

# 以 SOA 為基礎之中等學校校務行政協作系統規劃與建置

王金鳳

國立高雄第一科技大學 資管系  
u9324720@ccms.nkfust.edu.tw

曾守正

國立高雄第一科技大學 資管系  
imfrank@ccms.nkfust.edu.tw

## 摘要

近年來以 Service-Oriented Architecture (SOA) 為架構的系統，逐漸成為開發系統的主流。由於 SOA 具有鬆散耦合 (Loosely Coupling) 的特性，因此系統開發者可藉由組合方式來進行系統建置。本研究以 SOA 為基礎、應用 Service-Oriented Computing (SOC) 的概念、採用 XML 文件做為資料交換標準，針對中等學校校務行政協作系統進行規劃與實際建置，以期達到共用服務元件、跨平台、減少個別開發的成本及人力為目的。

關鍵詞：SOA，SOC，校務行政系統

## 1. 緒論

### 1.1 研究背景、動機、目的

中等學校校務行政系統既龐大且複雜，在開放原始碼的概念興起後，由一群熱衷校務行政系統的伙伴共同成立「台灣中等學校資訊管理人學會」並制定資料庫綱要 (Database Schema)，供大家共同遵守，讓各校共同開發系統，再將系統發表 (Release) 給各校。

然而在此急驟變化的資訊環境下，系統仍然很難敏捷地調適。而 SOA 架構概念的興起正可解決此一問題，因此學會發起人之一尤弘志先生提出 DSA (Document Service Architecture) 架構 (如圖 1-1) 來實踐 SOA 的精神。本研究以中等學校校務之作業流程為基礎，進行服務元件之分析，藉以建構出以服務為導向之系統架構，希望藉此探討以 SOA 架構對原先系統進行架構上改良與提升的可行性，以便提升未來的應用彈性。

### 1.2 研究方法

我們採用 Peacock [13] 的分散式資料處理架構為基礎，主要的 Service 由處理邏輯層 (Business Logic) 負責，在此 SOA 的架構中，資料展現層可以更具彈性地將處理邏輯層任意組合、抽換與互用，讓系統以更靈活的方式應付多變的資訊科技環境。



圖 1-1：Document Service Architecture 架構圖 [1]

## 2. 文獻探討

### 2.1 SOA 相關研究

SOA 架構在業界的大力推介下，漸漸蔚為風潮，系統開發人員在此架構下，迎接一種新的概念與方法，以下是有關 SOA 之相關研究。

#### 2.1.1 從分散式元件架構到 SOA

SOA 如同物件導向一般，運用元件組成應用系統，其架構上傾向於自主與異質的架構[8]。SOA 具有下列技術特性[4]：

1. 分散式架構 (Distributed) — SOA 的組成元件是由許多分散在網路上的系統組合而來。
2. 鬆散耦合的界面 (Loosely-Coupled) — SOA 實質上就是一套鬆散耦合的服務，以介面標準來組合系統，只要符合介面要求，零組件可以任意替換，大幅提高系統變更的彈性度。
3. 以開放標準為依據 (Open Standard) — 使用開放標準 (XML) 是 SOA 的核心特色，SOA 著重於標準與互用性，可避免不同平台所開發程式間相互整合的困擾。
4. 以流程角度出發 (Process-Centric) — SOA 的服務能夠完成對應其商業流程活動的各部分工作。在 SOA 中，開發人員撰寫服務需要考慮周詳、面面俱到，使得服務可達到最大的重複使用效益。

#### 2.1.2 Service-Oriented Architecture 介紹

SOA 是新一代軟體開發的架構理論，從結構化與物件導向程式開發的特點中擷取兩者的優點，而提供簡單的設計原則[10]。其觀念對於舊有的資訊系統，在更新整合時屬於延伸性的架構，將共同的商業邏輯流程 (Common Business Logic) 製作 Services，以便共同分享[5]，如圖 2-1 所示。

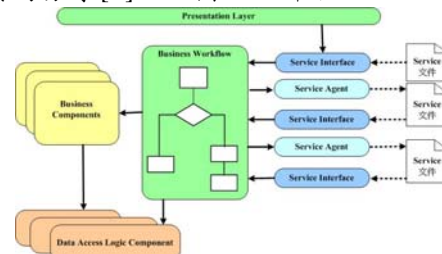


圖 2-1：服務導向於企業應用架構設計 [5]

