第二章　相關文獻探討

本章主要是相關文獻的整理與探討。第一節探討多樣化模式存取網站的服務機制，第二節介紹 XML 的語音技術：VoiceXML，第三節舉出目前以語音瀏覽網頁的系統實例，並探討其優缺點。

2.1 多樣化模式存取網站的服務機制

一般來說，我們講到要上網存取網站的資訊或服務，大都指的是使用一般電腦，使用一般的視覺瀏覽器例如 Internet Explorer 或 Netscape 等。但是，這樣並不能滿足所有的使用者，如果所有的網路服務也可以使用語音來存取，將有更多的人受惠。於是 W3C 為語音標記語言制訂了多樣化模式(Multi-modal)存取網站的服務機制標準【11】。

未來要存取網站的服務，除了現在的視覺瀏覽器外，更可以使用語音合成、語音辨識的技術在語音瀏覽器上存取網頁。而且存取網路服務的裝置除了使用現在的一般個人電腦以外，更可以使用電話按鍵、公共場所的觸控式螢幕電子服務台、平版電腦、PDA 等嵌入式系統、行動電話等行動裝置等等，至於輸出輸入的方式，除了鍵盤以外，還可以使用語音、滑鼠、觸控螢幕、觸控筆、影音顯示等各種不同的方式。把語音技術整合到目前一些電子裝置的使用者介面上，將可以提高其使用的方便與效率，使得網路服務顯的更為人性化【8】。

在多樣化模式存取網站的服務機制中，與日常生活最有關係的，就是使用電話來存取網路資訊了。人們打電話到語音瀏覽器，接通「電話語音入口網頁」(TelePortal) 【3】以進行接下來的 VoiceXML 對話事件，進而獲得所需資訊或服務，在與語音瀏覽器互動的同時，使用者的手和眼還可以空下來做別的事

情，所以十分的方便。這裡的「電話語音入口網頁」在概念上，就像在 HTML 上的入口網站、首頁一樣，都是當我們要瀏覽網頁時，要先連到一個網站的首頁，然後才能開始在這個頁面上選擇要進行的項目。

2.2 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)，所以仍需去精細的控制屬於應用程式底層的語音辨識系統。

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」【16】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 VoiceXML1.0 標準【17】。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【20】繼續努力，制訂了目前的 VoiceXML 2.0【18】，以及目前仍是處於草案(Working Draft)階段的 VoiceXML 2.1【19】。

在 XML 推出的兩年後，用來連接網路與電話的 VoiceXML 才誕生，在這大約三年的時間裡，VoiceXML 現在幾乎已是全球支援最廣的語音標準，這都要歸功於 XML 開放標準架構的功勞。目前全球已經有一些大企業採用 VoiceXML 來啟動數千台電話系統，這些系統每天忙著回答各種來電詢問，使得客戶滿意度節節上升，而且由於這些電話系統不需聘僱人員來接聽，所以為

企業節省上百萬元費用。

依據美國一家具權威性的市場調查顧問公司 Allied Business Intelligence Inc.(ABI)的研究指出，在西元 2005 年之前，語音入口網的固定使用者在北美將會達到四百萬人，在歐洲將會達到六千萬人，且整個市場中，提供語音相關商務服務的企業，營收將會接近於五百億美金【4】。

2.2.1 VoiceXML 的相關技術

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。這兩項技術，即是 VoiceXML 所需的關鍵技術【9】，以下將分別敘述之。

1. 語音辨識(Speech Recognition)：

「語音辨識」常會與「聲音辨識」(voice recognition)混淆，聲音辨識是指電腦能夠辨識特定的聲音或者特定說話者的聲音，所以常被用在保全機制(security)或者安全認證(authentication)相關方面的應用上。而語音辨識是則指電腦有接收口語陳述的命令或字詞並加以處理的能力，這種會自動化辨識語音的能力又稱為自動語音辨識(Automatic speech recognition , ASR)。當電腦接收得到人們講出來的語音時，會將接收到的聲音數位化並作切字或切詞的處理，然後和詞彙資料庫的內容加以比對，找出相似度最高的字或詞，簡而言之，語音辨識就是「語音轉文字」(Speech→Text)。

