一種字元轉換至中文資料庫之方法
An Approach on Transforming Characters to Chinese Databases

姚良穎 黃玄煒 李政宏 †詹景裕
國立台灣海洋大學
電機工程研究所
†國立台北大學
電機工程研究所
e-mail: { M94530083, B0136, D93530006} @mail.ntou.edu.tw
†e-mail: gejan@mail.ntpu.edu.tw

摘要

在本論文中，我們提供 Unicode 與 ASCII 編碼互相轉換的機制，其可將使用者輸入之 Unicode 中文字碼，經轉換處理後以 ASCII 內碼格式儲存於中文資料庫中；於讀取資料庫時，則透過逆轉換處理程序，將 ASCII 內碼格式字串再轉換為 Unicode 字元，並顯示於前端使用者介面。其目的在於改善使用 Big5 編碼的中文資料庫無法接受 Unicode 字元的問題，以增進中文資料庫的字元相容性、正確性與可維護性。

關鍵詞：Unicode、Big5、中文資料庫、字元轉換、字元編碼。

Abstract

A transforming approach is proposed in this paper. A mechanism is used to do the transformation between Unicode and stuffed ASCII code, where the Unicode is obtained from the user input and the stuffed ASCII code is used by the database. The major purpose of this approach is to provide the Unicode support for the Chinese database using Big5 coding. Furthermore, the compatibility, correctness, and maintainability of the Chinese database can also be improved.

Keywords: Unicode, Big5, Chinese database, character transformation, character encoding.

一、緒論

前端使用者介面與後端資料庫系統是電腦資訊系統的二大元件，用以處理使用者的輸出入與其資料的存放。然而，於更新處理中，操作介面僅需更新軟體或硬體元件，而資料庫系統除需考量舊資料的備份、轉存、匯出與匯入問題外，因其所存放的資料多屬歷史資料，尚須考量資料內容或格式變更後所產生的相容或維護的問題。當前端電腦更新到可以處理 Unicode 編碼字元，而後端資料庫卻只能使用 Big5 編碼儲存資料，其中存在著字元相容之問題。

元智大學開發之教務資訊系統[8]，所使用之 Microsoft SQL Server [5]並不支援將 Unicode 編碼字元儲存至舊式中文資料庫中。CNS11643 中文標準交換碼將字元字碼標準化[3]，Unicode 補完計畫利用字碼對應表做必需的字元轉換[7]，兩者提供了有關文字輸出入與顯示之處理；但，對於後端資料庫的 Big5 編碼限制，仍沒提供任何支援。以致前端使用者做任何輸入模式或字碼表的修改，無法被資料庫辨識，中文字元之轉換與儲存問題依然存在於前述技術中。例如，「峯」、「綉」、「塋」等字，於資料庫無法判讀的情況下，該些字元將會以「?」的符號儲存在資料庫中，造成資料內容的錯誤。對於中文資料庫缺字的問題，一直是以造字程式解決；但電腦須安

起始識別符號	ASCII式中文內碼	分隔符號/結束識別符號

圖 1　ASCII 字元化中文內碼格式

裝造字程式，否則無法正常顯示該造字字元，對於維護整體資訊的完整性亦會產生相當困擾[9,13]。

因造字方式與字碼轉換都不能解決中文資料庫的字元相容問題，如何使用 Unicode 編碼來處理輸出入與正確儲存中文字元至中文資料庫中，乃為本文所欲解決之問題。基於 Microsoft 的 ASP.NET 伺服端網頁技術[6]，搭配.NET Framework 類別庫，本文的處理分為二部份，第一部份是針對使用者介面文字編碼的轉換與儲存，將使用者所輸入的字元字串加以判別與編碼，並將 Big5 編碼範圍以外的字元轉換成為 ASCII 字元化內碼；於資料庫的儲存中，透過 ODBC（Open Database Connectivity）所建立的資料庫連結，將 ASP.NET 處理後的字串儲存至 SQL Server 中文資料庫中。第二部份是有關於文字的輸出，透過逆轉換處理，將資料庫中 ASCII 式中文內碼轉換為 Unicode 編碼字元，使之可以直接在網頁上呈現字元樣式，並可進行資料內容的修編。上述的處理過程，不僅不需變更影響到 SQL Server 原始結構，也可以不須或大幅減少造字程式的使用，並且能夠提供資訊系統更正確的資訊內容。