使用 SOA 的優點如下[8][11][12]：

1. 標準化：標準化可改善發展應用軟體的生產力與提昇品質、降低成本。
2. 以流程為主的設計：將商業流程所產生的

服務組合，讓流程被平衡與再使用，以貫穿整個組織。

3. 協作便利：專業分工，提高人力效能，並可發展系統生命週期的管理工具。
4. 快速反應需求：提供富含語意、彈性的運算模式來發展應用軟體，快速反應使用者需求。
5. 一致性的安全、權限策略：由服務提供者制定相關管理策略供服務需求者遵循，唯有具備權限的使用者方可引用服務，而傳遞資訊的加密方式也由服務提供者制定，並由服務需求者配合。

### 2.1.3 Service-Oriented Computing 簡介

相對於 SOA 來說，SOC 是一種計算概念，它藉由 SOA 來建立模式，達到共用元件、節省程式開發的目的，其概念如圖 2-2 所示。它透過發現 (discovering) 及組合 (composing) 適合的服務滿足系統目標 (圖上半部) 或工作流程 (圖下半部)。

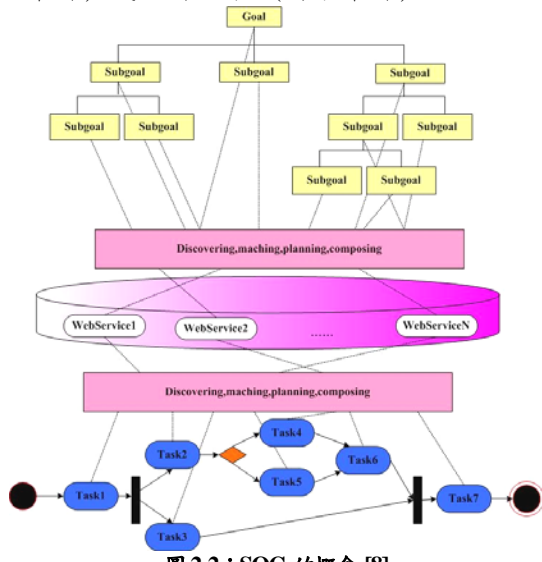


圖 2-2：SOC 的概念 [8]

此外，SOA 與 SOC 兩者之間共同具有下列關鍵要素 [8][14]：

1. 鬆散耦合 (loosely Coupling)。
2. 建置中立：以服務為基礎的計算模式與架構不能侷限於一套程式語言，須配合不同建置者及舊有應用的技術，達到技術中立的目的。
3. 彈性的組態：一個 SOC 系統嵌入的元件是最新的程序，可動態地將需求元件嵌入。
4. 持續性：唯一可提供估計聲譽 (Reputation) 的方法是他們的可靠性，因此服務必須存在一段時間，短暫服務將無法發展聲譽。
5. 粒度 (Granularity)：粒度是衡量抽象化大小的方式。粗糙抽象化在參與者中降低依賴性，並且把通訊降低到更少且更具意義的訊息上 [2]。
6. 團隊 (Teams)：團隊工作的管理方式已經由集中管理到分散管理，因此必須有效地

在開放系統環境下，讓自主性高的合作夥伴能參與團隊工作並解決合作上的問題。

### 2.1.4 Service 的特質

Service 是一個應用程式的包裝，可以重覆使用在商業流程的元件，一般來說 Service 具有以下幾項特點 [8][9]：

1. 互用性 (Interoperability)：各服務元件可以互相使用，節省程式開發時間。
2. 位置透通性 (Location transparency)：所謂 Transparency (透通性) 即當客戶端呼叫伺服器端時，系統應自動幫使用者處理所有尋找位置、安全性與可靠度方面的事務 [12]。
3. 鬆散耦合 (Loosely-coupled)。
4. 定義完整 (Well-defined)：使用獨立於任何技術的共同定義，可被任何技術引用。
5. 無狀態 (Stateless)：服務處於無狀態模式，任何服務無需知道上個步驟的內容，所有的服務都是處於一種獨立的狀態。

服務 (Service) 可以從兩個觀點來理解其概念的差別：

1. RPC View：把 XML 文件視為分散式計算的伴隨物件。
2. 文件 (Document) View：認為 XML 文件為主要分散式計算的表現和目的。

由於在 Web 的文件交換上，文件類型的觀點比較具有非同步性及一致性，因此，Singh and Huhns [14] 也明白指出：未來將以文件 View 較為盛行。本研究所採用的技術正是以 XML 文件為中心的服務導向架構，也符合未來 SOA 的主流格式。

