國 立 台 東 大 學

環境經濟資訊管理學系

碩士論文

植基於 SOA 機制的校務資訊系統授權認證雛形之研究

指導教授：謝 昆 霖 博 士

研究生：劉 世 泓

中華民國九十八年六月

國立台東大學

學位論文考試委員審定書

系所別：資訊管理學系

本班 劉世泓 君

所提之論文 植基於SOA機制的校務資訊系統授權認證雛形之研究

業經本委員會通過合於 碩士學位論文 條件

論文學位考試委員會：

（學位考試委員會主席）

（指導教授）

論文學位考試日期：98 年 6 月 22 日

國 立 台 東 大 學

博碩士論文授權書

本授權書所授權之論文為本人在 國立臺東大學 環境經濟資訊管理 系(所)
_____組 97 學年度第 2 學期取得 碩 士學位之論文。
論文名稱：植基於 SOA 機制的校務資訊系統授權認證雛形之研究

本人具有著作財產權之論文全文資料，授權予下列單位：

同意	不同意	單　位
☑	☐	國家圖書館
☑	☐	本人畢業學校圖書館
☐	☑	與本人畢業學校圖書館簽訂合作協議之資料庫業者

得不限地域、時間與次數以微縮、光碟或其他各種數位化方式重製後散布發行或上載網站，藉由網路傳輸，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

☑同意 ☐不同意，本人畢業學校圖書館基於學術傳播之目的，在上述範圍內得再授權第三人進行資料重製。

本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將全文資料延後半年再公開。

公開時程

立即公開	一年後公開	二年後公開	三年後公開
			√

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，本人同意視同授權。

指導教授姓名：[signature] (親筆簽名)

研究生簽名：[signature] (親筆正楷)

學　　號：4396016 (務必填寫)

日　　期：中華民國 98 年 6 月 22 日

1.本授權書 (得自 http: //www.lib.nttu.edu.tw/theses/ 下載) 請以黑筆撰寫並影印裝訂於書名頁之次頁。

2.依據 91 學年度第一學期一次教務會議決議:研究生畢業論文「至少需授權學校圖書館數位化，並至遲於三年後上載網路供各界使用及校內瀏覽。」

授權書版本:2008/05/29

謝誌

在歷經二年的時間後，本研究終於順利完成，首先衷心感謝謝老師的悉心指導與鼓勵，從一開始研究的規劃、研究方向的選擇、觀念架構之建立，以迄本文之撰寫，在老師不斷地予以指導與啟發，使得本專題得以順利完成，謹致以最深的謝意。

其次，感謝同學的辛勞，由於大家的認真與互助，亦是本研究能順利完成最不可或缺的因素，而在這段研究期間，同學間培養出的革命情感，相信會在每個同學的心中留下深刻的印象，畢業在即，期許各同學在未來的日子裡，能更加成長、茁壯。

植基於 SOA 機制的校務資訊系統授權認證雛形之研究

作者：劉世泓

國立台東大學 資訊管理學系環境經濟資訊管理碩士班

摘 要

校園內對資訊系統需求殷切，故系統發展蓬勃，隨著應用服務增加而日益複雜的網站結構，造成各個服務網站間的資訊無法有效率相互交換，在各資料間的轉檔、整合，相當的費時繁雜。致使這些服務系統必須一再地建置基本功能，諸如帳號、密碼驗證及服務權限控管等系統功能。

在本論文中，透過服務導向架構(Service-Oriented Architecture，SOA) 讓異質系統整合變得容易，程式再用度也提高，應用系統是由程式的概念，轉換為服務的概念，將驗證服務以 Services Provider 之介面提供，並製作 WSDL 文件，提供各 Host 端提出驗證需求，建立一個通透而統一標準之認證授權機制，且不改變使用者端服務程序，同時結合 Role-Based 的權限控管，以 Web Services 為基礎提供認證授權服務介面，整合系統各項帳號管理，在資源取得窗口上單一化與集中管理以提高安全性、一致性與完整性。

關鍵字：服務導向架構、Services provider、WSDL、Web Services

ABSTRACT

The strong demands of the Information System at campus prosper the rigorous development of system applications. The growth of services causes rapid expansion of different web-based applications. Those hosts must build common functions including account validation and role authorization. As a result, the redundancy has incurred inefficiency in file sharing, information exchange, and data integration among the non-integrated systems.

