

運用 AJAX 技術建構動態 Web 應用於電子化校園資訊服務 -以「選課清單管理」為例

晁瑞明* 林毅力** 呂理賢** 邱顯翔**
大葉大學資訊管理研究所助理教授*、研究生**
ruey밍@mail.dyu.edu.tw

摘要

Web 應用在網際網路之中已朝向多元化發展，對於 Web 的傳統應用始終存在著瀏覽緩慢和互動限制的問題。針對此類問題，倘若使用 AJAX (Asynchronous JavaScript And XML)技術之非同步處理特性可讓網頁應用無需重新讀取整個頁面，以減低伺服器負擔。然而，近年來各校園行政處理上漸轉為電子化資訊服務以增加效率，但與學生事務有關資訊服務，卻常因短時間內大量使用者造成伺服器負擔過大無法負荷。故本研究設法以 AJAX 技術應用在電子化校園選課清單管理系統中，將非需要同步性處理的資料在客戶端完成，在將需同步性處理的資料向伺服器請求傳遞部份資料之方式，減輕伺服器讀取負擔，提升使用者使用上的效率。本研究以 AJAX 技術為基礎建置「校園選課清單管理平台」於電子化校園環境，經由使用者對於選課資訊選擇，透過 AJAX 技術向伺服器請求部份資訊，以 XML 資料傳輸，讀取部份資料即可完成使用者選課資訊需求。「校園選課清單管理平台」成功解決以往在電子化校園中，大量使用者使用校園選課系統上，在同時間內使用造成伺服器嚴重負荷產生緩慢甚至當機之情形，也讓使用者在讀取資訊不會延遲等待或無法讀取。透過 AJAX 技術應用推廣至其他校園資訊服務更能達到豐富網路應用(RIA, Rich Internet Application)及便利。

關鍵詞：AJAX、XML、RIA、e-campus

Abstract

When the web applications have developed toward diversification in the World-Wide Web, the normal usages of the web still have slowly browsing and interactive restriction. About those problems, the AJAX technology can be used for asynchronous processing to make homepage application don't need whole page reloading and to reduce burden of server. Nevertheless, campus administration efficiency has gradually increased because of the involving of electronic information services. On the other hand, the student affairs must deal with in shortly but bundled users to cause server overloading. This research tried to use AJAX technology applied on the information management systems of courses registration to process on client server for unnecessary synchronous processing data, so contrariwise is.

Base on AJAX technology, we created "Campus

Courses Registration Management Platform-CCRMP" for e-campus, with users courses registration, the AJAX will request a part of information access via XML transmit to read. The CCRMP could solve the above mentioned problems successfully to make browser without waiting and avoid systems idling. Hopefully, the AJAX applied applications can be promoted to the other campus services to reach the "Rich Internet Application" conveniently.

Keyword: AJAX, XML, RIA, e-campus

1. 前言

網際網路漸趨成熟，一般 Web 應用有瀏覽緩慢和互動限制的問題存在，尤其與典型的桌面應用程式相比時[13]。而以豐富應用為特色的使用者介面回應與互動功能愈趨容易使用及多元化呈現，讓使用者在操作上不會遭遇困難而變得愈來愈受歡迎，進而讓使用者更樂意在網路上進行互動，近來 Web 2.0 的特色便是讓使用者參與及體驗網路活動。

Web 2.0 是一個新名詞，係由 O'Reilly Media 的 Dale Dougherty 與 MediaLive 的 Craig Cline 在共同合作的 Brain storming 會議上所提出，Web 2.0 最核心的觀念就是「Web 作為平台」。與傳統作業系統平台不同的是，任何人不擁有這個平台，Web 平台是通過分享或參與體系搭建起來的，是對「集體智慧」的充分理解和利用，而不再是大企業的資源壟斷[14]。Tangos 列出 Web 2.0 四個核心特性：可重用的微內容(Microcontent)、以人為中心、社會性、用戶參與的參與架構。這是一種觀念升級非新的技術發展。