## 2.2 XML

「可擴充式標示語言」(Extensible Markup Language, XML) 是 SOA 的核心，可以解決不同程式語言、資料庫、作業系統平台所引發的資料格式問題。

XML 文件可以被分成 Tree-based, Event-based, 以及 class-based 等介面來操作使用。其中，W3C 提供了 Tree-based 的 API 方法，稱為 DOM (Document Object Model)，而 Event-based 的 API 則以 SAX [7] 為代表。

## 2.3 Web Service

Web Service 的技術標準包含有：XML、UDDI (Universal Description Discovery and Integration)、WSDL (Web Service Description Language)、SOAP (Simple Object Access Protocol)、HTTP [6] 等，如圖 2-3 所示。

## 2.4 SOA、DSA 與 Web Service 的關係

Web Services 在 SOAP 的規範中有兩種格式 [8]，詳述如下 [17]：

1. RPC-Style 的 Web Services：是屬於緊密耦合的方式。客戶端將參數及回傳值的描述置於 WSDL 描述，並將其包裝於 SOAP

內，屬於同步訊息通訊的方式。

- Document-Style 的 Web Service：是鬆散耦合，透過 XML 資料的格式進行請求/回應的動作。客戶端將參數以 XML 文件的型式，送到 Web Service，而不是分離的參數值，屬於非同步訊息通訊方式。

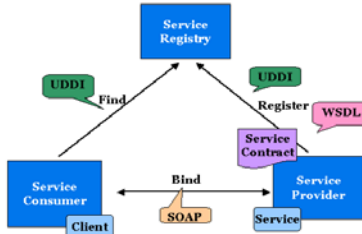


圖 2-3：Web Service 架構圖 [15]

如上所述，若 Web Service 採用 RPC-Style 將不符合 SOA 鬆散耦合的精神。因此，想要以 Web Service 來實踐 SOA 的精神，就應該在 Document-Style[8]的形式上探討。

DSA 的組成元件服務有：需求者 (Requester) 與服務供應者 (Provider)。目前的 DSA 尚未提供註冊的功能，但符合 SOA 的基本精神：鬆散耦合、互用性、建置中立、Well-Defined、無狀態等。

簡單來說，DSA、Web Service 都是 SOA 的實作技術，前者是屬於羽量級的 SOA，而後者則是重量級的 SOA。基於中等學校校務行政協作系統本身的特性，以及考量其建置的容易性、技術的簡易性，本研究因此採用 DSA 技術來實現 SOA 的精神。

## 2.5 安全性

在開放性網路架構下，數位化資料傳輸的安全性，成為不可或缺的重要問題；在毫無安全機制下的網路中，使用者可能面臨資料遭到冒名傳送、竄改、偽造、不法擷取等問題。

建立 Web Services Security 可以採用 SSL (Secure Socket Layer) 的方式，以公用密鑰技術確保兩個應用間的通信保密性和可靠性，並可在伺服器 and 客戶端同時支援 [3]。

## 2.6 軟體協作開發

SOA 的主要成功因素取決於合作夥伴的互助模式。目前的專案系統已不再是一人可以完成的，需要團隊的合作。一個專案管理的組成包含人 (People)、團隊 (Team)，以及組織 (Organization)。團隊合作的績效是複雜且難於管理的，因此如果能有一套適合的架構來協助開發，將可以達到完善的效率。

## 3. 相關技術

### 3.1 DOM

XML 文件物件模型 (DOM) [18] 可透過程式設計介面，存取及更新 XML 的元件，包括其元素、屬性、處理指令、註解及實體

宣告。當 XML 文件被載入 DOM 時，會將該文件從頭到尾讀取，並儲存在 DOM 中，成為節點的邏輯模式。

### 3.2 VBA 巨集語言

Microsoft Visual Basic for Applications (VBA) [16] 主要是為了應用程式自訂化與自動化而設計，它讓程式開發人員可以在不同應用程式使用共同的巨集語言進行開發，並且使不同桌面應用程式得以結合成完整的資訊系統。