Service-Oriented Architecture (SOA) streamlines the integration of heterogeneous hosts and software reuse. The concept of SOA is turning "program-based" into "service-based". In the thesis, the interface is designed to offer the validation mechanism for various hosts with WSDL document communication. Through the unified and transparent authorization mechanism, the role-based authorization on various applications needs no change in procedures (APIs) provided to users. In short, the framework in my research is a web-based service integration in validation, authorization, and account administration to ensure the data security, consistency and completeness.

keyword：Service-Oriented Architecture、Services provider、WSDL、Web Services

目　錄

圖　目　錄

第一章　緒論

1.1 研究動機

HTTP 開啟資訊世界的網路時代，大量的網站服務利用 HTML(hypertext markup language)格式傳遞資訊，校園的資訊系統也有廣泛的應用，舉凡校務行政、教學研究、學生社團及生活與學習環境，都已離不開校務系統的輔助，校務資訊系統已成為不可或缺的行政中樞。

校園內各行政單位對資訊系統需求殷切，故構築在 Internet/Intranet 上之校務系統發展蓬勃，隨著應用服務增加而日益複雜的網站結構，造成各個服務網站間的資交換不易，王昭嵐和朱斌妤(2000)提到早期所發展的系統多是單位各自為政，沒有經過整合，因此現在要重新開發新的軟體系統，在各資料間的轉檔、整合，就相當的費時繁雜。劉燕燕(2004)認為早期校務行政資訊是獨立運作之資訊系統，所用之技術各有不同，因缺乏統一標準，致資訊無法跨平台交流，而嚴重影響行政效率。這些服務子系統必須一再地建置基本功能，諸如帳號、密碼驗證及服務權限控管等系統功能，致使資源浪費。

Bill Gates(1999) 認為：公元 2000 年後的關鍵是「速度」，而企業要擁有「快速回應」的能力，須有效的工作流程與資源整合，因此更加凸顯出「整合」(Integration)的重要性。校園 e 化(政府 e 化)即應用網路與資訊科技的幫助，以改變行政作業流程、教學及研究的環境，使經營得以更有效率。效率提升不單單是網路基礎建設的提升，提供軟體服務的品質亦須相對提升，然國內的軟體服務相關研究似乎仍有不足。

在本論文中，期望建立一個標準認證的授權機制，結合角色為主(Role-Based)的權限控管機制，構築服務導向架構(Service-Oriented Architecture，SOA)之模式將認證服務，以 Web Services 為基礎提供服務，此 Provider 的角色將服務以 WSDL (Web Services Description Language)描述，建立標準介面，透過這個機制整合各系統在帳號管理的功能，增加其一致性、完整性與正確性。在資