傳統 Web 應用中請求(Request)及回應(Response)模式一直被使用，透過使用者觸發事件，Server 接收處理動作再回傳「整個頁面」，當網路負荷流量較大且頻寬限制時，使用者送出動作處理會有延遲等待時間，造成頁面短時間內無法動作。若網頁只有小部份資料改變，瀏覽器需重新下載整個頁面來重新顯示，無形浪費了網路頻寬及增加伺服器負擔[3]。為了能有效降低此情形發生的頻率，除了提升網路頻寬，優化網路配置外，AJAX(Asynchronous JavaScript and XML)技術也是一解決方式，其透過 JavaScript 與 XML 主要技術結合，把部份動作處理轉移到客戶端完成，只需更新部份頁面，即可完成使用者需求。AJAX 是實現 Web 2.0 的一種關鍵技術，能讓伺服器減輕負擔，提高 Web 應用的執行效率，因此讓更多使用者參與及分

享，以更符合 Web 2.0 精神。

校園中由於學生人數眾多，因此許多校園服務皆改為電子化以提升服務效率，但是校園所提供的電子化服務往往也因為人數的關係而仍然無法達到讓人滿意的程度，選課、信箱的功能常由於短時間內的大量使用者造成伺服器負擔過大，無法負荷，增加使用者在使用上的不便。因此本研究將試圖以 AJAX 技術應用在電子化校園選課清單管理系統中，把不需要同步性的資料在客戶端完成，需要同步性的資料再向伺服器請求傳遞部份資料的方式，減輕伺服器讀取負擔，提升使用者使用上的效率。

1.1 AJAX 定義

開發者追溯到使用 AJAX (Asynchronous JavaScript And XML) 建立 Web 應用時，主要技術在 1990 年代開始被發展。一個 AJAX 應用的關鍵優勢是能像桌面應用程式般的方式[13]。首次出現在 Adaptive Path 顧問公司的 Jesse James Garrett 在 2005 年於網路上發表「A New Approach to Web Application」中[4]。這位顧問提出一個新的觀點，認為採用 JavaScript 與 XML 等已存在的技術，即可解決網頁設計時，靜態與動態呈現方式難以抉擇的問題，而不是一味地追逐新的資訊技術，並且使得 Web Application 的操作能像桌上型電腦中的應用程式般，達到 RIA (Rich Internet Application) 的同樣目的。試想，如果我們能使一個網頁的互動方式可以達到類似 Microsoft Office 的境界，那麼網頁的可用性可以達到另一個層次。

在 Jesse Garrett 的文章中，對 AJAX 的定義如下：

- (1) 使用 XHTML 與 CSS 作為展現標準
- (2) 使用 DOM 作為動態顯示與互動
- (3) 使用 XML 與 XSLT 作為資料交換與運用
- (4) 使用 XMLHttpRequest 作為非同步的資料回饋
- (5) 使用 JavaScript 結合以上所有結果

1.2 組成技術、物件方法與屬性

AJAX 也不是一項單獨的技術，它是由一堆現有的技術所組成，它的組成技術有：

- (1) XHTML(或者 HTML)加上 CSS 來作為資料的呈現。
- (2) 利用 DOM 以及 JavaScript 來進行存取資料(大多為 XML)的處理。
- (3) 利用 XMLHttpRequest 物件與遠端的 Web server 進行非同步的資料交換。

而 AJAX 核心技術為 XMLHttpRequest，其物件方法如表 1 所示[17]。

表 1 標準 XMLHttpRequest 操作

方法	描述
abort()	停止當前請求。
getAllResponseHeaders()	把HTTP請求的所有回應標頭作為鍵/值對傳回。
getResponseHeader("header")	傳回指定標頭的字串值。
open("method", "url")	建立對伺服器的呼叫。method參數可以是 GET、POST 或 PUT。url 參數可以是相對URL或絕對URL。這個方法還包括3個可選的參數。
send(content)	向伺服器發送請求。
setRequestHeader("header", "value")	把指定標頭設定為所提供的值。在設定任何標頭之前必須先呼叫open()。

