

多站台網路預約系統之 AJAX 即時資料更新機制

張逸中、黃耀文
致遠管理學院數位資訊學系
台南、中華民國
ycc@dwu.edu.tw

摘要

以網際網路為基礎的預約系統通常必須面對資料不易即時更新的問題，因為多使用者會同時參考著一個內容隨時變動中的資料庫，且其變化又攸關使用者操作系統的目的，無法自動更新的系統必然會造成困擾而難以使用。本文以一個復健科診所的網路預約系統為例，使用 AJAX (Asynchronous JavaScript and XML) 技術的網頁幕後通訊與局部更新技術設計，可達到即時自動更新所有站台網頁資料的目的，且更新過程中操作者的打字或滑鼠點選等操作都完全不受影響，即時同步的程度甚至達到逐字更新的地步，達到近似桌面程式系統的所謂理想網際網路操作環境(Rich Internet Application, RIA)。

關鍵字: AJAX, RIA, Web, 資料同步, 即時更新

I. 前言

傳統的網際網路通訊方式以客戶端主控為原則，客戶端首先對伺服器端提出網頁要求，伺服器端依據資料庫資料搜尋結果產生動態網頁或直接找出靜態網頁回傳客戶端，之後除非客戶端提出新的需求，如重新整理或要求新的網頁，否則伺服器端無法主動更新客戶觀看中或操作中的網頁。

這種機制在伺服器端資料庫接受某客戶要求而更新內容時無法即時主動的通知其他使用中的客戶，對於執行網路訂票或其他預約功能的系統而言，這是一個重要的議題。因為客戶端操作的主要目的是依據最新資料選擇是否可以進行預約，資料過時將會導致許多錯誤或無效的操作。

目前類似功能的網際網路架構(web-based)系統處理這個問題的方式主要是在新資料寫入資料庫之前的查核，譬如其他客戶若已搶先預約則回覆客戶預約失敗。相對於桌上型程式系統來說，可以用 TCP 通訊協定直接撰寫客戶端與伺服器端的通訊機制，一樣可以使用網路，但不受限於 http 協定中必須客戶端主動的限制，因此可以即時主動的由伺服器端更新客戶端顯示的資料，有如一般車站機場的即時資訊資料看板。因此上述在網際網路系統上因為資料已經過時，客戶卻不知情而繼續預約又被拒絕的情況便不會發生，可以有效提升系統操作的效率。因此目前多數需要高效率的類似系統多半仍採用非網際網路程式的桌上型程式架構進行設計。但是考慮到系統設計與安裝的難度及成本，對於一般中小型企業而言，如果可以順利解決資料同步的困擾，網際網路架構的系統仍是較佳的選擇。

如前所述，網際網路系統在此議題上的主要劣勢在於客戶端資料即時更新的機制。事實上即使可以使伺服器端主動推播更新資料，如果遵守傳統的網頁回傳(Postback)機制，也會產生客戶端畫面閃爍，以及客戶端的資料輸入動作被中斷的問題，使用者的經驗與觀感亦將大大不如桌上型系統。

欲解決上述問題，可以採用 AJAX 技術(Asynchronous JavaScript and XML)[1]重新設計網頁。此項技術內容相當廣泛，對於本議題而言，主要是可以經由設計於網頁中的 JavaScript 程式，或稱 AJAX 引擎，提供無需執行網頁回傳動作的伺服器與客戶端即時通訊。通過 AJAX 引擎，我們可以讓伺服器端資料更新時主動通知並傳送更新資料(非完整網頁)到客戶端，但不會觸動網頁的全面更新，而是由 JavaScript 程式執行精準的網頁局部資料更新。

這個解決方案除了資料可以即時的與資料庫同步，解決資訊過時的問題，也同時讓使用者有了類似桌上型系統的使用經驗，亦即所謂的 Rich Internet Application (RIA)[2]的效果。首先網頁不會因為更新而閃爍跳動，其次如果更新的資料不是使用者操作中的欄位，使用者的操作可以繼續。舉例來說，當您在 A 欄位輸入一個姓名至第一個字時恰巧 B 欄位資料更新，您可以完全不受影響繼續輸入名字的其他字，即使是中文打字等輸入動作也不會受到影響。

