

## 第三章 網頁表格資訊自動對話模式設計

本章我們將提出網頁表格資訊自動對話模式設計的整體概念。在第一節中介紹了「VoiceXML 電話語音網站製作中心」(VoiceXML Portal Builder, 簡稱 VXPB)系統的運作與內部設計；第二節將內嵌於 VXPB 系統中的「語音表格產生器」(Voice Table Generator, 簡稱 VTG)系統作詳細的解說，探討網頁型表格資訊對話模式；第三節將介紹有別於網頁型表格的另類表格，也就是資料型表格，並以實作語音行事曆系統的方法作為例子，說明以 XML 文件配合 VoiceXML 語音技術，就可以將資料型表格的功能發揮出來。

### 3.1 電話語音網站製作中心

在上一章的相關文獻探討之中，現今語音瀏覽網頁的系統大都尚未成熟，功能也都有所限制，但是人們對於使用電話語音來瀏覽網頁又是越來越需要，除了視覺障礙者可以藉由電話取代眼睛來瀏覽網頁之外，一般人亦可在遠離辦公室或者住家時，使用電話來瀏覽網頁，查得所需要的資訊。因此，本研究決定開發一個製作語音網站的系統，稱為「VoiceXML 電話語音網站製作中心」(VoiceXML Portal Builder, VXPB)，其功能是讓一個網頁設計者就算不會 VoiceXML 語音技術，也可以輕鬆的建立一個語音網站。

#### 3.1.1 VXPB 運作概念

VXPB 稱為「VoiceXML 電話語音網站製作中心」，顧名思義就是用來製作一個語音網站的工具，網頁設計者需要自行規劃這個網站的樹狀架構，與每個樹狀節點的內容，當編輯好一個網站的時候，VXPB 系統會將此語音網站所

有的語音網頁檔打包輸出讓網站設計者下載，當網站設計者得到這些語音網頁檔之後，只要將這些網頁置於語音平台上，就可以讓電話使用者打電話進到這個語音網站了。圖 3-1 說明了 VXPB 系統的運作概念。



圖 3-1 VXPB 的運作概念

在一個語音網站的樹狀結構中，會出現兩種節點，分別是「對話目錄」與「對話節點」，在對話目錄下，還會有一到多個對話目錄或者對話節點，但是對話節點，就已經是樹狀結構的末端了，不會再有任何的節點成為對話節點的子節點了。例如圖 3-2 是一個編輯中的網路書店小範例，其中「BookStore」與「查詢書籍排行榜」這兩個節點由於下一層仍有節點的存在，所以被歸類為「對話目錄」，而其他的節點，由於現在沒有下一層節點的存在，所以目前被歸類為「對話節點」。



圖 3-2 樹狀節點範例說明

但是無論是「對話目錄」或「對話節點」，在這個語音網站裡，都是一個語音網頁，其中對話目錄類型的語音網頁是由 VXPB 系統本身產生，而對話節點類型的語音網頁，則是由我們另外開發的一個模組，稱為「語音表格產生器」(Voice Table Generator, VTG) 來產生，關於 VTG，在本章的第二節會有詳細敘述。

### 3.1.2 對話目錄類型語音網頁的必備要素

一個對話目錄類型的語音網頁需包含很多資訊，當 VXPB 系統收集到網站設計者輸入的資訊時，將會把這些資訊以 XML 文件來記錄之，在此我們將此 XML 文件稱為「目錄頁面 XML 文件」，在 VXPB 中，每一個目錄頁面 XML 文件的命名方式都是以「Level\_Count.xml」這樣的形式來命名，其中 Level 表示這個對話目錄位於樹狀結構的第幾層，Count 表示這個對話目錄位於樹狀結構所在那一層的第幾個。表 3-1 說明了目錄頁面 XML 文件的標籤及屬性意義。

標籤名稱	屬性名稱	說明
page		<page>標籤為這個 XML 文件的根元素。
	Level	Level 屬性表示這個語音網頁在整個語音網站樹狀結構中的哪一層(從 0 開始算)
	Count	Count 屬性表示這個語音網頁在整個語音網站樹狀結構中所在那一層的第幾個(從 0 開始算)
	name	name 屬性表示這個目錄型語音網頁在樹狀結構中的名稱
welcome		<welcome>標籤記錄的資訊為電話語音網頁要說給來電者聽的歡迎訊息。
prompt		<prompt>標籤記錄的資訊為電話語音網頁要說給來電者的提示訊息，提示來電者有哪些選項可以選擇。
noinput		<noinput>標籤記錄的資訊為當電話語音平台沒聽到來電者說的話時，要對來電者說出什麼訊息。
nomatch		<nomatch>標籤記錄的資訊為當電話語音平台沒聽清楚來電者說的話時，要對來電者說出什麼訊息。
allchoice		<allchoice>標籤記錄了這個語音網頁有哪些選項可以給來電者選擇
	num	num 屬性記錄來電者總共有幾個選項可以選，也就是記錄在<allchoice>標籤下，會有幾個<choice>標籤。
choice		<choice>標籤記錄了某個選項的相關資訊
	Level	Level 屬性表示使用者選了這個選項後，接下來會被傳送到樹狀結構哪一層的語音網頁
	Count	Count 屬性表示使用者選了這個選項後，接下來會被傳送到樹狀結構某一層的第幾個的語音網頁
grammar		<grammar>標籤記錄這個語音網頁的文法內容。
	num	num 屬性表示在這個語音網頁裡，每個選項的辨識詞可以有幾個，也就是記錄在<grammar>標籤下，會有幾個<recognize>標籤。
recognize		<recognize>標籤記錄在文法敘述裡每個選項可容許的辨識詞內容。在樹狀結構中
goto		<goto>標籤記錄來電者選了某選項後，會被傳送到語音網頁位置。
	leaf	leaf 屬性表示來電者選了某選項後，會被傳送到位置位於樹狀結構中是否為「對話節點」。
	name	name 屬性表示來電者選了某選項後，會被傳送到語音網頁名稱
	file	file 屬性表示來電者選了某選項後，會被傳送到語音網頁檔名。

表 3-1 目錄頁面 XML 文件的標籤及屬性意義

「目錄頁面 XML 文件」的標籤結構則如圖 3-3 所示，<page>為第 0 層根元素，第一層的<welcome>, <prompt>, <noinput>, <nomatch>標籤為記錄一個語音網頁的提示訊息，同屬第一層的<allchoice>標籤則是記錄了所有的選項，第二層的<choice>標籤則是<allchoice>標籤的子標籤，而<choice>標籤的子標籤則是位於第三層標籤的<grammar>與<goto>，其中記錄文法資訊的<grammar>標籤的子標籤為第四層的<recognize>，而記錄選了選項後被傳送到語音網頁位置資訊的<goto>標籤，在第四層的子標籤有<name>與<file>。