除了這些標準方法，XMLHttpRequest 物件還提供了許多屬性，如表 2 所示。

表 2 標準 XMLHttpRequest 屬性

屬性	描述
onreadystatechange	每個狀態改變時都會觸發這個事件處理器，通常會呼叫一個JavaScript函式。
readyState	請求的狀態。有5個可取值：0 = 未初始化，1 = 正在載入，2 = 已載入，3 = 互動中，4 = 完成。
responseText	伺服器的回應，表示為一個字串。
responseXML	伺服器的回應，表示為 XML。這個物件可以剖析為一個DOM物件。
status	伺服器的HTTP狀態碼(200對應OK，404對應NotFound(未找到)，等等)。
statusText	HTTP狀態碼的相應文字(OK或Not Found(未找到)等等)。

1.3 特性

採用 AJAX，客戶端瀏覽或使用網頁時，如同在個人電腦中的應用程式一樣，不會感覺到網頁與後端伺服器動作請求過程，以及資料更新的細節，頁面也不需要換頁，資料的交換等動態動作隱藏在背景執行。簡單地說，AJAX 網頁有兩種主要特性：向伺服器發出請求以及回傳結果時，瀏覽器不需要重新載入所有頁面；藉由 XML 解析傳遞與交換的資料。AJAX 的特點在於非同步處理資料，動態更

新 Web 頁面，因此它的適用範圍是客戶端與伺服器之間互動較多，頻繁讀取資料的 Web 應用。

優點有兩項：一、不需重覆更新頁面，即可完成非同步的資料交換。二、提升用戶參與網路活動動機。AJAX 是實現豐富的網路應用(RIA, Rich Internet Application)最好的技術之一。而以下為 AJAX 核心的 XMLHttpRequest 物件互動模式，如圖 1 所示。

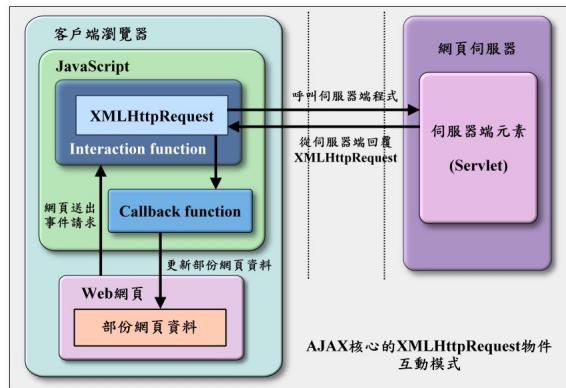


圖 1 AJAX 核心的 XMLHttpRequest 物件互動模式

以下為四個 AJAX 的應用場景[6]：

(1) 資料驗證-由 XMLHttpRequest 物件發出驗證請求

由 XMLHttpRequest 物件發出驗證請求，依據返回的 HTTP 回應判斷驗證是否成功，整個驗證過程不需彈出新視窗，也不需將整個頁面提交到伺服器，速度加快也會減低伺服器負擔。

(2) 依需要向伺服器取得數據

分類樹或者樹狀結構在 Web 應用架構系統中使用得非常廣泛。傳統對分類樹的操作都會引起頁面重新載入，為了避免這種情況出現，一般不採用每次重載頁面的方式，而是將分類結果中的資料一次讀取出來並寫入陣列，然後根據使用者的操作，用 JavaScript 來控制節點的呈現，這樣雖然解決了操作回應速度，不重複載入頁面以及避免向伺服器頻繁發送請求的問題，但是如果使用者不對分類進行操作或者只對分類樹中的一部分資料進行操作的話，那麼讀取的資料中就會有大部份沒有使用，浪費了使用者的資源。特別是在分類結構複雜，資料龐大的情況下，這種弊端就更加明顯了。