### 3.3 DBHelp 語法

在撰寫 Service 時，學會的 DSA 架構中提供了 DBHelp 的語法，其方法是結合 SQL 語法，並增加其彈性，將使用者傳入的參數內容結合原 SQL 語法，完成一個完整的 SQL，接著再執行 SQL。

## 4. 系統建置

### 4.1 系統架構圖

我們依據圖 2-2 的 SOC 概念，將本系統從主目標、子目標階層關係與 Services 間的關聯性，對應到各項 Task 的關係，以及如何透過工具協助找到合適的服務，繪製如圖 4-1 所示。圖 4-1 上方表示整個校務行政系統的各项子系統，以及子系統內功能模組。這些需求可透過 DSA ToolKit 來規劃、設計或合成現有服務來達成，下方表示校務行政系統下某一子功能之工作流程 (WorkFlow)，每一個工作流程可透過 DSA ToolKit 找到合適的服務，完成此一子功能。

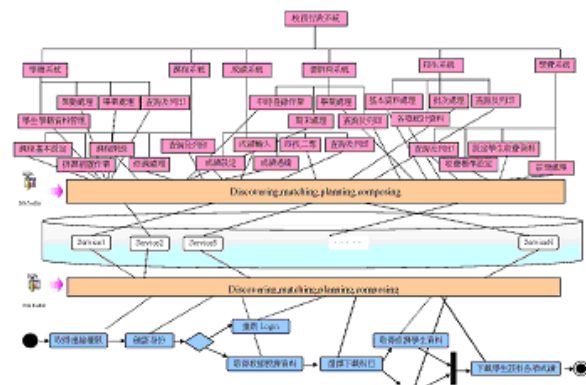


圖 4-1：系統架構圖

### 4.2 系統使用者需求分析

本研究針對舊有校務行政系統所欠缺之功能來加以補強，作為 SOA 的示範設計。我們希望讓這類需求可以跨平台取得資料，其主要使用者為教師以及學校主管，並依據其平時業務需求來分析，提供使用者可在任何時間、地點下載、上傳其業務內之資料，其中與一般教師相關的有下列幾種：

- 上課表現：供教師下載授課課程之學生基本資料，並可登錄上課表現成績，於期末上傳至資料庫，做為學生該科成績之一部



分。

- 成績表：分為二種，一種是綜合高中使用，另一種則為其他科使用。此報表亦提供了統計的功能，可算出各級距分數的學生人數，並有圖表顯示，讓教師在評分時可以更容易掌握學生的分數分佈狀況。

此外，由於導師平時工作繁瑣，我們也提供了與其相關業務的功能，如下列所示：

- 班級課表：讓導師在期初將班級課表下載，列印出來給學生，以便知會學生本學期之課程安排。
- 班級學生基本資料：導師於學期初將學生基本資料下載，可作為平時聯絡學生用，若學生基本資料有更動，亦可直接修改，再上傳即可。
- 個人獎懲明細：導師於每星期或每個月下載班級學生之獎懲明細，供學生參考。
- 缺曠請假清單報表：導師每星期將上星期之出缺勤資料下載列印，提醒學生進行請假手續之辦理。
- 班級個人出席暨獎懲狀況：列出班上學生某日期區間的出席及獎懲統計資料，以掌握每位學生之出席率（以缺席時數除以實際上課總時數）。
- 每月生活通知單：導師於每個月初將上個月學生出缺勤及獎懲資料下載，並填入導師通訊及當月表現評比，然後列印寄出。
- 月考成績單：供導師下載其班級的月考成績，藉由這個報表的統計功能看出班上月考成績的分佈情形，並列印出來供學生核對，檢查是否有輸入錯誤之情形，再列印寄出給家長。
- 班級學期六育成績明細：由於導師必須填寫導師手冊、AB卡之成績資料，因此下載六育成績供導師填寫之用。
- 期末德行成績明細：導師於期末時下載學生該學期之德行表現資料，及結算後之德育成績，最後填入加分、群育及評語等資料後，上傳至資料庫。
- 班級學生學期科目成績統計狀況：導師於期末可以下載學生學期的科目成績，並了解其成績的分佈情形。

最後，我們也依主管所重視的統計資料，提供了以下幾種報表的輸出：

- 班級出席率：主管可以列出各班級日期區間的出席狀況，並知道該班之出席率，其中出席率仍以缺席時數除以實際上課總時數。
- 班級獎懲統計：主管可以列出各班級日期區間的獎懲統計，方便了解班級學生的獎懲狀況，以作為進一步的輔導措施安排。
- 教師授課成績範圍一覽表：主管可以列出各教學研究會之教師其授課課程之成績