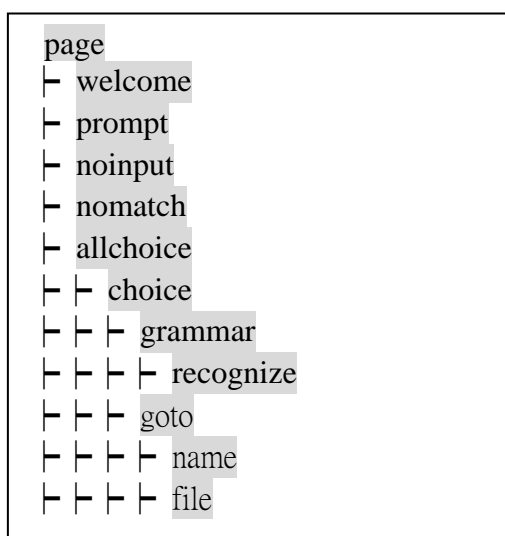


圖 3-3 「目錄頁面 XML 文件」的標籤結構

關於「目錄頁面 XML 文件」的 Schema 請參見本論文附錄一。

圖 3-4 是一個「目錄頁面 XML 文件」範例，也是圖 3-2 樹狀節點例子中，「查詢書籍排行榜」這一個對話目錄的目錄頁面 XML 文件。從這個文件裡可以看出這個語音網頁的所有資訊，例如這個語音網頁有兩個選項：查詢文學類，與查詢非文學類，這兩個選項各有兩種辨識詞，連到的語音網頁分別是 2\_0.vxml 與 2\_1.vxml，且連到的這兩個語音網頁檔在樹狀結構中，都是屬於對話節點等等。

```

<?xml version="1.0" encoding="UTF-8" ?>
<page Level="1" Count="4" name="查詢書籍排行榜">
  <welcome>歡迎查詢書籍排行榜</welcome>
  <prompt>請說出您要查詢文學類、或查詢非文學類</prompt>
  <noinput>對不起，我沒聽到你說什麼。</noinput>
  <nomatch>對不起，我沒聽清楚。</nomatch>
  <allchoice num="2">
    <choice Level="2" Count="0">
      <grammar num="2">
        <recognize>查詢文學類</recognize>
        <recognize>文學類</recognize>
      </grammar>
      <goto leaf="true">
        <name>查詢文學類</name>
        <file>2_0.vxml</file>
      </goto>
    </choice>
    <choice Level="2" Count="1">
      <grammar num="2">
        <recognize>查詢非文學類</recognize>
        <recognize>非文學類</recognize>
      </grammar>
      <goto leaf="true">
        <name>查詢非文學類</name>
        <file>2_1.vxml</file>
      </goto>
    </choice>
  </allchoice>
</page>

```

圖 3-4 「目錄頁面 XML 文件」範例

### 3.1.3 語音網站的內部結構

在 VXPB 系統裡，網頁設計者可以自行決定所設計的語音網站的樹狀結構，對於這個設計中的語音網站，VXPB 必須想辦法將網頁設計者所設計的樹狀結構記錄下來，因此，我們使用一個 XML 文件來記錄，我們將此 XML 文件稱為「樹狀網站 XML 文件」，在 VXPB 中，每一個樹狀網站 XML 文件的命名都是「tree.xml」，每一個語音網站專案中只會有一棵表示語音網站架構的「結構樹」，所以也只會有一個 tree.xml 檔案。表 3-2 說明了樹狀網站 XML 文件

的標籤及屬性意義。

標籤名稱	屬性名稱	說明
root		<page>標籤為這個 XML 文件的根元素。
	Level	Level 屬性記錄根元素在樹狀結構裡的層級。
	Count	Count 屬性記錄根元素在樹狀結構裡所在那一層的第幾個。
	leaf	leaf 屬性記錄根元素這個節點是否為對話節點。
	name	name 屬性記錄根元素代表的語音網頁的名稱。
	file	file 屬性記錄根元素代表的語音網頁的檔名。
node		在樹狀網站 XML 文件裡，除了根元素之外的每個節點都用<node>標籤來記錄資訊。
	Level	Level 屬性記錄這個節點在樹狀結構裡的層級。
	Count	Count 屬性記錄這個節點在樹狀結構裡所在那一層的第幾個。
	leaf	leaf 屬性記錄這個節點是否為對話節點。
	name	name 屬性記錄這個節點代表的語音網頁的名稱。
	file	file 屬性記錄這個節點代表的語音網頁的檔名。

表 3-2 樹狀網站 XML 文件的標籤及屬性意義

「目錄頁面 XML 文件」的標籤結構則如圖 3-3 所示，<root>為第 0 層根元素，之後的第一層至第 N 層都是以一個<node>標籤來記錄每一個節點，任何一個<node>標籤的子標籤都可以再是一個<node>標籤，這樣一層一層的記錄下去，樹狀結構的內容就會越來越豐富，此樹狀結構所代表的語音網站資料量也就會越來越龐大了。

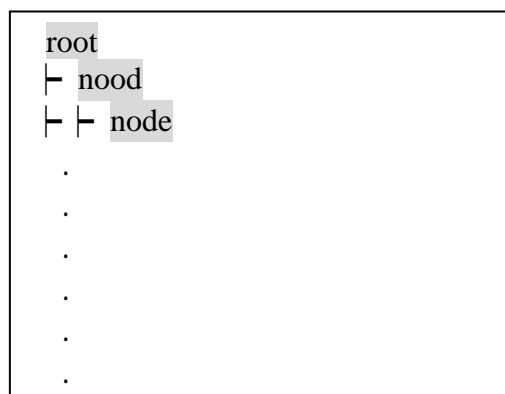


圖 3-5 「樹狀網站 XML 文件」的標籤結構

關於「樹狀網站 XML 文件」的 Schema 請參見本論文附錄二。

圖 3-6 是一個「樹狀網站 XML 文件」範例，也就是圖 3-2 樹狀節點例子裡的那個樹狀結構樣子。從這個文件裡可以看出這個語音網頁樹狀結構資訊，例如同一層每個節點之間或父節點與子節點不同層之間的關係，與某個節點是位於整棵結構樹的哪個位置的資訊等等。

```
<?xml version="1.0" encoding="UTF-8" ?>
<root Level="0" Count="0" leaf="false" name="BookStore" file="0_0.xml">
  <node Level="1" Count="0" leaf="false" name="查詢新書" file="1_0.xml" />
  <node Level="1" Count="1" leaf="false" name="查詢暢銷書" file="1_1.xml" />
  <node Level="1" Count="2" leaf="false" name="查詢特價書" file="1_2.xml" />
  <node Level="1" Count="3" leaf="false" name="查詢推薦書" file="1_3.xml" />
  <node Level="1" Count="4" leaf="false" name="查詢書籍排行榜" file="1_4.xml">
    <node Level="2" Count="0" leaf="true" name="查詢文學類" file="2_0.vxml" />
    <node Level="2" Count="1" leaf="true" name="查詢非文學類" file="2_1.vxml" />
  </node>
</root>
```

圖 3-6 「樹狀網站 XML 文件」範例

### 3.1.4 VXPB 的運作方式

在 VXPB 裡，「目錄頁面 XML 文件」與「樹狀網站 XML 文件」與整個系統之間的關係是很密切的，在圖 3-7 的 VXPB 系統流程圖中，詳細的記錄了所有會存取「目錄頁面 XML 文件」或「樹狀網站 XML 文件」的情形。

當網站設計者要新增一個語音網站的專案時，需先輸入專案名稱，然後填入語音網站首頁所需要的所有資訊後，系統便會產生出一個「樹狀網站 XML 文件」(tree.xml)與一個「目錄頁面 XML 文件」(0\_0.xml)，tree.xml 在之後會一直記錄這個語音網站的結構，而 0\_0.xml 記錄的是根節點的目錄頁面資訊。

接下來，網站設計者就可以一直不停的編輯對話目錄或者編輯對話節點，



當每次編輯對話目錄時，系統會生成這個對話目錄的目錄頁面 XML 文件，並且修改此語音網站的 tree.xml 文件，藉以更新整個語音網站的樹狀架構；而當每次編輯對話節點時，系統會更新此對話節點的父節點之目錄頁面 XML 文件裡的資訊，並且也會修改此語音網站的 tree.xml 文件。

當使用者編輯完語音網頁的所有對話目錄與對話節點時，可以選擇結束編輯，此時 VXPB 系統會檢視 tree.xml 裡有哪些節點，並且把這些節點的每一個目錄頁面 XML 文件都找出來，把這些目錄頁面 XML 文件送進系統的轉換包裝模組，作轉換成 VoiceXML 檔案的動作，然後網站設計這就可以得到這個語音網站的所有語音網頁檔案了。