AJAX 為改進分類樹讀取的實現機制。在初始化頁面時，只獲取第一層子分類的資料並且顯示；當使用者點開第一層分類的第一節點時，頁面會通過 AJAX 向伺服器請求當前分類所屬的第二層子分類的所有資料；如果再請求已經呈現的第二層分類的某一節點時，再次向伺服器請求當前分類所屬的第三層子分類的所有資料，以此類推。頁面會根據使用者的操作向伺服器請求它所需要的資料，這樣就不會造成資料的浪費，減少資料下載的總量。同時，更新頁面時不需要重新載入所有內容，只更新

需要的那部分內容即可，相對於以前後台處理並且重新載入的方式，有效縮短了使用者的等待時間。

(3) 自動更新頁面

在 Web 應用中有很多資料的變化是十分迅速的，使用者為了即時瞭解相應的內容必須不斷更新頁面，查看是否有新的內容，或者頁面本身設定定時更新。有可能會發生有一段時間網頁的內容沒有發生任何變化，但是使用者並不知道，仍然不斷的更新頁面；或者使用者失去了耐心，放棄了更新頁面，卻很有可能在此有新的消息出現，這樣就錯過了第一時間得到消息的機會。

應用 AJAX 可以改善這種情況，頁面載入以後會通過 AJAX 引擎在後台進行定時的輪詢，向伺服器發送請求，查看是否有最新的消息。如果有則只將新的資料下載並且在頁面上進行動態的更新，通過一定的 JavaScript 方式通知使用者。這樣即避免了使用者不斷手動更新頁面的不便，也不會因為重複更新頁面造成資源浪費。

(4) e-mail 系統

在一般 e-mail 頁面中，對於瀏覽、移動或處理每封郵件，都需向伺服器請求動作來完成，相對以 AJAX 技術來說，只需對郵件部份做處理，不需向伺服器請求動作，即可完成使用者請求，達到提高互動性與操作性。

1.4 應用程式代表性例子

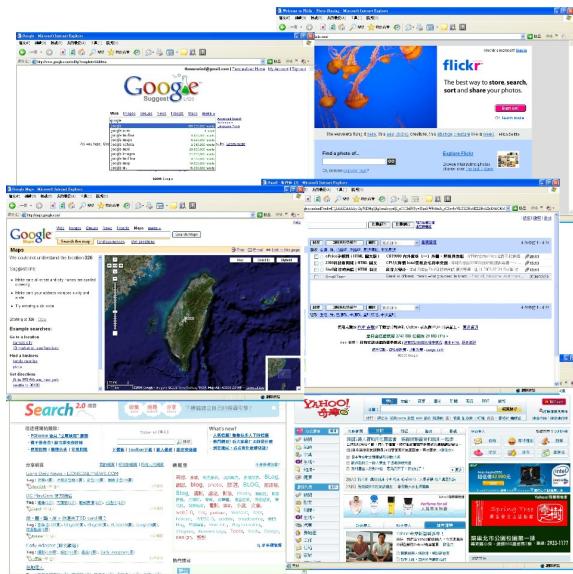


圖 2 AJAX 技術應用網站例子

如圖 2 所示，Google Map 將 AJAX 技術導入到 GIS 系統，它提供在有短時間內能提供使用者所需地圖。但在讀取整個地圖資料是需要非常龐大的設備及資源，為了能有效減低資源與時間，運用 AJAX 技術特性，以 XML 傳輸資料及非同步性地圖資料讀取，透過使用者在瀏覽地圖中的使用動作行為，向伺服器僅需讀取需要地圖部份，即可完成使用者需求，不僅可減低伺服器負擔也能加快使用者瀏覽

效率。

另一例子是 Google Suggest，透過使用者輸入關鍵字中，Google Suggest 使用 AJAX 技術將類似關鍵字做搜尋動作，利用下拉式清單回傳可能性結果數，當關鍵字有改變，將會不斷搜尋及回傳，讓使用者在尋找想要的資料或需求時能更容易。其他如 Yahoo!首頁(2006 年 7 月)、PCHome Search 2.0(2006 年)、Flickr 與 Gmail 等皆有使用 Ajax 技術的網站例子。