於往後的篇幅中，第二節將介紹相關文字編碼原則與相關研究之介紹；第三節將詳述本論文之處理架構、程序與編碼格式；第四節提供實作展示；第五節則為本文總結。

二、字元編碼

2.1 中文編碼介紹

目前全世界存在許多不同的字元集(Character Set/字集)，每個國家皆有自己的標準字集，再依不同需求訂定不同的字元編碼(Character code/字碼)；例如：Big5、EUC、TCA 碼的字集均使用 CNS11643 中文標準交換碼[3]的字集。

字碼表示每一個字元在字集中的編碼，依照既定的編碼範圍，將一固定的字集依照某種訂定的方法排列，分別賦予每個字元一個對應的二進位碼，通常以十六進制數字的區、位表示法，將字元的高、低位元組數字化。整個中文字庫的結構可視為一個二度空間[13]，由代表高位元與低位元組的二條縱線與橫線所組成的平面。二條線所交叉的每一個點都是一個編碼位置。以一般七位元組組成的字面，扣除控制字元，共有 94×94 個編碼位置；而以雙八位元組所組成的字面，扣除控制字元，則共有 256×256 個編碼位置。

Big5 最初是委託民間所研發設計，運用時並不如預期的順利[2,9,10,12]。儘管如此，Big5 編碼還是因為高市場佔有率而儼然成為了業界標準，90%以上的中文字碼皆使用 Big5 碼。Big5 碼為雙位元組的內碼系統，其高位元組區間為：81_{16} – FE_{16}（避開控制字元集 C1）共計 126 碼；低位元組區間為：40_{16} – $7E_{16}$（63 碼）與 $A1_{16}$ – FE_{16}（94 碼）（避開控制字元集與部份 ASCII 符號[13]），共計 157 碼。所以可編碼位共計可定義為 126×157＝19,782 個字碼。

2.2 ASCII 字元化中文內碼

現代電腦與通訊設備都是以 7 位元或 8 位元的位元組做為處理或傳輸文字資料的基本單位。但中文字集的字數非常龐大，至少需要 2 個位元組，因此需要一種特別的機制，才能讓電腦識別某 2 個連續的位元組為一個中文字元碼，而不是單純使用二個 ASCII 或延伸 ASCII 字元碼（Extended ASCII Codes）[1]。

針對中文碼的十進制或十六進制數

值，將每一數位都轉換為對應的ASCII碼（簡稱ASCII式中文內碼），並在每一個中文碼之前附加起始識別符號；之後再加附加分隔符號或結束識別符號。其格式如圖一所示，在字元的Unicode字元碼前端加上「&#」做為起始識別符號；後端加上「;」做為結束識別符號。針對這類中文內碼，依序讀取每一個位元組並檢視其是否是起始識別符號，否則就將該位元組視為ASCII字元碼處理；若檢視到其為起始識別符號後，必須再連續讀取位元組直到遇到間隔符號或結束識別符號後為止，並將所讀取到的連續幾個位元組結合為一個中文內碼加以處理。HTML文件有時採用ASCII字元化字元碼來表現非ASCII字元，以便HTML文件可以通行無阻於網際網路而不受某些網路節點只限七位元位元組的資料流（data stream）影響 [13]。例如「峯」這個字的Unicode內碼為「$5CEF_{16}$」，其ASCII十進制為「23791」，在HTML文件中加上起始識別符號與結束識別符號後，所顯示為「峯」字串，經由瀏覽器的編譯後即可以顯示所代表的「峯」字樣。但「峯」一共8個字元，若使用ASCII字元化中文內碼將比直接使用內碼大幅增加檔案的大小，而也是為了網路透通性所必須付出的代價，因此在一般的HTML文件中，並不會使用ASCII字元化中文內碼來編輯網頁檔案。