在過去的三十年來，語音辨識經過了很多的研究與發展，辨識率

已經大幅提昇，加上電腦中央處理器的快速升級與價格下降、網路的流行與 VoiceXML 的崛起，讓很多企業組織或者一般大眾對於語音辨識相關的應用都有相當的興趣，帶給語音辨識技術很好的發展環境。目前語音辨識技術有兩個類型：語音依賴型(speaker-dependent)與非語音依賴型(speaker-independent)。

語音依賴型的語音辨識系統需要以使用者的聲音來訓練，目標是讓語音辨識系統可以辨識出這個使用者在說什麼。在語音辨識系統的訓練過程中，會建立該位使用者的聲紋資料庫，當這個資料庫的內容越完備，之後此語音辨識系統的辨識度也就越高，但是如果今天來了另外一個使用者，除非語音辨識系統要再重新訓練並建立聲紋資料庫，否則系統將無法正確辨識新使用者的聲音。

非語音依賴型的語音辨識系統則是不需要經過訓練，就可以辨識每個人的聲音，但是這種類型的辨識系統需要定義文法結構，符合文法內容的語句，才會被語音辨識系統辨識出來，並不是任何人隨便說任何一句話，都可以被語音辨識系統辨識出來的。而 VoiceXML 語音系統中所使用者的語音辨識技術，就是屬於非語音依賴型的語音辨識系統，因為我們無法對每個會打電話進語音系統的人做聲紋的訓練與記錄，所以就使用文法敘述，讓每個人只要講到關鍵字詞，就能夠被辨識出來。

2. 語音合成(Speech Synthesis)

語音合成是指電腦將接收到的文字內容，以語音的方式播出讓使用者可以聽到。當電腦接收得到一串文字時，會分析這些文字，在語音合成系統中尋找符合的聲音單元檔案，然後將這些聲音單元組合，

成為播放出來的語音。簡而言之，語音合成技術就是「文字轉語音」(Text→ Speech)，也就是一般所說的 TTS(Text-to-Speech)技術。

在 VoiceXML 裡，TTS 是一項很重要的技術，因為配合程式的運作，有一些要播放給使用者聽到的訊息都是程式運行中才會產生，所以無法將這些訊息以預錄的方式錄成錄音檔播出，必須使用 TTS 技術，將這些語音內容即時產生即時播出。

目前語音辨識和語音合成這兩項技術已經相當成熟, TTS 對於大量的文字資料以語音輸出有相當大的助益, 而 ASR 對於特定詞彙的辨認相當重要。與 ASR 息息相關互相配合的，就是語音文法(speech grammar)了，語音文法定義了哪些詞或單字在被使用者說了之後需要被 ASR 引擎辨識出來，而 W3C 對語音文法亦訂定了標準【13】。

2.2.2 VoiceXML 標準

VoiceXML 文件像 XML 一般擁有結構化的樹狀結構資料資料，在樹狀結構中，每個節點我們稱之為一個元素(Element)。在 W3C 制訂的 VoiceXML 的規格中所定義的標籤可分為以下 10 種：【4】

1.根元素(Root Elements)：

這類元素是建構一個 VoiceXML 樹狀結構最基本的的根部元素，是每個 VoiceXML 文件必備的。根元素包含的標籤有：<meta>, <vxml>。

2.對話定義元素(Dialog Definition Elements)：

這類元素是 VoiceXML 中用來定義一段對話的元素。對話定義元素包含的標籤有：<form>, <menu>。

3.表單項目元素(Form Item Elements)：

就像 HTML 文件有表單物件一樣，VoiceXML 中也有表單項目元素，這些元素是用在對話定義元素裡<form>標籤的結構之下，負責的動作是收集來電者的各項資訊，或者轉接來電者到另一個分機號碼。表單項目元素包含的標籤有：<block>,<field>,<initial>,<object>, <record>, <subdialog>,<transfer>。