2. AJAX 運作架構

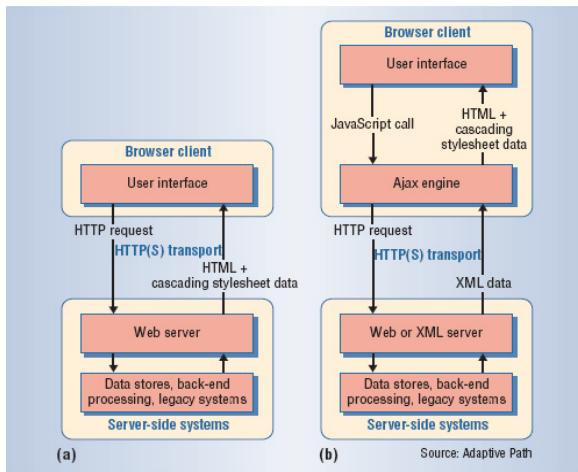


圖 3 傳統 Web 應用模式與 AJAX 模式比較

如圖 3，以(a)圖所示在傳統 Web 應用中，使用者使用動作行為觸發 Http request 向 Web 伺服器請求，伺服器根據請求傳回 HTML 頁面給瀏覽客戶端。當使用者在使用者介面不斷有新的請求動作，會向伺服器請求與伺服器回傳頁面。以(b)圖所示 AJAX 技術應用以 JavaScript 為基礎在使用者瀏覽客戶端建立 AJAX Engine 與執行。AJAX Engine 預先攔截使用者輸入動作請求，顯示將請求資料與處理完成使用者需求。如果 AJAX Engine 需要更過的資料，它在背景執行像伺服器請求資料，在這過程中仍然繼續與使用者保持回應。

在傳統 Web 應用模型中，使用者向 Web 伺服器請求資料，伺服器根據請求傳回 HTML 頁面，這技術並不完全提供給使用者參與網路活動，因為它限制網頁互動性與要求每次網頁重新讀取資料。AJAX 技術在使用者瀏覽客戶端建立 AJAX Engine 與執行，替代傳統 Web 應用，透過 AJAX Engine 顯示需求資料，AJAX Engine 在使用者輸入動作請求前，處理完成使用者需求，如簡單的資料確認。如果 AJAX Engine 需要更過的資料，它在背景執行像伺服器請求資料，在這過程中仍然繼續與使用者保持回應。如果 AJAX Engine 需要更過的資料，它在背景執行像伺服器請求資料，在這過程中仍然繼續與使用者保持回應。AJAX Engine 讓使用者不用與 Web 伺服器直接請求動作，降低伺服器回應等待時間。

3. 系統規劃與實作

本章內容分別就系統流程、研究平台，敘述本研究的設計與建置。

3.1 系統流程

電子化校園中，透過校園資訊網路處理眾多文件，其中以校園選課對於學生應用廣泛，對於選課清單管理在校園選課系統上，學生瀏覽量加重伺服器負擔情形，本研究將以 JSP 為程式語言基礎使用 AJAX 技術的非同步處理特性與 XML 資料傳輸，解決以往電子化校園中，大量使用者使用校園選課系統上，在同時間內使用造成伺服器負荷嚴重造成緩慢甚至當機情形。傳統電子化校園資訊服務選課清單管理與 AJAX 技術應用於選課清單管理比較表如表 3 所示。

表 3 大量使用者使用電子化校園選課清單管理傳統與使用 AJAX 技術比較表

	傳統電子化校園資訊服務 選課清單管理	AJAX 技術應用於 選課清單管理
讀取資料	整個頁面更新 讀取緩慢	部份頁面更新 讀取所需
傳輸	字串、陣列物件傳輸	以 XML 格式傳輸
使用者瀏覽器	讀取緩慢延遲 造成畫面空白 或延遲	讀取不延遲 畫面動態更新
整體效率	較慢	較快