當網站設計者從 VXPB 得到語音網站的所有檔案後，將這些 VoiceXML 檔案放到語音平台中，啟動語音平台，那麼電話使用者就可以打電話到這個語音網站查詢所需要的資訊了。

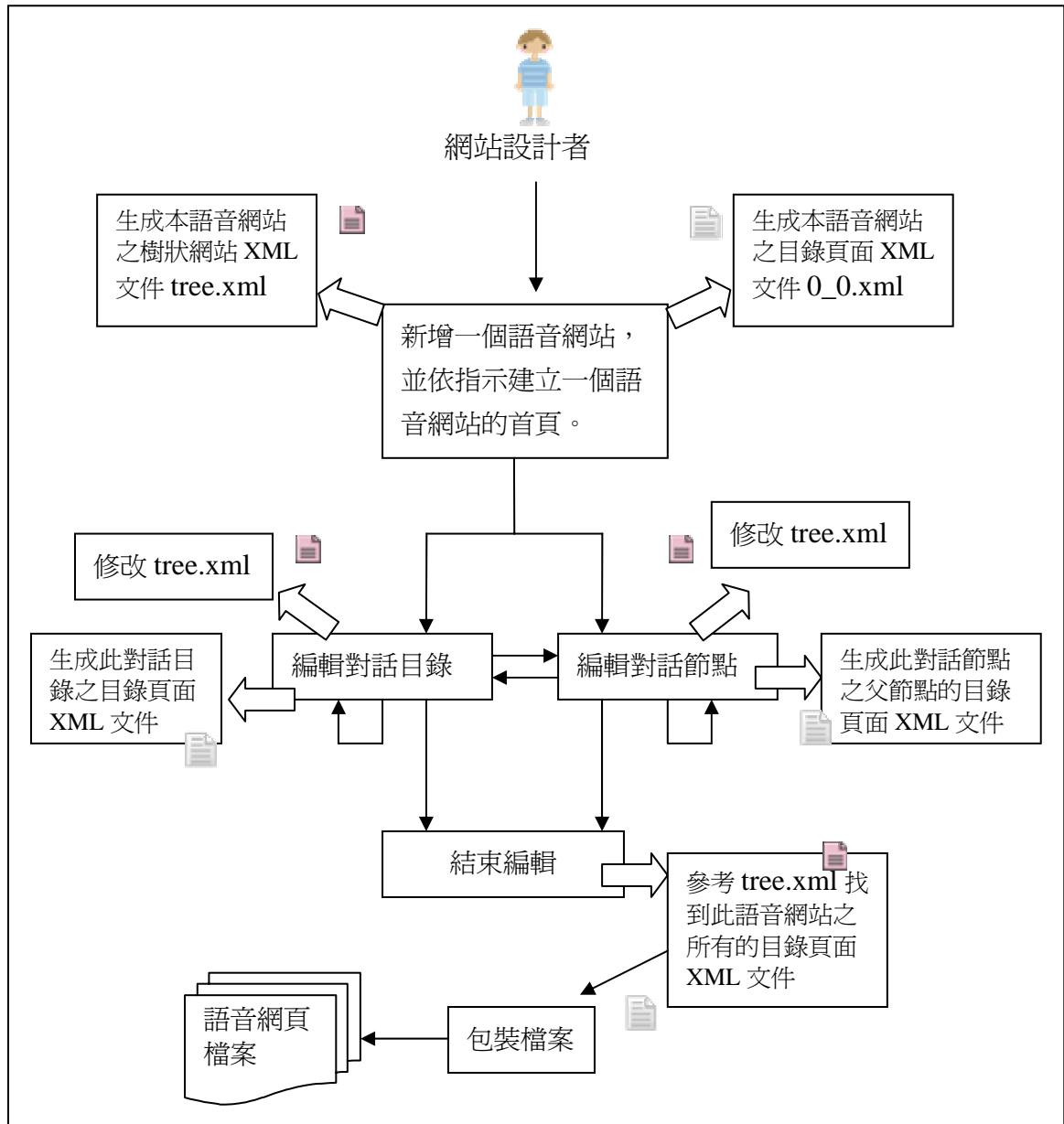


圖 3-7 VXPB 系統流程圖

### 3.2 網頁型表格資訊對話模式

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，實在不是一件容易的事情。所以我們在這裡從表格<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。

### 3.2.1 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做分析與歸類，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)

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。圖 3-8 是一個以行為主型的網頁表格，此表格的標題有：項次、品名、契約單價等三個標題，其餘則是標題對應下的資料欄位。

第九組護目鏡		
SUBFORM (項次)	ITEM(品名)	契約單價
1	15 吋陰極映像管顯示器護目鏡	953
2	15 吋液晶顯示器護目鏡	1,025
3	17 吋陰極映像管顯示器護目鏡	1,056
4	17 吋液晶顯示器護目鏡	1,261
5	19 吋液晶顯示器護目鏡	1,599

圖 3-8 以行為主型網頁表格範例

## 2. 以列為主型(Row-wise)

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。圖 3-9 是一個以列為主型的網頁表格，此表格的標題有：本系地址、聯絡電話、聯絡傳真、電子佈告欄等四個標題，其餘則是標題對應下的資料欄位。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。

聯絡資料	
本系地址	國立臺灣師範大學資訊教育學系 台北市和平東路一段162號
聯絡電話	(02)23622841、23410420、23944288
聯絡傳真	(02)23512772
電子佈告欄	bbs.ice.ntnu.edu.tw

圖 3-9 以列為主型網頁表格範例

## 3. 行列型(Column & Row)

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。圖 3-10 是一個行列型的網頁表格，此表格的行標題有天氣預測、氣溫、降雨機率，列標題有台北市、高雄市、基隆北海岸、台北地區、桃園地區等等。行列型的表格是一種在網路上最常見的一種表格。

地區名稱	天氣預測	氣溫	降雨機率
台北市	 晴時多雲	26~34°C	0%
高雄市	 晴時多雲	27~34°C	0%
基隆北海岸	 晴時多雲	25~33°C	0%
台北地區	 晴時多雲	25~33°C	0%
桃園地區	 晴時多雲	25~33°C	0%

圖 3-10 行列型網頁表格範例

#### 4. 跨行型(Column-span)

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，而且某一行有跨行的現象，故名為跨行型表格。跨行型表格其實與第一類型表格，也就是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。圖 3-11 是一個跨行型的網頁表格，此表格的標題有兩列，第一列為學生、作業一、作業二，這三個標題的每一個都跨是跨了兩行的標題；而第二列為：座號、姓名、程式、文件、程式、文件。其餘則是標題對應下的資料欄位。

學生		作業一		作業二	
座號	姓名	程式	文件	程式	文件
1	范小敏	87	89	67	97
2	林文靜	53	44	54	63
3	成恩恩	74	84	84	54
4	黃耀輝	67	86	86	63
5	高英偉	34	46	78	74

圖 3-11 跨行型網頁表格範例

## 5. 跨列型(Row-span)

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。跨列型表格其實與第二類型表格，也就是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。圖 3-12 是一個跨列型的網頁表格，此表格的標題有兩行，第一行為台大、師大，這兩個標題的每一個都跨是跨了兩列的標題；而第二行為：電話、住址、電話、住址。其餘則是標題對應下的資料欄位。

台大	電話	(02)23630231
	地址	台北市106羅斯福路四段一號
師大	電話	(02)23625105
	地址	台北市和平東路一段162號

圖 3-12 跨列型網頁表格範例

## 6. 跨行跨列型(Column-span & Row-span)

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，某一行有跨行的現象，故名為跨行跨列型表格。跨行跨列型表格其實與第三類型表格，也就是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。圖 3-13 是一個跨行跨列型的網頁表格，此表格的行標題有兩列，第一列為網路聯絡方式、電話，這兩個標題的每一個都跨是跨了兩行的標題，而第二列為：電子信箱、系所首頁、外線、校內分機；而列標題有兩行，第一行為教育學院、文學院，分別跨了四列、兩列，第二行為：資訊教育系、教育系特殊教育系、