4.文法元素(Grammar Elements)：

這類元素定義了在這個 VoiceXML 中的語音文法，也就是規定來電者可以說什麼字詞，以及被期待說什麼關鍵字詞。文法元素包含的標籤有：<choice>,<example>,<grammar>,<item>, <link>, <option>。

5.事件管理元素(Event Management Elements)：

這類元素負責啟發或者處理各種在 VoiceXML 對話進行中會發生的事件。事件管理元素包含的標籤有：<catch>, <error>, <help>, <noinput>, <nomatch>, <throw>。

6.轉變元素(Transition Elements)：

當一段對話進行到某個時候需要轉換到另一個對話，或者要從一個 VoiceXML 文件轉換到另一個 VoiceXML 文件時，就需要這類轉變元素了。轉變元素包含的標籤有：<disconnect>, <exit>, <goto>, <param>, <return>, <submit>

7.範疇元素(Field Item Elements)：

這些元素是用在表單項目元素的結構之下，負責收集來電者的選擇，以及對於這些選擇做出反應。範疇元素包含的標籤有：<audio>, <choice>,

<enumerate>, <filled>, <prompt>, <reprompt>。

8.邏輯元素(Logic Elements)：

這類元素用來處理 VoiceXML 事件中的一些邏輯運算。邏輯元素包含的標籤有：<assign>, <break>, <clear>, <data>, <foreach>, <if>/<elseif>/<else>, <script>, <value>,<var>。

9.TTS 元素(Text to Speech Elements)：

這類元素可以控制 VoiceXML 平台中，TTS 系統的合成語音。TTS 元素包含的標籤有：<emphasis>, <p>, <paragraph>, <phoneme>, <prosody>, <s>, <say-as>, <sentence>, <voice>。

10.混和元素(Miscellaneous Elements)：

這類元素提供了 VoiceXML 一些其他的附加功能。混和元素包含的標籤有：<property>, <log>。

以上 10 類元素，形成了 VoiceXML 文件的架構，使 VoiceXML 可以成功的將網路應用的彈性與品質擴張到電話上。表 2-1 將這 10 類元素做一整理與歸類。

類別	標籤
根元素 (Root Elements)	<meta>, <vxml>
對話定義元素 (Dialog Definition Elements)	<form>, <menu>
表單項目元素 (Form Item Elements)	<block>,<field>,<initial>,<object>, <record>, <subdialog>,<transfer>
文法元素 (Grammar Elements)	<choice>,<example>,<grammar>,<item>, <link>, <option>

事件管理元素 (Event Management Elements)	<catch>, <error>, <help>, <noinput>, <nomatch>, <throw>
轉變元素 (Transition Elements)	<disconnect>, <exit>, <goto>, <param>, <return>, <submit>
範疇元素 (Field Item Elements)	<audio>, <choice>, <enumerate>, <filled>, <prompt>, <reprompt>
邏輯元素 (Logic Elements)	<assign>, <break>, <clear>, <data>, <foreach>, <if>/<elseif>/<else>, <script>, <value>,<var>
TTS 元素 (Text to Speech Elements)	<emphasis>, <p>, <paragraph>, <phoneme>, <prosody>, <s>, <say-as>, <sentence>, <voice>
混和元素 (Miscellaneous Elements)	<property>, <log>

表 2-1　　VoiceXML 標籤歸類

2.2.3 VoiceXML 的執行環境

一個完整的 VoiceXML 執行環境包含了電話使用者、語音伺服器、網頁伺服器、以及儲存資料內容的相關軟硬體。語音伺服器中包含語音瀏覽器、自動語音辨識模組(ASR)、文字轉語音模組(TTS)，還有抓取語音或按鍵音、播放語音的模組；而網頁伺服器中會有 VoiceXML 文件、定義語音文法的文件、聲音檔案等等。儲存資料內容的相關軟硬體，例如資料庫伺服器，並不會直接跟語音伺服器互動，若語音伺服器需要存取資料庫內容，需要透過網頁伺服器裡的程式內容來與資料庫溝通。

如圖 2-1 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、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】【12】