使用 AJAX 技術在 Web 應用於電子化校園資訊服務選課清單管理，本研究擬定以下流程圖：

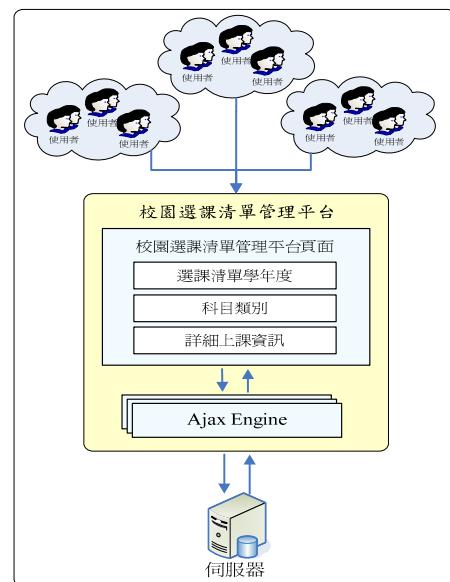


圖 4 系統流程圖

系統流程圖如圖 4 所示，當大量使用者使用「校園選課清單管理平台」，在校園選課清單管理平台頁面中，透過選課清單學年度、科目類別、上課資訊使用者選擇方式，不斷向伺服器請求傳遞需求瀏覽資料，透過 AJAX Engine 為媒介，把非需同步性資料在客戶端完成，及需同步性資料再向伺服器請求傳遞部份資料的呈現。

3.2 研究平台

本研究建置「校園選課清單管理平台」，藉由 JSP 程式語言執行速度快的特性與 AJAX 技術的非同步性資料處理特性運用在校園選課清單管理上。系統可分為四個部份：一、執行頁面；二、服務端處理請求的 Servlet 部份程式碼；三、產生的 XML 文件檔；四、客戶端 JavaScript 程式碼。

3.2.1 執行頁面

學年	第	學期	科目類別	英文類					
年	班	科目名稱	科目代號	科自序號	餘額	必選修	學制名稱	教師	開課人數
1	3	英語口語表達5	CDC1101	3654	32	必修	大學日間部	石慧華	50
1	3	英語口語表達3	CDC1102	3651	49	必修	大學日間部	石慧華	50
1	3	英語口語表達1	CDC1103	3652	44	必修	大學日間部	王方	50
1	3	閱讀技巧E1	CDC1104	3656	41	必修	大學日間部	王方	50
1	3	閱讀技巧E2	CDC1105	3657	15	必修	大學日間部	王方	50
1	3	閱讀技巧E3	CDC1106	3658	0	必修	大學日間部	張天民	50
1	3	英文作品欣賞F1	CDC1107	3659	7	必修	大學日間部	俞婷	50
1	3	英文作品欣賞F2	CDC1108	3660	22	必修	大學日間部	俞婷	50
1	3	進階英文法2	CDC1109	3661	28	必修	大學日間部	俞婷	50

圖 5 選課清單管理運行結果

「校園選課清單管理平台」執行結果如圖 5 所示，本研究透過兩個部份讓使用者選擇選課清單，根據使用者的需要傳達請求，使用 AJAX Engine 向伺服器讀取資料。資料可依使用者需求讀取部份資訊，使伺服器減少讀取負擔，也能讓使用者讀取資料更為快速。第一部份，使用者從學年度、學期與科目類別中選擇課程選項，再經由結果選擇第二部份各科目的相關資訊，即可在選課清單獲取完整的選課及上課資訊。

3.2.2 伺服器端處理請求的 Servlet 部份程式碼

如本研究前面所述，使用 AJAX 技術的主要目的在於從 Server 端非同步得到訊息，避免持續更新整個選課清單的頁面。伺服器端處理請求的 Servlet 部份程式碼如圖 6 所示，運作說明如下：

(1) 為了使用 JavaScript 對伺服器發送 HTTP 要求，必須先以相關的類別(class)製作出實體(instance)。有些版本的瀏覽器在伺服器送回的資料中未含 XML mime-type 檔頭(header)時，會產生錯誤。為了避免這個問題，使用 response.setContentType(“text/xml”) 的方法覆寫

伺服器傳回的檔頭，以免傳回的不是 text/xml。

(2) 接著是要決定伺服器傳回資料後的處理方式，此時只要以 `onreadystatechange` 這個屬性指明要處理傳回值的 JavaScript 函式名稱即可，隨後透過 `HTTP request` 類別的 `open()` 方法來發送非同步回應的請求。