源取用窗口上單一化，以減少各系統建置成本、工作產能與效率，簡化與集中管理以提高資訊安全，整體概念示意如圖 1-1。

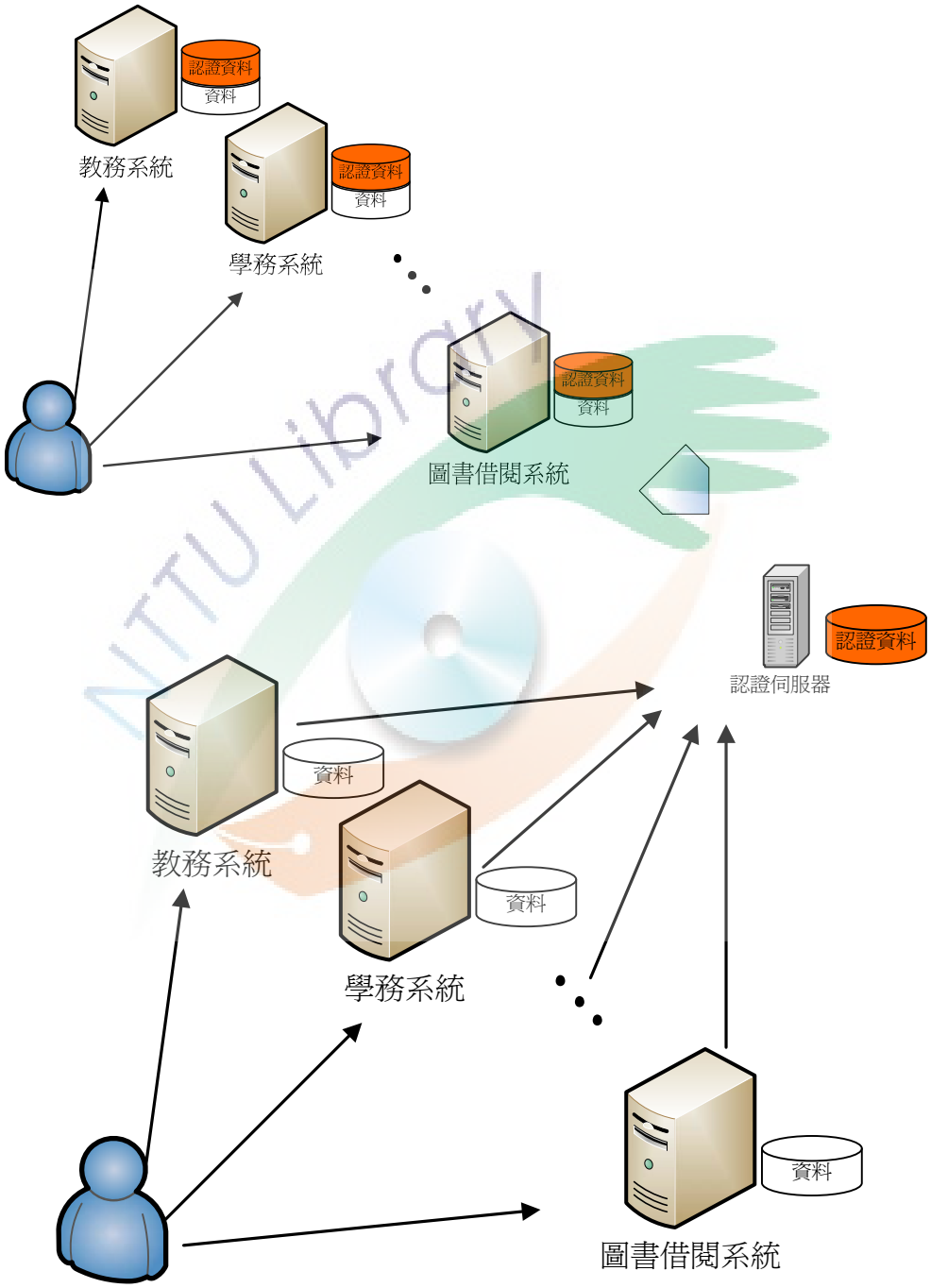

圖 1-1 統一認證授權機制示意圖

1.2 研究目的

謝文全(2002)認為行政工作則包含教務行政、訓導(學務)行政、總務行政、人事行政以及公共關係行政等項目。是對學校教學以外的事務做系統化的管理，以求有效而經濟的達成教育的目標。校務行政系統提供教職員生透過網路，進行各式資料建檔、維護、申辦等業務，透過網路化減少紙張浪費、也降低地域時間等限制。相對往往各子系統需建置相同學籍資料及認證機制，時間久遠將造成資料不同步而有所差異，且各子系統對帳號安全防護亦須著墨，而增加維護的難度。

由於 Web Services 與 XML（Extensible Markup Language）前導技術的成熟，基於 Internet 公開且標準的架構，建構具低成本的認證機制為目標，一方面提高帳號管控力，帳號認證的同時附加權限控管，Ferraiolo and Richard. Kuhn 學者(1992)提出以角色為主之權限控管(Role-Based Access Control)，以使用者角色權利三層配對的概念，每個使用者可以分配到一個或多個角色，而每個角色又可以分配一個或多個權利。也就是依照使用者所分配到的角色來判斷使用者可以得到何種權利，此概念示意如圖 1-2。