的最高分、最低分、平均分數。

上述功能我們也予以分類整理如表 4-1 所示。

表 4-1 系統功能比較表

功能	適用時機			適用對象		
	期初	平時	期末	主管	導師	教師
上課表現	✓	✓	✓			✓
成績表	✓	✓	✓			✓
班級課表	✓				✓	
班級學生基本資料	✓	✓	✓		✓	
個人獎懲明細		✓			✓	
缺曠請假清單報表		✓			✓	
班級個人出席暨獎懲狀況		✓			✓	
月考成績單		✓			✓	
每月生活通知單		✓			✓	
班級學期六育成績明細			✓		✓	
期末德行成績明細			✓		✓	
班級學生學期科目成績統計狀況			✓		✓	
班級出席率		✓		✓		
班級獎懲統計		✓		✓		
教師授課成績範圍一覽表			✓	✓		

### 4.3 系統需求

本系統的入口程式以 .NET Framework 1.1 撰寫，因此其伺服器端的相對環境則必須安裝 .NET Framework 1.1，而入口程式必須置於 Web Server 上，因此必須啟動 IIS (Internet Information Services) 服務，接著再將 DSA 程式安裝至伺服器端，並開啟 Web 共用功能，即完成伺服器端的安裝。

撰寫好的服務則置入 DSA 元件中的 Service 目錄中，即可接受客戶端的 Request，表 4-2 即 Server 端的環境需求表。

表 4-2 Server 環境需求表

- IIS
- .NET framework 1.1
- DSA 基本原件
  - Service(所有 Service 的 Description)
  - Bin(所有 Service 的 DLL)
  - Config(DSA Server 的組態設定)
  - Doorway.config 內設定預設資料來源
  - Doorway.aspx (DSA Server 進入點)
 可跨平台 (利用 Java 選寫 Doorway)

接著，客戶端只須安裝 Office (含 Excel)，並設定完成 Internet 連線，即可執行各項功能。

#### 4.3.1 系統開發架構及伙伴關係

在 DSA 的架構下，使用者是透過 XML 文件傳遞需求，伺服器再依據需求來產生 XML 回應文件，使用者不需知道資料的來源，權限的控管等，這些都是 DSA 的工作範圍，圖 4-2 即是 Server 與 Client 的關係圖。

本系統的核心架構 Doorway、Toolkit、

DSA 基本物件 DSConnection、DSRequest 與 DSResponse 是由「台灣中等學校資訊管理入學會」提供，本研究則是針對中等學校校務行政系統的功能撰寫相關服務，並實際建置 Excel 工作表，系統的協作關係如圖 4-3 所示。

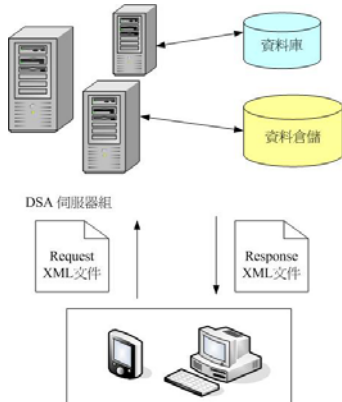


圖 4-2 Server 與 Client 關係圖

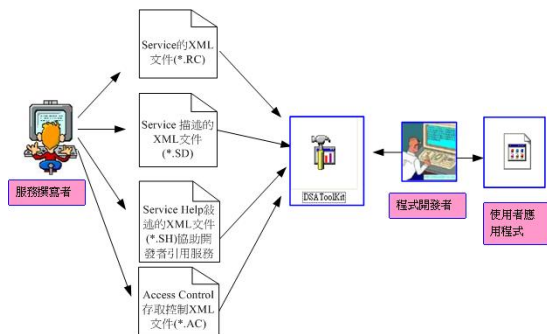


圖 4-3：系統伙伴合作關係圖

#### 4.4 服務撰寫

在撰寫服務時，我們必須了解服務的規格與語法，服務 (Service) 是以目錄為管理單位，每一個目錄下都有服務的資源規劃，例如：引用的資料庫為何？那些人可以使用這些服務？最後，統一將這些資源規劃納入 Default.GRC 及 Default.GAC 檔案中。