II. 實務系統概述

本示範系統為某復健科診所預約診療時間之網際網路程式。該診所有十餘名復健師，每天區分早午晚三個時段各有七至九位復健師值班，各復健師名下再細分為以十分鐘為間隔的小時段作為預約單位。系統主要資料包含復健師名單與客戶預約紀錄，依據診療師名單，系統有一排程網頁可排定各日期與時段的復健師班表，畫面如圖 1。

排程後產生的主要預約畫面如圖 2 所示，在此頁面直接選擇各時段的某位復健師名字，該時段預約名單與細節時間便會顯示於右側表格。操作者經過帳號密碼登入取得權限後，即可直接編輯修改預約名單。一般情況下，病患會依時段或指定特定復健師進行預約，因此本系統同時設計有依據時段預約(圖 3)，以及依據個別復健師班表預約的方式(圖 4)的網頁。

上述的三個可進行預約操作的頁面參考的是相同資料庫的資料，每個頁面切換時都會重新讀取資料庫資料建構新的網頁，因此各使用者本身操作的不同頁面之間資料可以確保同步無虞，本文討論的重點將是不同使用者之間的資料同步機制。

III. 資料更新的訊息流程

資料更新的目的是在最短時間內將更改的資料內容傳播於每一個線上的客戶，一個最簡單的機制是每一個客戶執行資料庫內容變更時都立即通知系統，系統再通知所有客戶更新其使用中的資料。但實際上各使用者都只會使用小部分的資料，此部分資料恰好同時被其他使用者更改的機率極低，為了提升效率減少不必要的資料更新與網路通訊，如果資料更改的部分與該使用者目前操作頁面無關，則應該不必執行更新的動作。欲達此目的，資料更新的訊息必須更為詳細，以供系統判斷是否需要更新。

因此本系統在某客戶更改資料後，除了資料庫的寫入之外，傳出到程式系統的資訊還包括：資料被更改的時間、更改的欄位以及該欄資料的內容。在 ASP.NET 的程式設計架構之中，可以使用 Application 公用變數承載這些資訊。換言之，使用者在某個客戶端輸入某預約者姓名之後，除了資料將寫入資料庫，同時也將此更改訊息及內容以公用變數的方式讓其他用戶知道某筆資料已於某時刻被修改。其他的客戶端程式會比對此項訊息與自己目前使用中的資料，如果必須更新則執行，否則僅改變自己的版本時間表示已經處理過此項訊息即可，簡要流程如圖 5。

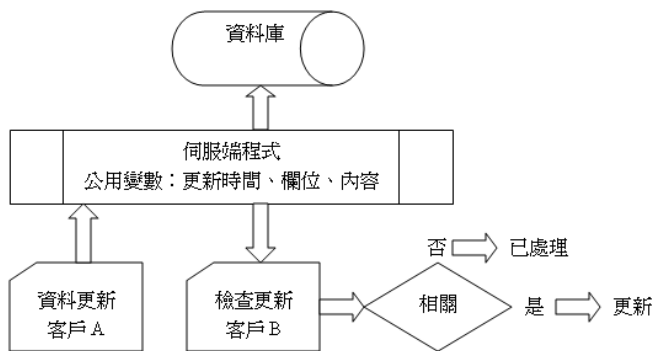


圖 5. 利用網站公用變數通知的更新機制

IV. 客戶端網頁更新之程式設計

如上所述，若系統判斷客戶網頁必須更新時，可以使用 ASP.NET 提供之 AJAX 工具之更新面板(UpdatePanel)來定義必須更新的範圍，這個工具可以精確且不產生整頁回傳的更新網頁上的局部資料。但是在實務上，本系統操作的主要頁面上有多達數百個資料填寫欄位，每一次我們必須更新的卻只有其中一個欄位，如果要使程式可以精確指定更新任何一個欄位，就必須讓每個欄位都套用一個更新面板，原始程式碼將因此龐大到難已處理的地步，在程式編輯階段即可能遭遇到記憶體用罄的情況。

本系統對此的解決方式是不在伺服端的程式中執行網頁更新的動作，而是將必須更新的欄位與新內容寫入網頁的一個隱藏欄位(Hidden Field)之中，並自製一個 JavaScript 程式於網頁之中，定時檢查此隱藏欄位，如有更新資料即以 JavaScript 透過文件物件模式(Document Object Modal,