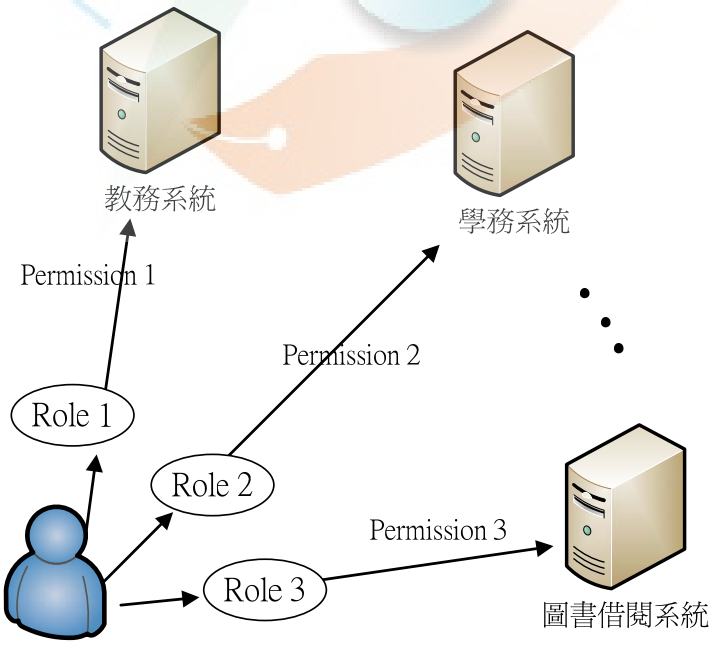

圖 1-2 各系統依照使用者角色判斷可得權利示意圖

本研究期望能提供完善認證架構，不受平台限制，發展具備以角色為主之權限控管之運作機制，且利用現行網路標準及不更改使用端環境，一方面增加校園內部工作流程效率，另一方面還使得系統間進行資訊流通的方式更加迅速和經濟，將降低資料交換傳送的時間與成本而提高整體效能優勢。

1.3 研究範圍與限制

本研究將建立實作系統，分成幾個主要部份：

1. 範圍僅止於校園網域內之部分組織、角色模型及工作管理機制，提供主機認證服務介面。
2. 將驗證服務以 Services Provider 之介面提供服務，並製作 WSDL 文件。
3. 各 Host 主機透過 Web Services 經由網路在 UDDI(Universal Description, Discovery and Integration) 取得 WSDL 文件後，向 Services Provider 發出 Request，並獲得回應，取得認證服務。

1.4 研究流程架構

依據研究動機與目的，界定出所想探討之問題方向，並針對研究主題進行文獻蒐集與探討。整合探討過程，除先分析目前各相關研究的方向，亦從文獻和前人研究成果中去比較，以探討功能需求，進行構思。完成研究架構設計後，進一步著手建置權限控管，詳細研究步驟如圖 1-3 所示。