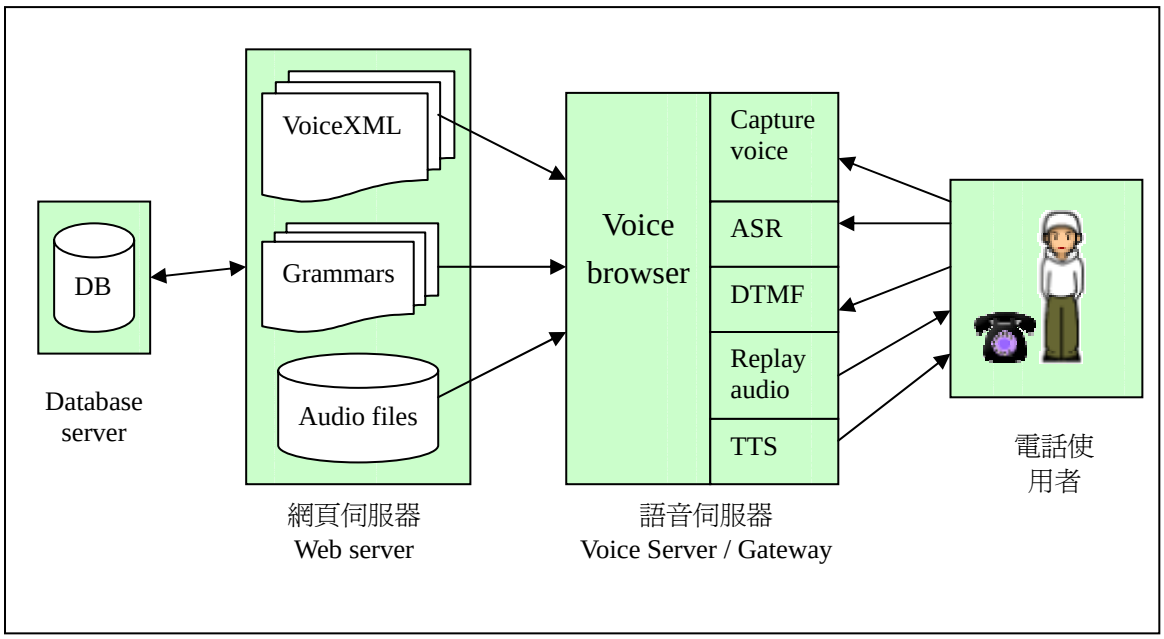

圖 2-1. VoiceXML 平台架構圖

2.2.4 動態 VoiceXML 語音應用程式

由於 VoiceXML 標記語言承襲 XML 標記語言的特性，所以 VoiceXML 的內容也有靜態與動態之分。靜態的 VoiceXML 指的是，一個 VoiceXML 文件的內容是在設計的時候就固定好的，並不是隨著使用者的存取而改變任何的標籤或屬性內容；而動態的 VoiceXML 指的是，藉由與 JSP 或 ASP 程式的結合，讓 VoiceXML 文件的內容可以在使用者使用 VoiceXML 系統的時候，動態的產生，然後以這個 VoiceXML 文件內容來與使用者互動【4】。

在 JSP 或 ASP 文件中，可以既包括 VoiceXML 元素又包括 JSP 或 ASP 程式碼，這樣形成的強力組合，可以實作以語音為基礎的應用程式，它可以存取各種資料庫或企業應用程式（例如：電子信件、帳務系統、時間和費用等），彈性以及功能都會非常強大。圖 2-2 是動態 VoiceXML 語音應用程式架構圖，網路應用程式可以內嵌於 VoiceXML 系統架構中，如圖中的 JSP 或 ASP 程式文件存在於網頁伺服器中，一旦語音伺服器對網頁伺服器發出某個請求時，網頁