2.3 Unicode 萬國碼與 UTF-8

Unicode 是在統一全球多國語言至單一編碼的前提下所產生的，其將電腦字集編碼的基本單位擴充為16位元，充份利用了2^{16} =65,536個碼位來容納全世界各種語言的字元與常用符號[4,11]。Unicode 使用16 bits 以形成一字碼，其未限定字元在記憶體、資料庫與網頁的呈現方式；每個字元的顯示必須透過編碼方式進行處理，並可經由對應表(Mapping Table)以與其他的編碼標準進行轉換，例如：Unicode 轉換至 Big5，或 GB(簡體中文)轉換至 Unicode。

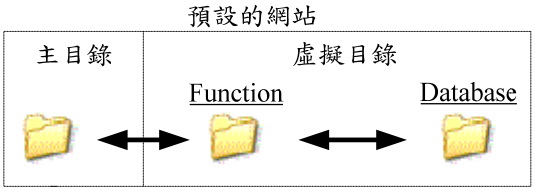

圖 2 網站軟體架構圖

UCS (Universal Multiple-Octet Coded Character Set)係為ISO於1984年發起制定的國際字元集編碼標準。其正規形式(簡稱UCS-4)，為四個 byte 所組成的 32 位元編碼結構，分別代表編碼結構中的 Group(群組)、Plane(字面)、Row(列)與 Cell(格)；其中第 0 群組的第 0 字面稱為「基本多語文字面」，而當電腦系統只使用基本多語文字面的字元碼時，可以省略群組與字面兩個byte，而將字元碼由四個 byte 縮短為兩個byte，稱為 UCS-2 字元碼，也就是習稱的Unicode。

在 ASCII 的位元組環境中，UCS-2 的字元碼被切割成兩個 8 位元的位元組，而失去編碼的意義與關連性；在資訊與通訊領域內只要涉及字元的標準與協定，無不受到 ASCII 的影響與約制，因此 UCS-2 與 UCS-4 都只能當做內碼使用，而無法真正流通於網際網路。為了讓 UCS-2 與 UCS-4 能流通於 ASCII 環境中，ISO 制定了 UTF-8 以解決問題。UTF-8 (UCS Transformation Format)是將原為 UCS-4 字元碼或 UCS-2 字元碼轉換為多個 8 位元的位元組；因 UTF-8 不抵觸 ASCII 的標準與協定，故能代替 UCS-2 與 UCS-4 以達到資訊交換的任務。因此在 HTML 設計中，多以 UTF-8 編碼來達到使 Unicode 能流通於網際網路的目的。

2.4 處理平台架構

本文的硬體架構，是採用基於 N-Tier 架構的.NET Framework 運作平台，網站的軟體架構主要分為三個部份，如圖二所示：網站主目錄，用於提供 URL 下載網頁程式的儲存；function 虛擬目錄，用於存

放網頁間共用之副程式或函數的模組檔：database 虛擬目錄，用於存放 Access 資料庫檔。主目錄文件包括 web.config 和副檔名為.aspx 的檔案，其中 web.config 是 ASP.NET 的組態檔，用來定義網站預設的 XML 文字編碼，可以出現在 ASP.NET Web 應用程式伺服器的任何目錄中，每個 web.config 檔案都會將組態設定套用到其位置所在的目錄與該目錄下所有的虛擬目錄；而副檔名為.aspx 的檔案皆為 ASP.NET 網頁檔，提供網頁文字輸入與顯示的介面。虛擬目錄文件包含 my.vb 和 Big5.mdb，my.vb 為開發階段自行定義的純文字檔，提供整個網站所共同使用的副程式、函數與參數，並使用 include 方式引入每個.aspx 檔案中，以供每個.aspx 網頁檔皆能使用；或以副程式呼叫方式，提供字元轉換功能。Big5.mdb 為 Access 97 格式資料庫檔，存放在 database 虛擬目錄中。文字的傳遞可分為二部份進行，一為網頁與網頁間的資訊傳遞，另一為網頁與資料庫之間的存取。惟兩者均可於主目錄文件中的 web.config 中指定預設編碼的方式和語系。為了讓系統能夠接受 Unicode 字元，故將 web.config 中指定的編碼方式都設定為 utf-8；在網頁編輯時，於每個.aspx 檔案中 HTML 標籤亦指定編碼方式為 UTF-8。該種設定可使網站於處理字元時，一律採用 Unicode 編碼；其原因在於網頁表單物件所輸入的文字外觀，無法判定該文字是否屬於 Big5 編碼系列，且能使系統將使用者輸入的文字一律以 Unicode 編碼處理。本文在系統開發階段，為不影響現行教務系統的舊資料，因此先以 Access 資料庫模擬 SQL Server 資料庫。

