Multimedia*, , Oct. - Dec. 2003 , Page(s): 91 -93
- 【8】 J. Kleindienst, L Seredi, P. Kapanen, J. Bergman, " CATCH-2004 multi-modal browser overview description with usability analysis ", *IEEE Fourth International Conference on Multimodal Interfaces, 2002*. Page(s): 442 -447
- 【9】 M. Mittendorfer, G. Niklfeld, W. Winiwarter, " Making the VoiceWeb smarter - integrating intelligent component technologies and VoiceXML ", *IEEE Web Information Systems Engineering, 2002* , Page(s): 126 -131 vol.2
- 【10】 S. H. Maes, " A VoiceXML framework for reusable dialog components ", *IEEE Applications and the Internet, 2002* . Page(s): 28 -30
- 【11】 Speech Recognition Grammar Specification Version 1.0, <http://www.w3.org/TR/2002/CR-speech-grammar-20020626/> , June 26, 2002.
- 【12】 Tablin: an HTML Table linearizer, <http://www.w3.org/WAI/References/Tablin/>
- 【13】 Z. Shao, R. Capra, M.A.Pérez-Quiñones, "Transcoding HTML to VoiceXML Using

Annotation”, *IEEE International Conference on Tools with Artificial Intelligence, 2003*.

- 【 14】 VoiceXML Forum, <http://www.voicexml.org/>
- 【 15】 Voice Extensible Markup Language (VoiceXML) Version 1.0, <http://www.w3.org/TR/2000/NOTE-voicexml-20000505/> , May 5, 2000.
- 【 16】 Voice Extensible Markup Language (VoiceXML) Version 2.0, <http://www.w3.org/TR/voicexml20/> , March 16, 2004.
- 【 17】 Voice Extensible Markup Language (VoiceXML) 2.1, W3C Working Draft, <http://www.w3.org/TR/voicexml21/>, March 23 , 2004.
- 【 18】 Voice Browser Working Group, <http://www.w3.org/Voice/Group/>
- 【 19】 WAI HTML Table Linearizer Entry Form, <http://www.w3.org/WAI/References/Tablin/form>
- 【 20】 Web Accessibility Initiative (WAI), <http://www.w3.org/WAI/>