第一章、緒論

說明本研究動機、目的、及研究架構。

第二章、文獻探討

本研究主要的相關研究探討，將介紹相關的背景知識以及現存的機制，內容涵蓋 XML、Web services、RBCA。

第三章、研究設計

本研究主要的系統功能架構研究設計，並將探討各項相關參考技術與問題。

第四章、本文所提之授權機制

針對本研究之相關的系統分析與設計。

第五章、結論與展望

歸納出本系統架構的特色，及提出未來研究方向與建議。

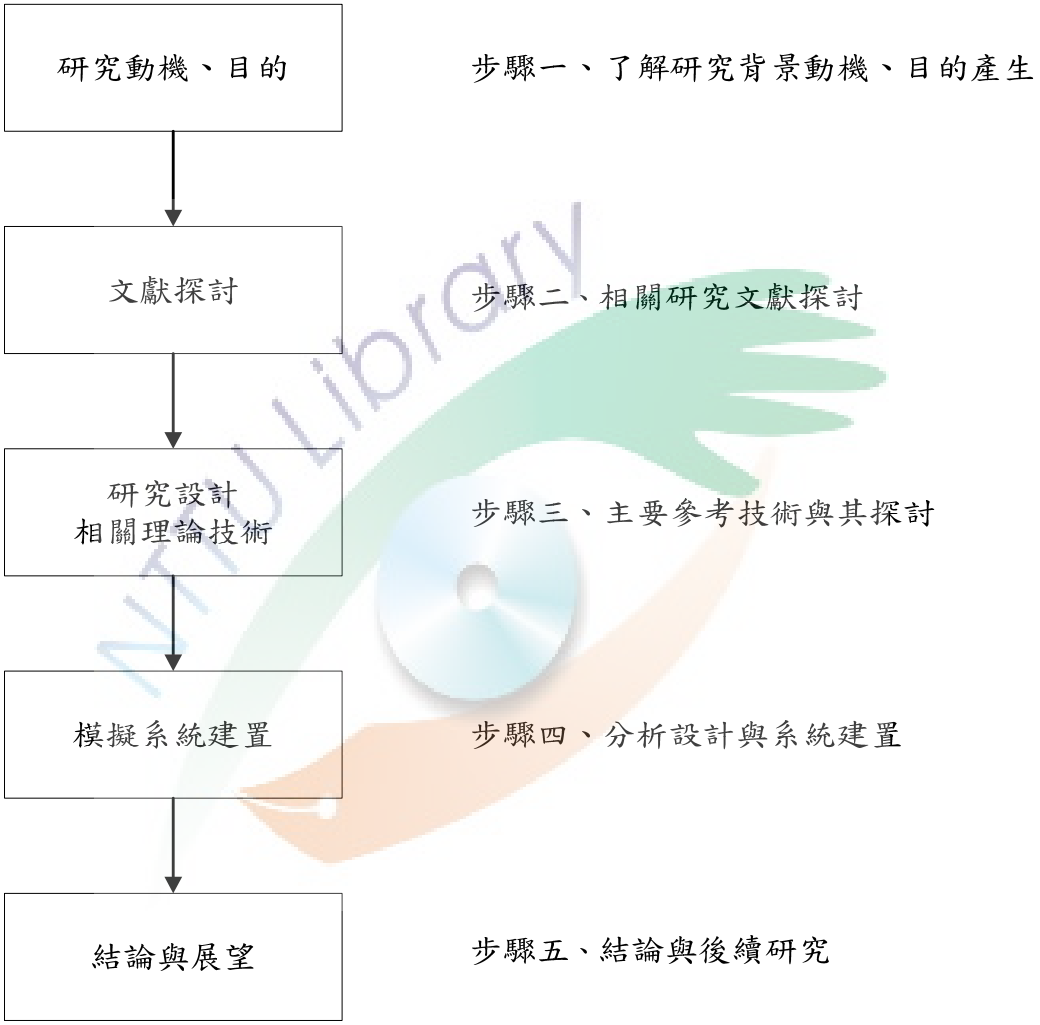

圖 1-3 研究流程架構圖

第二章 文獻探討

2.1 服務導向架構(Service-Oriented Architecture，SOA)

結構化資訊標準促進組織(2006)建立以下定義：“ Service-oriented architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.”。服務導向架構(SOA)是一種新興的系統架構，主要概念是以組織(學校)需求組合的軟體元件服務群，應用系統是由程式的概念，轉換為服務的概念。Michael Huhns and Munindar P. Singh (2005)提出服務架構應用主要有三個部分組成：供應商(provider)，消費者(consumer)，以及註冊機制(registry)，概念示意如圖 2-1。服務提供者(供應商)先將服務註冊到服務仲介者(註冊機制)，接著服務需求者(消費者)在服務仲介者發現所需的服務後，最後再與服務提供者做連結，來完成服務的傳遞。