圖書資訊學研究所、歷史學系、英語學系。其餘則是標題對應下的資料欄位。

		網路聯絡方式		電話	
		電子信箱	系所首頁	外線	校內分機
教育學院	資訊教育系	ccliu@ice.ntnu.edu.tw	http://www2.ice.ntnu.edu.tw	23410420	8355
	教育系	education@deps.ntnu.edu.tw	http://www.ed.ntnu.edu.tw/	23419613	8302
	特殊教育系	sp-edu@deps.ntnu.edu.tw	http://www.ntnu.edu.tw/spe/	23952441	8357
	圖書資訊學研究所	meiwu@cc.ntnu.edu.tw	http://www.glis.ntnu.edu.tw/	23936798	8351
文學院	歷史學系	t23028@cc.ntnu.edu.tw	http://www.his.ntnu.edu.tw/	23637881	8450
	英語學系	bartleby@cc.ntnu.edu.tw	http://www.eng.ntnu.edu.tw/	23632664	8426

圖 3-13 跨行跨列型網頁表格範例

### 3.2.2 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。為了要可以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」。表 3-3 說明了表格索引 XML 文件的標籤及屬性意義。

標籤名稱	屬性名稱	說明
info		<info>標籤為表格索引 XML 文件的根元素。
	model	model 屬性記錄這個表格式屬於六種類型表格的哪一種類型。
	row_num	row_num 屬性記錄這個表格總共有幾列。
	column_num	column_num 屬性記錄這個表格總共有幾行。
tr		<tr>標籤記錄表格橫排（行）內容。
	name	name 屬性記錄<tr>標籤所在的這一行的儲存格座標名稱
td		<td>標籤記錄表格直排（列）內容。
	name	name 屬性記錄<td>標籤所在的這一系列的儲存格座標名稱

表 3-3 表格索引 XML 文件的標籤及屬性意義

「表格索引 XML 文件」的標籤結構則如圖 3-14 所示，<info>為第 0 層根元素，之後的第一層是表示列的<tr>標籤，每個<tr>標籤都有一至數個表示行的子標籤<td>，每個<tr>與<td>標籤都有一個 name 屬性來記錄儲存格座標名稱，所以 VTG 系統使用這樣一個表格索引 XML 文件，可以將 HTML 表格資訊記錄的很清楚。

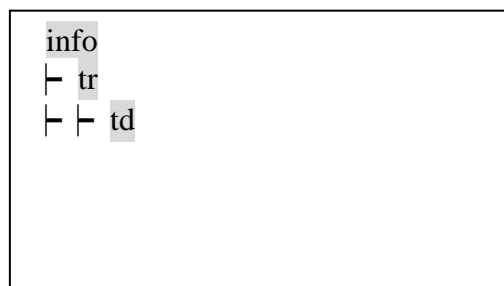


圖 3-14 「表格索引 XML 文件」的標籤結構

關於「表格索引 XML 文件」的 Schema 請參見本論文附錄三。

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 3-4 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> &lt;table&gt;   &lt;tr&gt;     &lt;td&gt;&lt;/td&gt;     &lt;td&gt;國文&lt;/td&gt;     &lt;td&gt;數學&lt;/td&gt;     &lt;td&gt;社會&lt;/td&gt;     &lt;td&gt;自然&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;王大頭&lt;/td&gt;     &lt;td&gt;80&lt;/td&gt;     &lt;td&gt;56&lt;/td&gt;     &lt;td&gt;87&lt;/td&gt;     &lt;td&gt;75&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;陳美美&lt;/td&gt;     &lt;td&gt;74&lt;/td&gt;     &lt;td&gt;85&lt;/td&gt;     &lt;td&gt;76&lt;/td&gt;     &lt;td&gt;37&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;張小猴&lt;/td&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;info model="column&amp;row" row_num="8" column_num="5"&gt;   &lt;tr name="\$r1"&gt;     &lt;td name="\$c1" /&gt;     &lt;td name="\$c2"&gt;國文&lt;/td&gt;     &lt;td name="\$c3"&gt;數學&lt;/td&gt;     &lt;td name="\$c4"&gt;社會&lt;/td&gt;     &lt;td name="\$c5"&gt;自然&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r2"&gt;     &lt;td name="\$c1"&gt;王大頭&lt;/td&gt;     &lt;td name="\$c2"&gt;80&lt;/td&gt;     &lt;td name="\$c3"&gt;56&lt;/td&gt;     &lt;td name="\$c4"&gt;87&lt;/td&gt;     &lt;td name="\$c5"&gt;75&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r3"&gt;     &lt;td name="\$c1"&gt;陳美美&lt;/td&gt;     &lt;td name="\$c2"&gt;74&lt;/td&gt;     &lt;td name="\$c3"&gt;85&lt;/td&gt;     &lt;td name="\$c4"&gt;76&lt;/td&gt;     &lt;td name="\$c5"&gt;37&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r4"&gt;     &lt;td name="\$c1"&gt;張小猴&lt;/td&gt; </pre>

<pre>&lt;td&gt;54&lt;/td&gt; &lt;td&gt;56&lt;/td&gt; &lt;td&gt;100&lt;/td&gt; &lt;td&gt;83&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</pre>	<pre>&lt;td name="\$c2"&gt;54&lt;/td&gt; &lt;td name="\$c3"&gt;56&lt;/td&gt; &lt;td name="\$c4"&gt;100&lt;/td&gt; &lt;td name="\$c5"&gt;83&lt;/td&gt; &lt;/tr&gt; &lt;/info&gt;</pre>
--	---

表 3-4 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 3-15 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。

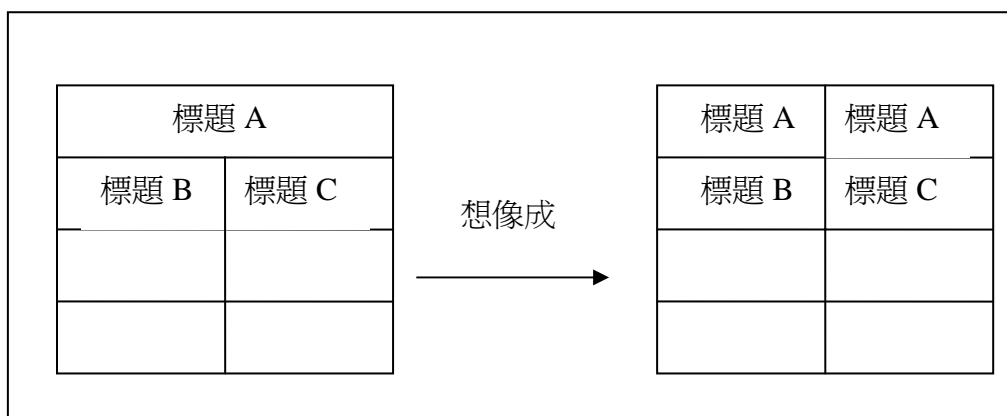


圖 3-15 跨行或跨列的表格處理方式示意圖

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。表 3-5 是第二大類的表格形成表格索引 XML 文件的對照表，本表格是一個「跨行跨列型」的網頁表格，資料內容為教育學院與文學院各兩個系的電話