為了方便服務可以給更多人使用，因此整個架構也規範了必須撰寫 Help 檔的要求，以描述如何引用該服務、該服務回應何種訊息、何種角色具有存取該服務的權限等，以便讓更多人輕易的使用所有服務。

以下描述我們所撰寫的服務與相關的資源檔案：

1. Default.GRC：描述此目錄中各項相關服務所共同引用之資源。
2. Default.GAC：描述目錄中的存取權限。
3. \*.RC：描述資料相關資源，即 Service。
4. \*.SD：每一個服務都需要一個相對應的描述檔，為 XML 文件格式，附檔名為 .SD。
5. \*.SH：每一個服務都有一個相對引用說明，是一個 XML 文件，附檔名為 .SH。
6. \*.AC：若個別服務的權限控管不同，則於此一檔案內描述，是一個 XML 文件，

附檔名為 .AC。

在撰寫服務時，將性質相同的服務置於同一目錄，可以方便管理，且權限控管也較為一致，而服務本身的彈性、傳回資料的筆數都必須考量。本研究總共使用了 38 個服務來完成 16 個程式。

#### 4.5 引用服務

當程式設計者找到合適的服務後，即可將該服務 Binding 到程式中，在撰寫 Client 端引用服務時，其撰寫步驟為：

1. 與 DSA Server 取得連線並連線成功。
2. 將 Request (XML 格式) 如圖 4-4，送至 DSA Server 端。
3. 取得 DSA Server 送回之 Response (XML 格式)，如圖 4-5。
4. 處理 Response 內容。

```
<?xml version="1.0"?>
<Request ServiceName="TAIS.ExamList/1.0" UserID="uid" Password="xxx">
</Request>
```

圖 4-4：Request 範例

```
<?xml version="1.0"?>
<Response Status="0" Version="1.0">
<考試類別>
<試別>
<試別編號>1</試別編號>
<試別名稱>第一次月考</試別名稱>
</試別>
<試別>
<試別編號>2</試別編號>
<試別名稱>第二次月考</試別名稱>
</試別>
<試別>
<試別編號>3</試別編號>
<試別名稱>期末考</試別名稱>
</試別>
</考試類別>
</Response>
```

圖 4-5：Response 範例

#### 4.6 系統範例

在系統內常用的物件有 DSRequest、DSResponse、DSConnection，其功能詳述如下：

1. DSRequest：在 DSA 架構中，扮演 Request 的角色，這個物件可以幫助使用者快速的建立 Request，以便對 DSA Server 提出 Request。
2. DSResponse：在 DSA 架構中，扮演 Response 的角色，這個物件可以幫助使用者取得 Server 的回應，並轉成 RecordSet。
3. DSConnection：在 DSA 架構中，扮演 Connection 的角色，負責與 DSA Server 取得連線，系統自動透過 Login 資料表取得帳號、密碼的認證。

圖 4-6 至 4-7 為本系統執行教師授課成績範圍一覽表的範例，在系統中亦結合了 Excel 的樞紐分析表，由於功能是主管級，

因此權限在 DSA Server 端的權限設定中，已指定那些角色可以使用該項服務，當沒有權限的使用者使用該服務時，系統即會回應一個 Response，告訴使用者權限不足。

本系統主管可以選擇多個教學研究會，來顯示教師授課成績範圍一覽表，藉由這個報表，主管可以了解各教師給分級距，如圖 4-6 與 4-7 所示。

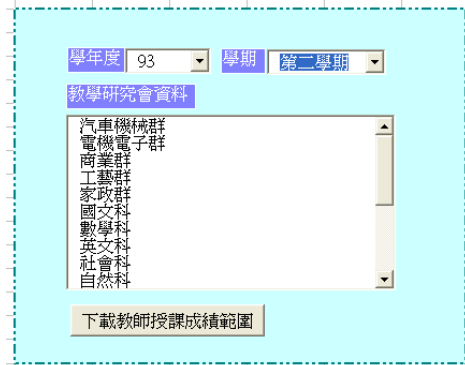


圖 4-6：教師授課成績範圍一覽表下載畫面

科目名稱	教師名稱	課程名稱	成績最高分	成績最低分	成績平均
三民主義	徐玲虹	美工-1-三民主義	98	0	74
		資訊-2-三民主義	93	0	66
		實處-1-三民主義	99	0	80
		電機-1-三民主義	93	0	76
		餐管-1-三民主義	95	60	72
		連子-1-三民主義	91	71	80
	許福晉	建電-1-三民主義	90	0	73
		建餐-1-三民主義	95	0	76
		建餐-2-三民主義	96	0	75
	許寶貴	商科-1-三民主義	81	71	77
		餐管-3-三民主義	95	0	72
		建處-1-三民主義	92	0	86
		電子-1-三民主義	96	0	85