三、字元轉換技術

3.1 文字轉換架構

圖三所示為文字轉換架構圖，文字的傳遞是由.aspx 網頁接收所輸入的 Unicode 字元，經過虛擬目錄文件中 my.vb 的字元判別及轉換程序後，將 Unicode 字元或

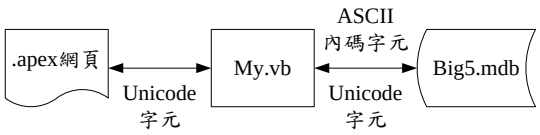

圖 3 文字轉換架構圖

ASCII 字元化內碼格式字串，再以 Big5 編碼儲存在資料庫中。由資料庫讀取文字時，也經由寫在 my.vb 中的逆轉換處理程序將 Unicode 字元顯示於.aspx 網頁上。須注意的是，假如 Unicode 字元被 Big5 編碼支援，則資料庫之寫入/讀出會自動轉換，不會產生相容性問題。會有相容性問題的 Unicode 編碼，來自於其字元不存在 Big5 編碼中。

3.2 ASCII 字元化內碼的轉換

.NET Framework 類別方法的 Strings.Asc 與 Strings.AscW 所能處理的字元碼範圍不相同，Strings.Asc 可傳回輸入字元的單一位元組或雙位元組字元碼，Strings.AscW 則傳回 Unicode 字元碼。因此可以使用 Strings.Asc 與 Strings.AscW 方法，來檢視傳回的輸入字元資訊；以「峰」與「峯」字為例，「峰』字的 ASCII 字元碼為「－20880」，Unicode 字元碼為「23792」；「峯」字的 ASCII 字元碼為「63」，Unicode 字元碼為「23791」。惟於 ASCII table 中所示，ASCII 字元碼 63 所對應的字元是「?」符號，此結果與儲存在舊式資料庫中所顯示的結果相同。因此可以得知，當 Strings.Asc 類別方法處理超過 Big5 編碼範圍以外的字元時，將傳回「?」所代表的「63」ASCII 字元碼；亦即，先前存入之 Unicode 碼於轉換成 Big5 碼時，發生了錯誤，產生字碼相容性問題。

我們可以利用這個特性，來決定是否要將 Unicode 字元轉換為 ASCII 字元化內碼格式。當傳回的 ASCII 字元碼為代表「?」符號的「63」，而 Unicode 字元碼卻不是「63」時，便可以判斷輸入的字元並不是真正的「?」符號，而是 Big5 編碼範圍以外的字元，此時便可將讀取到的 Unicode

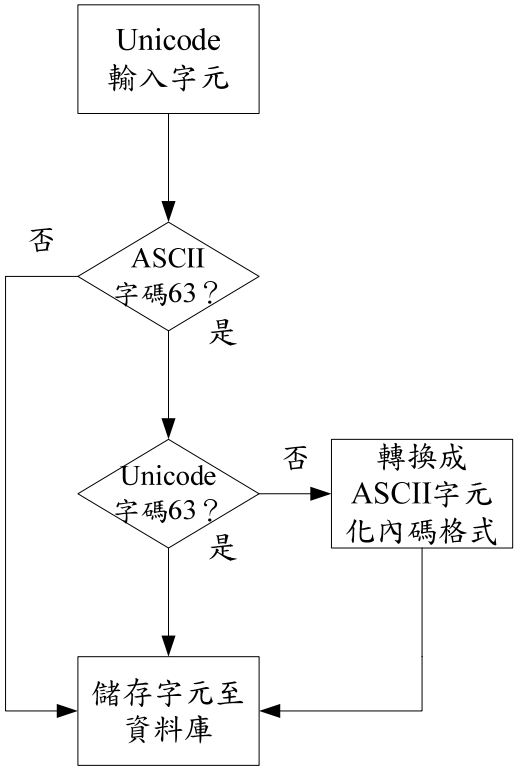

圖 4 字元轉換處理

字元碼轉換為 ASCII 字元化內碼格式；而若 ASCII 字元碼並非「63」，即代表輸入的字元一定是在 Big5 編碼範圍內，此時並不對該字元進行任何轉換。以「峯」字為例，新形成的 ASCII 字元化內碼字串即為「峯」。圖四為單一字元判斷是否要進行字元碼轉換的處理流程。