資料，電話有外線電話與校內分機，從對照表可以很明顯的看出左方的 HTML 文件內容與有方的表格索引 XML 文件的對比。另外同屬於這一大類的跨行型、跨列型網頁表格轉換成 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> &lt;table&gt;   &lt;tr&gt;     &lt;td&gt;&lt;/td&gt;     &lt;td&gt;&lt;/td&gt;     &lt;td colspan="2" &gt;電話&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;&lt;/td&gt;     &lt;td&gt;&lt;/td&gt;     &lt;td&gt;外線&lt;/td&gt;     &lt;td&gt;校內分機&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td rowspan="2"&gt;教育學院&lt;/td&gt;     &lt;td&gt;資訊教育系&lt;/td&gt;     &lt;td&gt;23410420&lt;/td&gt;     &lt;td&gt;8355&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;特殊教育系&lt;/td&gt;     &lt;td&gt;23952441&lt;/td&gt;     &lt;td&gt;8357&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td rowspan="2"&gt;文學院&lt;/td&gt;     &lt;td&gt;歷史學系&lt;/td&gt;     &lt;td&gt;23637881&lt;/td&gt;     &lt;td&gt;8450&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;英語學系&lt;/td&gt;     &lt;td&gt;23632664&lt;/td&gt;     &lt;td&gt;8426&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8" ?&gt; &lt;info model="column-span&amp;row-span" row_num="6" column_num="4"&gt;   &lt;tr name="\$r1"&gt;     &lt;td name="\$c1"&gt;&lt;/td&gt;     &lt;td name="\$c2"&gt;&lt;/td&gt;     &lt;td name="\$c3"&gt;電話&lt;/td&gt;     &lt;td name="\$c4"&gt;電話&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r2"&gt;     &lt;td name="\$c1"&gt;&lt;/td&gt;     &lt;td name="\$c2"&gt;&lt;/td&gt;     &lt;td name="\$c3"&gt;外線&lt;/td&gt;     &lt;td name="\$c4"&gt;校內分機&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r3"&gt;     &lt;td name="\$c1"&gt;教育學院&lt;/td&gt;     &lt;td name="\$c2"&gt;資訊教育系&lt;/td&gt;     &lt;td name="\$c3"&gt;23410420&lt;/td&gt;     &lt;td name="\$c4"&gt;8355&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r4"&gt;     &lt;td name="\$c1"&gt;教育學院&lt;/td&gt;     &lt;td name="\$c2"&gt;特殊教育系&lt;/td&gt;     &lt;td name="\$c3"&gt;23952441&lt;/td&gt;     &lt;td name="\$c4"&gt;8357&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r5"&gt;     &lt;td name="\$c1"&gt;文學院&lt;/td&gt;     &lt;td name="\$c2"&gt;歷史學系&lt;/td&gt;     &lt;td name="\$c3"&gt;23637881&lt;/td&gt;     &lt;td name="\$c4"&gt;8450&lt;/td&gt;   &lt;/tr&gt;   &lt;tr name="\$r6"&gt;     &lt;td name="\$c1"&gt;文學院&lt;/td&gt;     &lt;td name="\$c2"&gt;英語學系&lt;/td&gt;     &lt;td name="\$c3"&gt;23632664&lt;/td&gt;     &lt;td name="\$c4"&gt;8426&lt;/td&gt;   &lt;/tr&gt; &lt;/info&gt; </pre>

表 3-5 第二大類的表格索引 XML 文件的對照表

### 3.2.3 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。表 3-6 說明了閱讀次序 XML 文件的標籤及屬性意義。

標籤名稱	屬性名稱	說明
titleseq		<titleseq>標籤為閱讀次序 XML 文件的根元素。
	model	model 屬性記錄這個表格式屬於六種類型表格的哪一種類型。
	row_num	row_num 屬性記錄這個表格總共有幾列。
	column_num	column_num 屬性記錄這個表格總共有幾行。
titles		<titles>標籤記錄所有的標題內容。
title		<title>標籤記錄單一個標題內容。
	name	name 屬性記錄這個標題是行標題或者列標題。
seq		<seq>標籤記錄此表格所有的閱讀模式。
all		<all>標籤記錄此表格若要念全部資訊的閱讀順序。
onocol		<onocol>標籤記錄此表格若要念某一欄(行)的閱讀順序。
col		<col>標籤記錄念這一欄(行)的閱讀順序。
	name	name 屬性記錄這一欄(行)的標題儲存格座標名稱。
onerow		<onerow>標籤記錄此表格若要念某一列的閱讀順序。
row		<row>標籤記錄念這一系列的閱讀順序。
	name	name 屬性記錄這一系列的標題儲存格座標名稱。

表 3-6 閱讀次序 XML 文件的標籤及屬性意義

「閱讀次序 XML 文件」的標籤結構則如圖 3-16 所示，<titleseq>為第 0 層根元素，之後的第一層是記錄標題的<titles>標籤，以及記錄閱讀順序的<seq>標籤。<titles>標籤的子標籤則是記錄每一個行標題或列標題的<title>標籤。

<seq>的子標籤則是記錄閱讀全部資訊的<all>標籤，以及念某一行的<col>標籤，和念某一列的<row>標籤。在這個文件內，並沒有去記錄表格的真實資訊內容，而是只以儲存格座標名稱就足夠來記錄儲存格的閱讀順序了。

關於「閱讀次序 XML 文件」的 Schema 請參見本論文附錄四。

圖 3-17 則是一個「行列型表格」的閱讀次序 XML 文件，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。