圖 4-8：教師授課成績範圍一覽表

## 5. 結論

面對急驟變化的資訊科技環境，中等學校校務行政系統的開發需要快速隨環境更動，因此如何快速開發，並節省人力均是一大考驗。本研究提供了一個系統開發的架構與實驗驗證，讓參與的伙伴減輕負擔，並提供日後開發系統者一個核心架構方法論，作為日後開發程式的參考。在此研究中充分表現出 SOA 在中等學校校務行政系統協作上的優點。後續我們也希望能結合上、下游學校間的聯繫管道，並漸次將系統架構與 Web Service 接軌，讓整體的運作能更具彈性。

### 參考文獻

[1] 尤弘志，“文件服務架構介紹”，2001年4月。  
 [2] 吳信輝，“細說「軟體工廠」概念(四)”，<http://www.ascc.sinica.edu.tw/>，中央研究院計算中心，第22卷02期，2006年01月16日。  
 [3] 楊正宏，電子商務致勝教本，初版，金禾資訊，台北市，2002年5月1日  
 [4] 簡西村，“服務導向架構(Service Oriented Architecture)應用”，<http://www.mi>

[microsoft.com/taiwan/msdn/columns/soa/](http://microsoft.com/taiwan/msdn/columns/soa/)，微軟技術專刊，2004年12月。

[5] 簡西村，“運用服務導向於企業應用架構設計”，<http://www.microsoft.com/taiwan/msdn/columns/soa/>，微軟技術專刊，2005年1月。  
 [6] Bichler, M., Kwei-Jay Lin, “Service-Oriented Computing,” *IEEE Computer*, Volume 39, Issue 3, pp.99-101, March 2006.  
 [7] Erl, Thomas, *Service-Oriented Architecture -A Field Guide to Integrating Xml and Web Services*, Pearson Education, Prentice Hall PTR, 2004.  
 [8] Huhns, M., Munindar P. Singh, “Service -Oriented Computing Key Concepts and Principles,” *IEEE Internet Computing*, Volume 9, Issue 1, pp.75-81, January-February 2005.  
 [9] Liu, S., “Service Oriented Architecture Overview,” IBM developerWorks, August 10, 2005.  
 [10] Malloy, B.A., Kraft, N.A., Hallstrom, J.O., Voas, J.M., “Improving the predictable assembly of Service-Oriented Architectures,” *IEEE Software*, Volume 23, Issue 2, pp.12-15, March-April, 2006.  
 [11] Pallos, Michael S., “Service-Oriented Architecture : A Primer,” *eAI Journal*, pp.32-35, 2001 December.  
 [12] Papazoglou, M. P., “Service -Oriented Computing: Concepts, Characteristics and Directions,” *2003. WISE 2003. Proceedings of the Fourth International Conference on Web Information Systems Engineering*, pp.3-12, December 10-12, 2003.  
 [13] Peacock, Robert, “Distributed Architecture Technologies,” *IEEE IT Professional*, Volume 2, Issue 3, pp.58-61, March-June 2001.  
 [14] Singh, Munindar P. and Michael N. Huhns, *Service-Oriented Computing: Semantics, Processes, Agents*, John Wiley & Sons, Ltd., 2005.  
 [15] <http://www.w3.org/2003/Talks/0521-hh-wsa/slide5-0.html>, 2006.  
 [16] <http://msdn.microsoft.com/library/cht/default.asp?url=/library/CHT/modcore/html/deovrDevelopingOfficeSolutions.asp>, 2006.  
 [17] [http://www.oracle.com/technology/sample\\_code/tech/java/codesnippet/webseervices/docservice/index.html](http://www.oracle.com/technology/sample_code/tech/java/codesnippet/webseervices/docservice/index.html), 2006.  
 [18] <http://office.microsoft.com/zh-hk/assistance/HP010967253076.aspx>, 2006.