DOM)[7]取得指定網頁物件的控制，在客戶端直接執行網頁部分更新，此機制以技術內容上來說仍屬於 AJAX 的範疇，但避開了 ASP.NET AJAX 預設處理方式與本系統需求之間的一個實務操作上的矛盾。

精確的說，本系統的局部更新僅限於剛剛被其他客戶更改的某個欄位，亦即某一個病患的名字。這表示程式的執行過程並未觸及網頁中其他不須變更欄位的記憶體對應資料，在使用者的感受中，其本身操作中的欄位也是處於靜止的狀態，所以資料輸入動作不受影響。這一點也是可否稱為 RIA 經驗的關鍵之一，因為一般網頁更新時通常會導致畫面閃爍，輸入動作也必將隨之中斷，但桌上型系統則不會有此現象。

V. 字元等級的資料即時同步

依照一般網路使用者的認知，輸入資料在未按下網頁上的傳送鍵或鍵盤輸入鍵(Enter)之前，資料都還未送出；相對的，在傳統桌上型程式的空白欄位輸入資料則多半是所見即所得，每一個字的輸入其實都立即與資料庫同步。這對於網際網路系統的 RIA 程度亦是一個重要的關鍵，因為很明顯的我們在網頁上必須多做一個確認動作，更可怕的是如果遺漏了這個動作就會使整個輸入工作完全無效！

事實上，在網際網路的伺服器端技術開始實現之後，多數的網際網路程式工具是可以實現在客戶端輸入任一字元就立即傳送伺服器，達到伺服器與客戶端資料的字元等級的同步。但是諸如 ASP.NET 提供的 TextBox 等控制項都還是限制使用者必須按下輸入鍵之後才能觸發 TextBox_TextChanged 事件，程式師也才能在此事件中撰寫處理程序，包括寫入資料庫等等動作。推測其原因應該是考慮到網路傳輸量的控制，過於頻繁的客戶與伺服器互動極容易造成無謂的網路擁塞，尤其是未使用 AJAX 技術的網頁，每一次的互動就代表數十至數百 K 的網頁傳送。

但以本系統的實務考量而言，雖是網路架構，但實際上僅為診所內部封閉的區域網路系統，網路頻寬絕對充裕，更重要的是櫃檯操作人員一般而言非常忙碌，若因為漏按一個輸入鍵而造成輸入未完成確實令人扼腕，因此應可善用頻寬，使系統輸入資料達到字元等級的同步，亦即每一個字元的輸入都會即時的反應到資料庫甚至其他使用者的頁面。

本系統目前即利用 ASP.NET 中的 AJAX 元件之一的 Timer 計時器達到強迫驅動 TextChanged 事件的效果。經實驗證實 Timer 可以啟動 TextChanged 的檢查，只要文字有任何改變，即使不包含輸入鍵，事件程式碼也會被驅動執行。因此本系統的資料庫寫入並非在完整的姓名輸入之後，而是每一個字元完成輸入即有一次的寫入動作。當然其餘客戶端也會即時的看到輸入個別字元的過程。使得系統的即時性提升到字元同步的等級，當然也因此不再需要按下輸入或確定按鍵，使用者所見即所得，RIA 的方便程度當然也更為提升。

VI. 結論

本文以一個繁忙診所的預約系統製作為例，在網際網路程式之架構上使用 AJAX 相關技術，實作出可以令所有用戶端與伺服器端資料即時同步的程式系統。其特色包括：

- A. 資料即時同步，達到逐字輸入逐字同步顯示的地步。
- B. 被更新頁面中的使用者資料輸入動作不會受到影響。
- C. 無須以鍵盤輸入鍵或其他確定按鈕確定上傳資料。

參考文獻

- [1] J. J. Garrett, Ajax: A New Approach to Web Applications, <http://www.adaptivepath.com/publications/essays/archives/000385.php>
- [2] P. Cavaleri, "The use of AJAX in searching a bibliographic database: A case study of the Italian Biblioteche oggi database," Program: electronic library and information systems, vol.42, no.3, pp.275-285, 2008
- [3] <http://msdn.microsoft.com/zh-tw/library/4w3ex9c2.aspx>
- [4] http://en.wikipedia.org/wiki/Push_technology
- [5] <http://msdn.microsoft.com/en-us/library/ms178594.aspx>
- [6] <http://msdn.microsoft.com/en-us/library/ms178581.aspx>
- [7] <http://www.w3.org/DOM/>