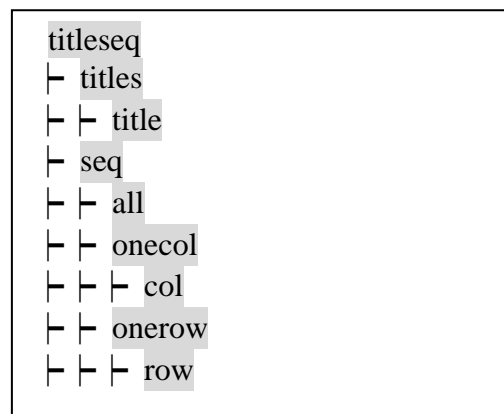


圖 3-16 「閱讀次序 XML 文件」的標籤結構

```

<?xml version="1.0" encoding="UTF-8" ?>
<titleseq model="column&row" row_num="8" column_num="5">
- <titles>
  <title name="col">$r1$c1</title>
  <title name="col">$r1$c2</title>
  <title name="col">$r1$c3</title>
  <title name="col">$r1$c4</title>
  <title name="col">$r1$c5</title>
  <title name="row">$r1$c1</title>
  <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,$r3$c1,$r1$c2,$r3$c2,$r1$c3,$r3$c3,$c4,$r1$c5,$r3$c5,$r4$c1,$r1$c2,$r4$c2,$r1$c3,$r4$c3,$r1$c4,$r4$c4,$r1$c5,$r4$c5,$r5$c1,$r1$c2,$r5$c3,$r1$c4,$r5$c4,$r1$c5,$r5$c5,$r6$c1,$r1$c2,$r6$c2,$r1$c3,$r6$c3,$r1$c4,$r6$c4,$r1$c5,$r6$c5,$r7$c2,$r1$c3,$r7$c3,$r1$c4,$r7$c4,$r1$c5,$r7$c5,$r8$c1,$r1$c2,$r8$c2,$r1$c3,$r8$c3,$r1$c4,$r8$c4,$r1$c5,$c5,</all>
- <onecol>
  <col name="$r1$c2">$r1$c2,$r2$c1,$r2$c2,$r3$c1,$r3$c2,$r4$c1,$r4$c2,$r5$c1,$r5$c2,$r6$c1,$r6$c2,$r7$c2,$r8$c1,$r8$c2,</col>
  <col name="$r1$c3">$r1$c3,$r2$c1,$r2$c3,$r3$c1,$r3$c3,$r4$c1,$r4$c3,$r5$c1,$r5$c3,$r6$c1,$r6$c3,$r7$c3,$r8$c1,$r8$c3,</col>
  <col name="$r1$c4">$r1$c4,$r2$c1,$r2$c4,$r3$c1,$r3$c4,$r4$c1,$r4$c4,$r5$c1,$r5$c4,$r6$c1,$r6$c4,$r7$c4,$r8$c1,$r8$c4,</col>
  <col name="$r1$c5">$r1$c5,$r2$c1,$r2$c5,$r3$c1,$r3$c5,$r4$c1,$r4$c5,$r5$c1,$r5$c5,$r6$c1,$r6$c5,$r7$c5,$r8$c1,$r8$c5,</col>
</onecol>
- <onerow>
  <row name="$r2$c1">$r2$c1,$r1$c2,$r2$c2,$r1$c3,$r2$c3,$r1$c4,$r2$c4,$r1$c5,$r2$c5,</row>
  <row name="$r3$c1">$r3$c1,$r1$c2,$r3$c2,$r1$c3,$r3$c3,$r1$c4,$r3$c4,$r1$c5,$r3$c5,</row>
  <row name="$r4$c1">$r4$c1,$r1$c2,$r4$c2,$r1$c3,$r4$c3,$r1$c4,$r4$c4,$r1$c5,$r4$c5,</row>
  <row name="$r5$c1">$r5$c1,$r1$c2,$r5$c2,$r1$c3,$r5$c3,$r1$c4,$r5$c4,$r1$c5,$r5$c5,</row>
  <row name="$r6$c1">$r6$c1,$r1$c2,$r6$c2,$r1$c3,$r6$c3,$r1$c4,$r6$c4,$r1$c5,$r6$c5,</row>
  <row name="$r7$c1">$r7$c1,$r1$c2,$r7$c2,$r1$c3,$r7$c3,$r1$c4,$r7$c4,$r1$c5,$r7$c5,</row>
  <row name="$r8$c1">$r8$c1,$r1$c2,$r8$c2,$r1$c3,$r8$c3,$r1$c4,$r8$c4,$r1$c5,$r8$c5,</row>
</onerow>
</seq>

```

圖 3-17 「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。為了解說方便，每一個類型的表格都會配合一個表格示意圖來解說。

1. 以行為主型(Column-wise)

圖 3-18 是一個以行為主型的表格示意圖，如圖所示，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」，而當電話使用者選擇唸「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」，注意此處的標題 A 與標題 B 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。

標題 A	標題 B
資料 1	資料 2
資料 3	資料 4

圖 3-18 以行為主型表格示意圖

## 2. 以列為主型(Row-wise)

圖 3-19 是一個以列為主型的表格示意圖，如圖所示，此表格的兩個標題是「標題 A、標題 B」；唸某一列時，當電話使用者選擇唸「標題 A」這一行，VTG 系統會唸出「資料 1、資料 2」，而當電話使用者選擇唸「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」，注意此處的標題 A 與標題 B 會穿插在語句中重複出現，目的也是為了讓表格內容更容易地讓電話使用者了解。

標題 A	資料 1	資料 2
標題 B	資料 3	資料 4

圖 3-19 以列為主型表格示意圖

### 3. 行列型(Column & Row)

圖 3-20 是一個行列型的表格示意圖，如圖所示，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，當電話使用者選擇唸「標題 A」這一行，VTG 系統會唸出「資料 1、資料 3」，而當電話使用者選擇唸「標題 B」這一行，VTG 系統會唸出「資料 2、資料 4」；唸某一列時，當電話使用者選擇唸「標題 C」這一系列，VTG 系統會唸出「資料 1、資料 2」，而當電話使用者選擇唸「標題 D」這一系列，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」，注意此處的標題 A、標題 B、標題 C、標題 D 會穿插在語句中重複出現，目的也是為了讓表格內容更容易地讓電話使用者瞭解。

另外，這一類型的表格有一種有別於其他種類型表格的查詢方式—「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」，當電話使用者說出「標題 B 標題 D」或者「標題 D 標題 B」的時候，VTG 系統會唸出「資料 4」，以此類推。

	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 3-20 行列型表格示意圖

### 4. 跨行型(Column-span)



圖 3-21 是一個跨行型的表格示意圖，如圖所示，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」，而當電話使用者選擇念「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」，注意此處的標題亦會穿插在語句中重複出現，目的也是為了讓表格內容更容易地讓電話使用者瞭解。

標題 A	
標題 B	標題 C
資料 1	資料 2
資料 3	資料 4

圖 3-21 跨行型表格示意圖

#### 5. 跨列型(Row-span)

圖 3-22 是一個跨列型的表格示意圖，如圖所示，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」這一系列，VTG 系統會唸出「資料 1、資料 2」，而當電話使用者選擇念「標題 A 標題 C」這一系列，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」，注意此處的標題亦會穿插在語句中重複出現，目的也是為了讓表格內容更容易地讓電話使用者瞭解。

標題 A	標題 B	資料 1	資料 2
	標題 C	資料 3	資料 4

圖 3-22 跨列型表格示意圖

### 6. 跨行跨列型(Column-span & Row-span)

圖 3-23 是一個跨行跨列型的表格示意圖，如圖所示，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，當電話使用者選擇唸「標題 A 標題 B」這一行，VTG 系統會唸出「資料 1、資料 3」，而當電話使用者選擇唸「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，當電話使用者選擇唸「標題 D 標題 E」這一行，VTG 系統會唸出「資料 1、資料 2」，而當電話使用者選擇唸「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」。注意此處的標題亦會穿插在語句中重複出現，目的也是為了讓表格內容更容易地讓電話使用者瞭解。

		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 3-23 跨行跨列型表格示意圖

### 3.2.4 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，以下就各個種類分別說明語音網頁的對話流程。

1. 「以行為主型表格」與「跨行型表格」：如圖 3-24 所示，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「只念標題」時，便會聽到標題內容；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「念某一欄」(也就是念某一行)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。

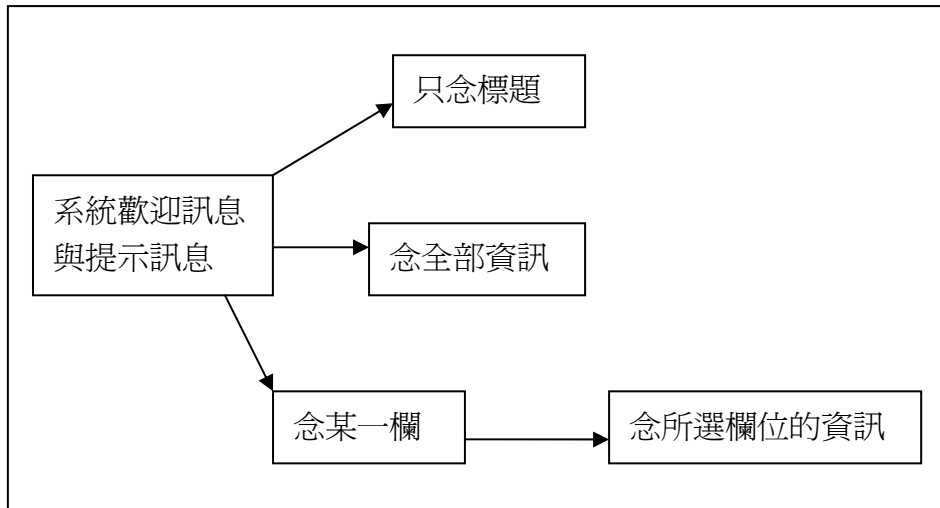


圖 3-24 「以行為主型表格」與「跨行型表格」對話流程

2. 「以列為主型表格」與「跨列型表格」：如圖 3-25 所示，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「只念標題」時，便會聽到標題內容；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。