伺服器就會以適當的JSP或ASP程式文件來回應，完成電話使用者要求的服務。

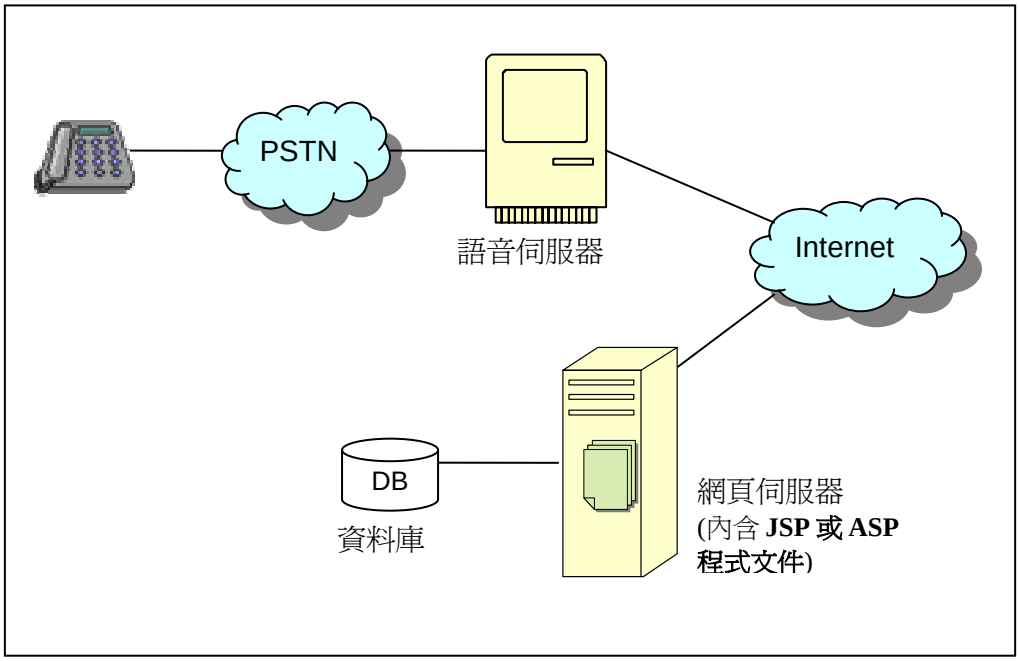

圖 2-2　　動態 VoiceXML 語音應用程式架構圖

2.3 現今網頁轉換成語音文件的實例探討

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。就軟體的角度來看，與「語音瀏覽網頁」相關的軟體系統不多，在此我們探討比較具有代表性的兩個：1.IBM 的 WebSphere Transcoding Publisher(WTP)；2.W3C 的資訊網可及性推動組織(Web Accessibility Initiative ,WAI【23】)的「Tablin【14】【22】」。以下就這兩點分別探討之。

1.WebSphere Transcoding Publisher(WTP)：

WTP 是 IBM 公司發展的「互聯網基礎架構軟件平台(WebSphere)」家族系列裡的一個成員，WebSphere 這個系列產品的功能是提供一個完整的、可延展的、彈性化的平台，幫助企業運轉的應用系統拓展到網際網路上。在 WTP 裡，有一個「HTML-to-VoiceXML Transcoder」模組可以將一個 HTML 網頁轉換成 VoiceXML 網頁，但是 HTML 網頁的內容繁雜多變化，要將 HTML 網頁的內

容精確的轉換成 VoiceXML 的格式並不容易，在目前的技術來說也是尚未成熟，故 WTP 也僅能就比較單純的網頁來做轉換，例如單純的文字內容或者超連結，且 WTP 會抓取 HTML 網頁裡的 header 標籤例如<H1><H2>等，來作為轉換成 VoiceXML 格式內容的判斷，但是並不是每個網頁都會有 header 標籤，如果遇到沒有 header 標籤的網頁，那麼轉換的效果很可能就會不如預期。所以對於 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，WTP 目前暫時並無解決的策略。

2.Tablin：

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，所以就還是依照一般的順序，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，這樣的方式就會顯的很沒效率，因為並不是每個人都想聽完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】【15】，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，所以本研究想朝著這個目標邁進，希望在考量與表格內容聆聽者互動的前提下，探討如何將 HTML 網頁的表格內容轉換成 VoiceXML 格式，讓想要知道表格資訊的人，只要打一通電話到語音平台，藉由語音與語音平台對話，就可以得到想要查詢的表格資訊。