獲取頁面參數之後，試圖建立與資料庫的連接，之後將資料庫記錄內容轉為 XML 的元素，最後利用 HTTP 通訊協定中 GET 的存取方式，產生課程有關的選課訊息來傳送資料。

```

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    response.setContentType("text/xml; charset=UTF-8"); //建立文件檔類型
    PrintWriter out=response.getWriter();
    String sql=request.getParameter("eselectid"); //獲取頁面參數
    String sql="select students.state,subject.name,students.subjectnum,subject.subjectnumber";
    Sql=sql+" from students,subject where subject.id=students.pid";
    Sql=" "+Sql+" and students.subjects='"+sid+"'"; //設置查詢字符串 sql
    DBConn=new DBConn(); //建立與資料庫的連接
    Document document = new Document (new Element ("eselectxml"));
    Element eeselect=new Element ("eselect");
    Document.getRootElement ().addContent (eeselect); //加入表單元素
    Try{ResultSet rs = eiselect.executeQuery (sql); while (rs.next ()) { //將資料庫記錄內容轉為XML元素
        eiselect.setAttribute("id", sid);
        eiselect.setAttribute("state", rs.getString (?state));
        Element element=new Element("subjects");
        Element1.setAttribute("name", rs.getString("name"));
        Element1.setAttribute("num", rs.getText(rs.getString("subjectnum")));
        Element e2=new Element("number").setText(rs.getString("subjectnumber"));
        Element1.addContent(e2);
        Element1.addContent(e3);
        eiselect.addContent (element1);
    }
}
catch(SQLException e) { System.err.println(e.getMessage()); }
.....//生出錯誤有關的選錯訊息
XMLOutputter output=new XMLOutputter ();
Output.output(document,out); //輸出XML
Out.close ();
}

```

圖 6 伺服器端處理請求的 Servlet 部份程式碼

3.2.3 產生的 XML 文件檔

以下為 responseXML 屬性的方法。首先，我們需要用格式正確的 XML 文件，以便之後取用。此檔名為 eselectxml，內容如圖 7 所示。

```
<? xml version="1.0" encoding="UTF-8" ?>
<eselectxml>
<eselect id = "B5CDC11013654" state="3">
<class name="英語口語表達B5">
<num>1</num>
<number> </number>
</ subjects >
<client>
.....選課清單訊息
</client>
</eselect>
</eselectxml>
```

圖 7 產生的 XML 文件檔

這樣一來我們便可取得 responseXML 所傳回的 XMLDocument 物件，之後以 DOM(Document Object Model)相關的方法取用 XML 文件內容。

3.2.4 客戶端 JavaScript 程式碼

```

function Openeselect (id){
    var url="eselectServlet?eselectid="+id;//請求URL
    .....//建立XMLHttpRequest對象req
    if(req){
        document.getElementById("title").innerHTML="正在讀取選課清單....";
        req.open("GET", url, true);
        req.onreadystatechange=complete; //指定處理伺服器回應的函數
        req.send(null);
    }
    function complete(){
        if(req.readyState==4){
            if(req.status==200){ //回應正常返回
                var eselect=req.responseXML.getElementsByTagName("eselect");
                var subjects=req.responseXML.getElementsByTagName("subjects");
                document.getElementById("title").innerHTML="選課清單:"+eselect[0].getAttribute("id");
                var myHolder=document.getElementById("insertdiv");
                //獲得頁面中ID="insertdiv"的DIV元素的引用。
                var myTable=document.createElement("<table width='100%' border=0>");
                .....
                myHolder.appendChild(myTable);
                //將表格加入頁面
            }
        }
    }
}

```

圖 8 客戶端 JavaScript 程式碼

如圖 8 所示，在客戶端 JavaScript 程式的部份，傳出 request 時必須提供處理傳回值的函式名稱。

在此為 Openeselect，此一函式首先檢查 request 目前的狀態，如本研究的程式碼狀態值(readyState)為 4 代表伺服器已經傳回所有資訊，便可以開始解析所得到的選課清單資訊。