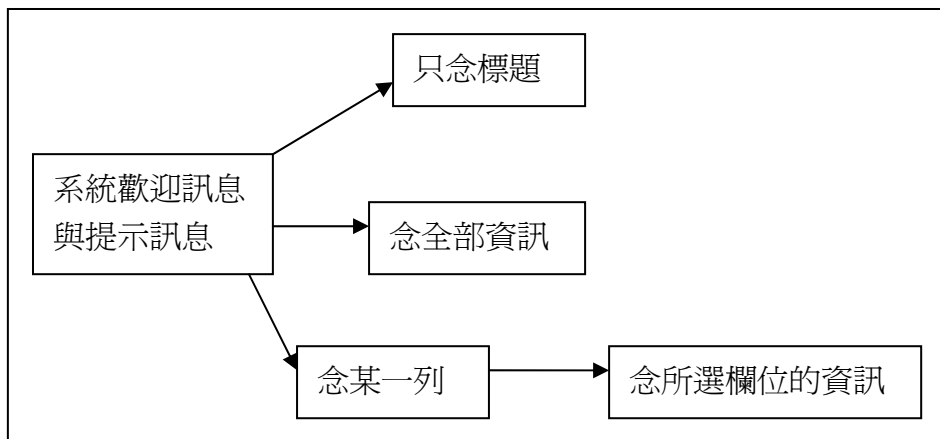


圖 3-25 「以列為主型表格」與「跨列型表格」對話流程

3. 「行列型表格」與「跨行跨列型表格」：如圖 3-26 所示，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。

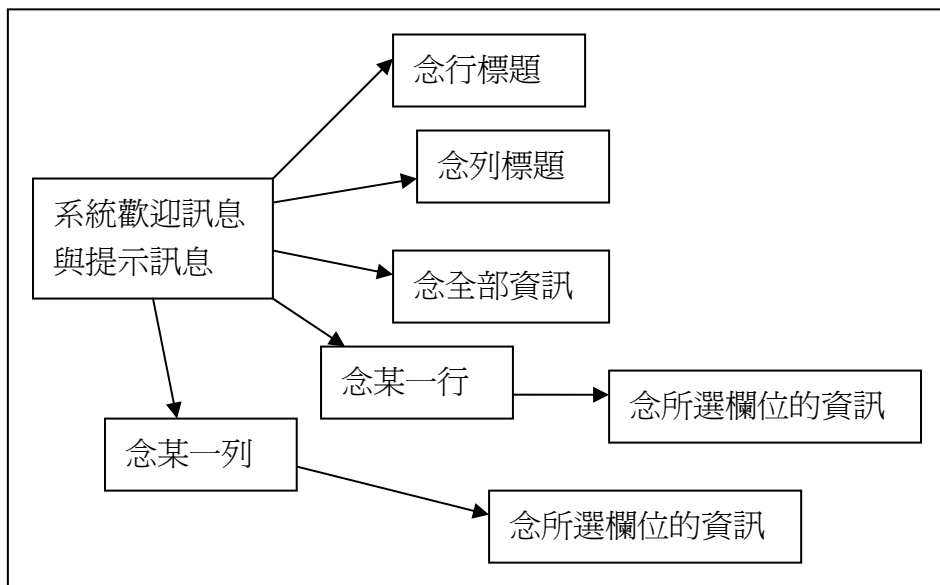


圖 3-26 「行列型表格」與「跨行跨列型表格」對話流程

以上是六種類型表格轉換聽覺呈現模式的機制，除了這些以外，「行列型表格」還有一種有別於其他類型表格的查詢方式：查某一筆資訊。當使用者想要查某一筆資訊時，只要說出想查詢的行標題與列標題，系統就會告訴使用者在這個行標題與列標題對應下的某一筆資料。圖 3-27 是查詢某一筆資料之示意圖，由圖中我們很清楚的看出來「行列型」表格很適合使用這一種查詢方式。

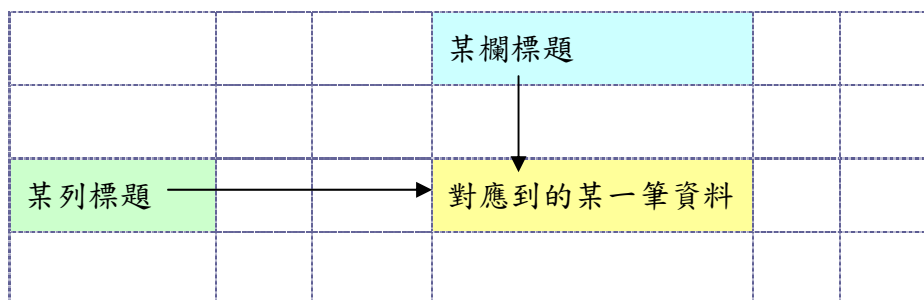


圖 3-27 查詢表格的某一筆資料示意圖

### 3.3 資料型表格資訊對話模式

在網際網路上的表格中，我們發現除了網頁型表格之外，還有另外一種比較複雜的表格，此表格在概念上不是一般所見的二維結構畫面，而是至少有三維結構的表格，我們將此種類型的表格稱為「資料型表格」，資料型表格雖難以使用單純的 HTML 形式來顯示，但是卻可以使用 XML 文件來記錄，並且配合 VoiceXML 的語音技術，就可將此表格的功能發揮出來。我們設計了一個資料型表格的實際例子—「語音行事曆」。一般我們所見的月曆，雖然是二維的畫面，但若要是在月曆的某一天裡填上一至多個當天計畫的事項時，這一至多個事項可以想像成彷彿就是很多儲存格疊在某一天的那一個儲存格裡一樣，所以行事曆是一種三維的表格。

#### 3.3.1 語音行事曆的行事項目

一般來說，行事曆裡面的行事項目，是依每天不同的計畫而定，在種類上是不會有個數限制的，但是由於語音行事曆是配合 VoiceXML 語音技術來實作，受到 VoiceXML 的文法規定所限制，所以語音輸入的項目無法是任意的字詞。因此我們將行事曆的行事項目內容訂為最多 10 個，電話使用者除了可以口語輸入外，熟悉系統的電話使用者亦可直接使用電話數字鍵 0~9 來快速輸入。但是為了讓這個系統更有彈性，所以這些項目的內容也是可以變動的，但

要由行事曆系統管理者來管理，管理者可以將最常使用的行事項目放入語音行事曆系統的行事曆項目中，讓電話使用者可以根據這些項目來新增他的行事曆內容。

我們使用一個「行事項目 XML 文件」來記錄所有的行事項目，表 3-7 說明了行事項目 XML 文件的標籤及屬性意義。注意每個行事項目都有其簡稱與對應的電話輸入按鍵，其目的在於讓熟悉系統的電話使用者能夠更方便更快速的建立行事曆內容。

標籤名稱	屬性名稱	說明
AllItems		<AllItems>標籤為行事項目 XML 文件的根元素。
Item		<Item>標籤記錄了一個行事項目的資訊。
	dtmf	dtmf 屬性記錄這個行事項目對應的電話輸入按鍵。
event		<event>標籤記錄了行事項目的內容。
abbr		<abbr>標籤記錄了行事項目的簡稱。

表 3-7 行事項目 XML 文件的標籤及屬性意義

「行事項目 XML 文件」的標籤結構則如圖 3-28 所示，<AllItems>為第 0 層根元素，之後的第一層是記錄各個行事項目的<Item>標籤，每個<Item>標籤又有<event>與<abbr>兩個子標籤，分別記錄行事項目的內容及簡稱。當語音行事曆系統管理者要維護這些項目時，其實也就是透過管理介面來修改這個行事項目 XML 文件的內容。