3.3 由資料庫讀取字元資料

由資料庫讀取之字元資料，有二種方式顯示於網頁，分別為 HTML 網頁文件顯示與 Web 表單物件顯示。由於 HTML 文件完全支援 ASCII 字元化內碼，因此直接讀取資料庫內容並顯示於網頁；但 Web 表單物件並不支援 ASCII 字元化內碼，只會顯示 ASCII 字元化內碼字串，故須做逆轉換處理，以得到正確的顯示。以「峯」字元為例，HTML 網頁可以顯示字型，但 Web 表單物件只會顯示 ASCII 字元化內碼字串。

Unicode 字元在被轉換成 ASCII 字元化內碼前，為一個或兩個字元；但經轉換

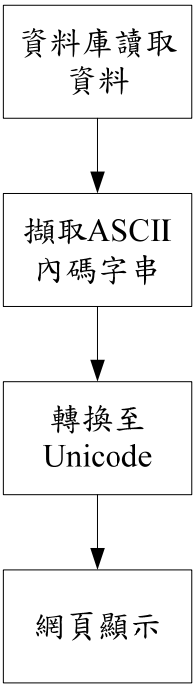

圖 5 字元逆轉換處理

後，以「峯」字為例，其 ASCII 字元化內碼「峯」共佔了 8 個字元。因此逆轉換處理過程中須先去除起始/結束識別符號；再將擷取到的字串型態轉換為數值型態以及 Unicode 字元，並顯示於 Web 表單物件上。圖五所示為 ASCII 字串轉換回 Unicode 字元之處理程序。

3.4 特殊字元之處理

因 ASCII 字元化內碼具有較大的儲存長度，為避免資料庫經轉換處理後變得過於巨大，故於前述轉換與逆轉換處理中，只對 Big5 編碼外的字碼進行 ASCII 字元化內碼轉換；對 Big5 編碼內的字碼，仍進行一般的 Big5 編碼轉換。惟於使用者的輸入字串含有「&」、「#」或「;」等符號時，其與 ASCII 字元化內碼識別符號產生衝突，於逆轉換處理時，將會產生錯誤。由於 HTML 網頁或 Web 表單物件中，該些符號均以 Unicode 字元碼表示，該些符號也一律轉換為 ASCII 字元內碼。於資料庫讀取時，該些符號因與 ASCII 字元化內碼的識別符號有所不同，故不會產生逆轉換處理時發生錯誤問題。

圖 6 未經轉換處理的資料庫內容顯示

圖 7 經轉換處理的資料庫內容

四、實例應用

限於篇幅，本節提供一網頁可取得之資訊，作一簡單的轉換處理說明。下列資料中含有 Big5 編碼範圍以外字元、特殊符號、一般符號與阿拉伯數字所混合的字串：

❶、教務處註冊組：(02)24622192#1015;1016
①、組長：蔡宏營(分機：1015)
②、組員：何水綉(分機：1021)
❷、輪機工程學系：(02)24622192#7110
①、系主任：王正平 教授(分機：7117)
②、兼任老師：楊仲箎 副教授

其中❶、❷、①與②為特殊符號；「綉」與「箎」二字為特殊字元，屬於 Big5 編碼範圍外的字元，而整組字串則包含有阿拉伯數字與「#」、「;」等符號。假如資料未經轉換處理，於讀出資料庫資料並顯示於網頁的結果，如圖六所示。經過判別並轉換後之資料庫資料，直接以 Access 軟體開啟資料庫來檢視，其顯示內容如圖七所示；其中，原本為符號「#」符號的字元在資料庫中以「#」字串取代；「;」符號以「;」字串取代，而特殊符號與字元也分別轉換為 ASCII 字元化內碼的格式儲存。