接著要檢查 Server 傳回的 HTTP 狀態碼。而在客戶端 JavaScript 程式碼此區塊中，req.status 為 200 狀態碼時，表示回應正常返回。檢查傳回的 HTTP 狀態碼後，決定如何處理傳回的資料。而本研究所採取的方式為 http_request.responseXML，這樣會把傳回值視為 XMLDocument 物件，而後可用 JavaScript DOM 相關函式處理，最後即可產生選課清單管理的頁面。

4. 結論

本研究以 AJAX 技術實作設計平台建置「校園選課清單管理平台」，以 JSP 程式語言為基礎，使用 AJAX 技術的非同步性資料處理特性與 XML 資料傳輸運用在校園選課清單管理上。透過使用者從學年度與學期、科目類別、各科目的相關資訊的選擇，只需讀取部份頁面不需整個頁面。

針對「校園選課清單管理平台」成功應用 AJAX 技術在電子化校園於校園選課清單管理上，解決以往電子化校園中，大量使用者使用校園選課系統上，在同時間內使用造成伺服器負荷嚴重造成緩慢甚至當機情形，也讓使用者在讀取資訊不會延遲等待或無法讀取。未來可推廣至校園資訊系統，透過 AJAX 技術解決傳統 Web 應用限制，增加校園行政效率，增廣豐富多元及便利性，減緩伺服器負擔，讓教職員工及學生使用者瀏覽及使用無延遲。

參考文獻

[1]沈時宇。2006。「AJAX 網頁程式設計－Google

成功背後的技術」。旗標出版股份有限公司。

[2]高橋登史朗。2006。「Ajax 與 Google Map API 入門實作」。博碩文化股份有限公司。

[3]趙曉峰。2006。「利用 AJAX 構建動態 WEB 應用」。深圳信息聯業技術學院學報。

[4]adaptive path, 2005, 網址：

<http://adaptivepath.com/publications/essays/archives/000385.php>

[5]Ahmet Sayar, Marlon Pierce and Geoffrey Fox, "Integrating AJAX Approach into GIS Visualization Web Services", Telecommunications, 2006. AICT-ICIW '06. International Conference on Internet and Web Applications and Services/Advanced International Conference on, September 2006.

[6]AJAX 中國，2006，網址：

<http://www.okajax.com/info/net/200605431.html>

[7]Brett McLaughlin, "Head Rush Ajax", O'REILLY, 2006.

[8]David Franklin, Kristian Hammond, "The intelligent classroom: providing competent assistance", Proceedings of the fifth international conference on Autonomous agents, May 2001.

[9]Harry Halpin, Henry S. Thompson, "XML & web services: One document to bind them: combining XML, web services, and the semantic web", Proceedings of the 15th international conference on World Wide Web WWW '06, May 2006.

[10]Helena Galhardas, Daniela Florescu, Dennis Shasha, Eric Simon, "AJAX: an extensible data cleaning tool", ACM SIGMOD Record , Proceedings of the 2000 ACM SIGMOD international conference on Management of data SIGMOD '00, Volume 29 Issue 2, May 2000.

[11]Kazuhiro Fur。2006。「最新 JavaScript 與 Ajax 範例活用辭典」。博碩文化股份有限公司。

[12]Keith Smith, "Simplifying Ajax-Style Web Development", Computer, May 2006.

[13]Linda Dailey Paulson, "Building rich web applications with AJAX", Computer, October 2005.

[14]O'Reilly, 2005，網址：

<http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>

[15]Seinturier, L., Laurent, A., Dumant, B., Gressier-Soudan, E., Horn, F, "A framework for real-time communication based object oriented industrial messaging services", Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA '99. 1999 7th IEEE International Conference on, October 1999.

[16]Win Treese, "Putting it together: Web 2.0: is it really different?", netWorker Volume 10 Issue 2, June 2006.

[17]W3C, 2006，網址：

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.htm>