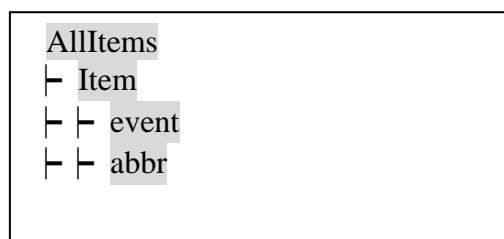


圖 3-28 「行事項目 XML 文件」的標籤結構

關於「行事項目 XML 文件」的 Schema 請參見本論文附錄五。

```

<?xml version="1.0" encoding="Big5" ?>
<AllItems>
<Item dtmf="1">
  <event>參加會議</event>
  <abbr>開會</abbr>
</Item>
<Item dtmf="2">
  <event>參加朋友餐會</event>
  <abbr>聚餐</abbr>
</Item>
<Item dtmf="3">
  <event>郊遊踏青逛街看電影</event>
  <abbr>休閒</abbr>
</Item>
<Item dtmf="4">
  <event>到學校上課</event>
  <abbr>上課</abbr>
</Item>
<Item dtmf="5">
  <event>參加考試</event>
  <abbr>考試</abbr>
</Item>
<Item dtmf="6">
  <event>到客戶公司進行簡報</event>
  <abbr>拜訪客戶</abbr>
</Item>
<Item dtmf="7">
  <event>週末或國定假日</event>
  <abbr>假日</abbr>
</Item>
<Item dtmf="8">
  <event>作業或工作進度的繳交日期</event>
  <abbr>交作業</abbr>
</Item>
<Item dtmf="9">
  <event>回到家鄉探望親友</event>
  <abbr>返鄉</abbr>
</Item>
<Item dtmf="0">
  <event>到外地工作</event>
  <abbr>出差</abbr>
</Item>
</AllItems>

```

圖 3-29 「行事項目 XML 文件」範例

圖 3-29 是一個「行事項目 XML 文件」實際範例，內容有十個行事項目和其簡稱，以及每個項目所對應的電話輸入按鍵。



### 3.3.2 XML 式行事曆內容

就像我們平常會將日常生活的行事曆記錄在記事本上一樣，語音行事曆系統會將行事曆的內容以一個 XML 文件來記載。我們稱此 XML 文件為「行事曆 XML 文件」。一個行事曆 XML 文件記載一個月份的行事曆內容，所以一年十二個月的行事曆，就要十二個行事曆 XML 文件來記錄。表 3-8 說明了行事曆 XML 文件的標籤及屬性意義。

標籤名稱	屬性名稱	說明
calendar		<calendar>標籤是行事曆 XML 文件的根元素。
	yyyy	yyyy 屬性記錄這個行事曆 XML 文件是記載哪一年的行事曆。
	mm	mm 屬性記錄這個行事曆 XML 文件是記載哪一個月的行事曆。
day		<day>標籤記錄了每一天的行事曆內容。
	dd	dd 標籤記錄這一天的日期。
	week	week 標籤記錄了這一天是星期幾。
event		<event>標籤記錄了當天計畫的行事項目。

表 3-8 行事曆 XML 文件的標籤及屬性意義

「行事曆 XML 文件」的標籤結構則如圖 3-30 所示，<calendar>為第 0 層根元素，之後的第一層是記錄在這一個月裡每天行事曆內容的<day>標籤，所以一個月有幾天，就有幾個<day>標籤，在<day>標籤的下一層則是零至多個<event>標籤，記載每一天零至多個行事項目。

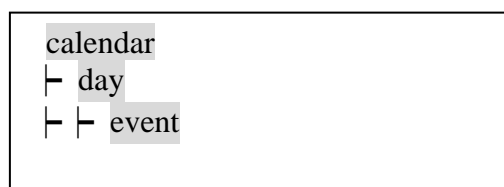


圖 3-30 「行事曆 XML 文件」的標籤結構

關於「行事曆 XML 文件」的 Schema 請參見本論文附錄六。

圖 3-31 是一個「行事曆 XML 文件」範例片段，由這個文件的內容我們可

以看出這是記載 2004 年 5 月份的行事曆 XML 文件，且在五月三日星期一的那一天，有兩個預定的行事項目為：交作業、上課。

```
<?xml version="1.0" encoding="UTF-8" ?>
<calendar yyyy="2004" mm="5">
  <day dd="1" week="6" />
  <day dd="2" week="7" />
  <day dd="3" week="1" />
    <event>交作業</event>
    <event>上課</event>
  </day>
  <day dd="4" week="2" />
  <day dd="5" week="3" />
  <day dd="6" week="4" />
```

圖 3-31 「行事曆 XML 文件」範例片段

### 3.3.3 語音行事曆系統的語音呈現模式

在語音行事曆系統中，擔任與電話使用者溝通角色的 VoiceXML 文件，其對話的過程主宰了整個行事曆使用者端的功能，圖 3-32 是語音行事曆系統之 VoiceXML 對話流程。一開始系統會先播放系統歡迎訊息與提示使用者有查詢行事曆內容與新增行事曆內容等選項可以選擇，當使用者選擇查詢行事曆內容時，系統會要求使用者說出要查詢幾月幾日的行事曆內容，待使用者說出日期後，VoiceXML 便會將日期交由程式運算，查詢那個月份的行事曆 XML 文件中該天的行事曆內容，然後語音行事曆系統會告知使用者查詢的結果；而當使用者選擇新增行事曆內容時，系統會要求使用者說出要新增幾月幾日的行事曆內容，待使用者說出日期後，系統會再詢問使用者要新增哪一個行事項目，當使用者也回答了行事項目，VoiceXML 便會將日期與新增事項交由程式處理，

將行事項目新增到該天的行事曆裡，並告訴使用者行事曆已經新增成功。

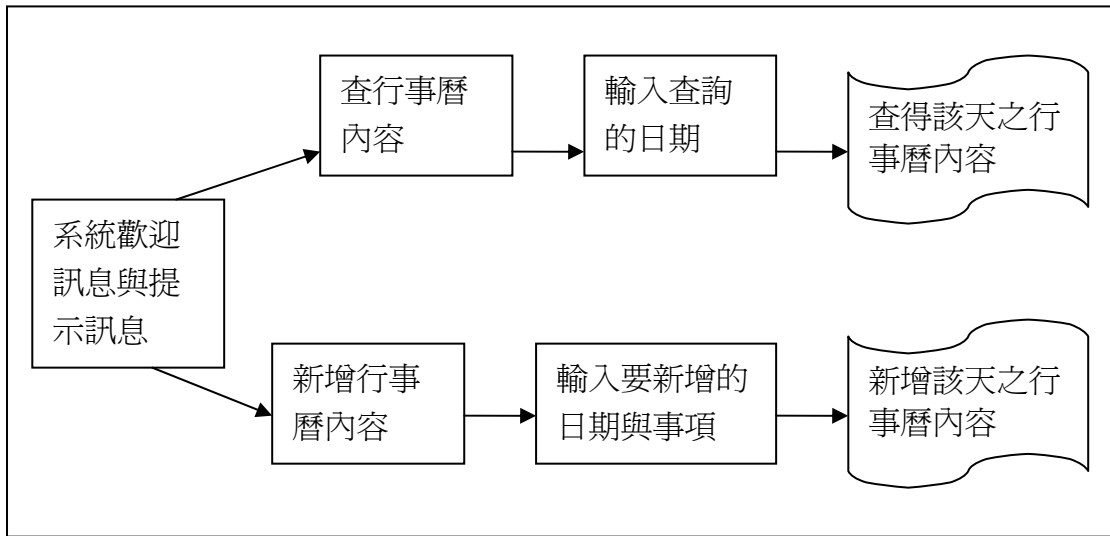


圖 3-32 語音行事曆系統之 VoiceXML 對話流程