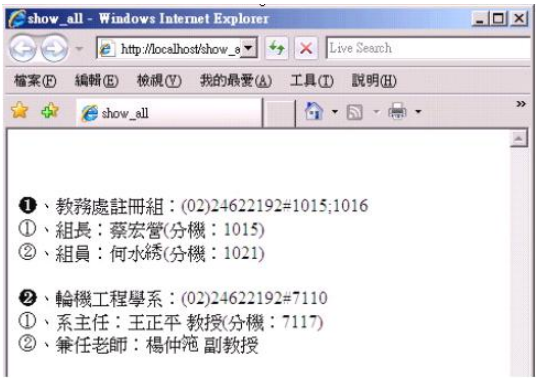

圖 8 HTML 網頁文件顯示結果

圖 9 Web 表單物件轉換後的結果

因為 ASCII 字元化內碼完全相容於 HTML 文件，不論是否經過逆轉換處理程序，皆能呈現原始文字內容；但必須要資料存入資料庫之前，有做過 ASCII 字元化內碼轉換處理程序。其結果都如圖八所示。圖九表示經過逆轉換處理程序後的表單物件內容，每個字元皆為儲存至資料庫前的字元，重新呼叫至表單物件後，將可進行文章內容的再編輯。

五、結論與展望

本文之目的在於解決資訊系統中使用舊式中文資料庫無法儲存所有 Unicode 字元的問題，本論文所提出的方法是以 Unicode 做為使用者介面文字的編碼選擇，經由.NET Framework 類別函式庫的轉換後，以 ASCII 字元化內碼的格式儲存；

配合逆轉換處理的程序，能將 ASCII 字元化內碼還原為 Unicode 字元，以供使用者再編輯資料內容。整體而言，本技術之優點在於使用習知的 Unicode 與 ASCII 編碼，以取代習用造字功能，並大幅提高資料庫系統之資料內容正確性與可維護性；並且 ASCII 字元化內碼完全相容於 HTML 文件，因此在網頁瀏覽時，並不需要特別的逆轉換處理程序，也能顯示正確的資訊內容。又使用本論文所提出字元轉換的方式所建置的 Web 資訊系統，未受限於罕見中文字元、特殊符號或數學公式等之使用，只要在 Unicode 編碼範圍內的字元皆能適用。目前已嘗試使用於海洋大學教務系統的課程綱要內容，並將逐步擴充至多種字元之轉換。雖然解決舊式資料庫字元儲存問題的最根本方式就是直接使用能接受 Unicode 編碼的資料庫；惟於大型資訊系統中既有資料的保存與搬移，需要耗費大量的人力與時間成本，甚或比建置新的系統更為龐大。故本文除可提供舊式中文資料庫之相容性保證外，如何將舊有資料庫中已存在的自造字元，能以 Unicode 字元對應方式取代之，或併合造字功能之聯合操作，以增進資料庫字元之更新能力，則是未來研究的方向。

參考文獻

[1] ASCII Table，http://www.asciitable.com。

[2] CNS11643 中文標準交換碼全字庫，中文碼介紹：Big-5 碼，http://www.cns11643.gov.tw/web/big5/index.html。

[3] CNS11643 中文標準交換碼全字庫，中文碼介紹，http://www.cns11643.gov.tw/web/word.jsp。

[4] International Organizational for Standardization，http://www.iso.org/。

[5] Microsoft 技術支援服務，說明在 SQL Server 中儲存 UTF-8 資料，http://support.microsoft.com/kb/232580/zh-tw。

[6] Microsoft Development Network，ASP.NET，http://msdn2. microsoft.com/en-us/library/aa286485.aspx。

[7] Unicode 補完計畫，http://uao.cpatch.org/。

[8] 元智大學，國立臺灣海洋大學教務系統電腦化系統說明書，1999。

[9] 邱菊梅，中文電腦缺字研究，碩士論文，玄奘人文社會學院中國語文研究所，2002。

[10] 許旭正，繁簡體轉換在 ERP 系統的應用與研究，碩士論文，私立東海大學資訊工程與科學研究所，2006。

[11] 陳威丞，Unicode 全文檢索之研究與實作，碩士論文，國立中正大學資訊工程研究所，2000 年。

[12] 黃大一，中文字碼-萬碼奔騰：一碼當先，永麒科技，1992 年二版。

[13] 曾士熊，認識中文碼，中央研究院計算中心，2004/11/15，http://www.sinica.edu.tw/~bear/charcodes/codeindex.htm。