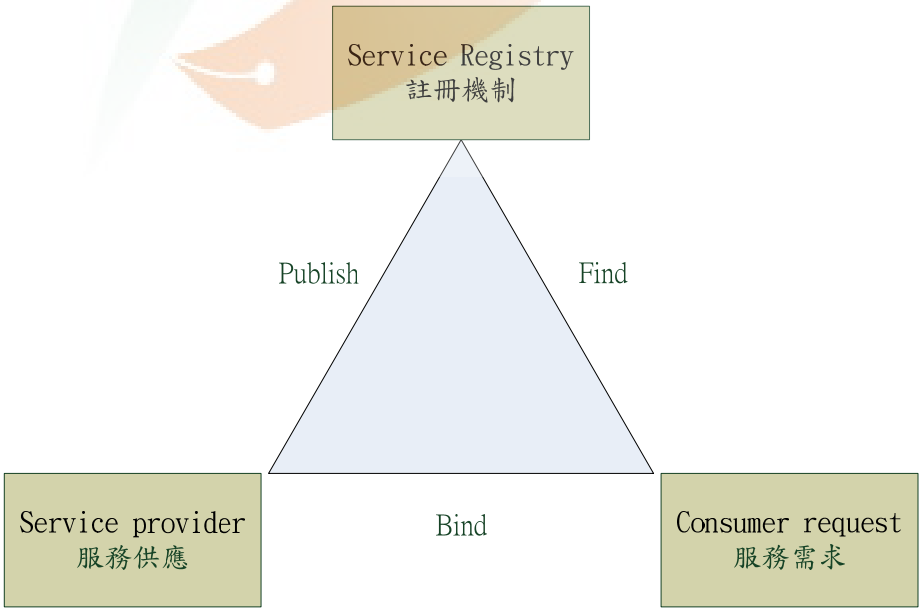

圖 2-1 服務架構三個部分組成示意

簡西村(2004)認為組織(學校)對需求組合元素通常包括：軟體元件、服務及流程三個部份。當組織(學校)面對服務要求時，流程負責定義外部要求的處理步驟；服務包括特定步驟的所有程式元件，而元件則負責執行工作的程式。以資訊系統登入驗證為例，當使用者(教職員生等)登入系統作業時，流程制定輸入或查詢權限輸出的處理步驟，服務則包括驗證查詢授權，元件則是指真正完成授權等工作的程式。

2.1.1 服務導向架構之特性

透過 SOA 讓異質系統整合變得容易，程式再用度也提高。Bieberstein 等(2004)指出 SOA 是一個基礎框架，用來整合業務流程並支援 IT 基礎架構，成為一個安全、標準化的元件服務，而這些服務元件可以整合與再利用，以滿足商務需求的快速變化。IBM 的全球服務（Global Services）是 SOA 的領導廠商之一，其對 SOA 的概念為，將組織本身視為流程和服務的彙整，而 SOA 是這些應用軟體的框架，它讓這些應用軟體可以更容易的再利用，並結合不相連結的流程及服務，以構成組織的整體內容架構。無論是針對商業模式（business model）、基礎架構模式（architecture model）及程式化模式（programming model）都採用一個單一的概念 —「服務」。

「business model」中清楚定義業務服務內容，並且將它們整合到業務流程中。而「architecture model」則是加入了業務服務及實行架構的型態、原則和模式，利用現有或是新的 IT 基礎架構來實現這些服務。最後，「programming model」裡則實作這些服務，讓這些服務在異質環境中也能夠相互作用。能應用於系統上的相關特點：

1. 重用性(reusable)

Thomas Erl(2005)所描述 SOA 是一個開放性，可擴展、聯合、組合結構服務的方向概念，是基於它有利於打破一個大問題分成了一系列的個別問題。這使得需要解決的問題，得以由分解成較小相關組件組裝服務，可以協調工作並組合起

來形成一個組合服務，服務的目的是支持潛在的重用。透過利用現有的元件和服務，可以減少完成軟體發展生命週期（包括收集需求、進行設計、開發和測試）所需的時間，這使得可以快速地開發新的業務服務。

2.鬆散的契合(loosely coupled)

系統主要是將應用程式功能需求切割，繼而由小模組、物件或元件組合成相互關聯的應用，開發者要費心於了解零組件是如何設計及使用，以確保不會違反零組件連接關係限制。若要以不同零組件替換原始組件，就成為一件不容易的事。SOA 的作法是以標準界面來組合系統，只要符合界面要求，零組件可以任意替換，大幅提高系統的彈性。Nicolai Josuttis(2007)指出大型分佈式系統中兼顧可擴展性和容錯性，組件修改或失能時，整體的影響得以降低。

3.分散式多層架構 (Distributed N-Tier)

SOA 的組成元件是由許多分散在網路上的系統組合而來，例如網路服務技術(Web Services)就是運 Internet HTTP Protocol 來相互連結，而擺脫地域的限制，可能是區域網路，也可能是來自廣域網路，將平台架構上的元件與模組以層次化的設計與架構來提供服務。

2.2 Web Services

當一個組織想要利用網路來達成，將分散在各地、各個不同的主機上的資料整合，因應趨勢的資訊科技發展就會孕育而出，早期的傳真服務 (Fax)、EDI (Electronic Data Interchange)等，都是為了解決彼此間的資料溝通，Web Services 的誕生即在於網路基礎建設普及與 XML 發展成熟。Web Services 是一套可讓各服務提供者將自己的服務經 XML 描述出來，並且放在公眾的資料庫上讓服務需求者去查詢、驅動，這樣的環境是建構基於 SOAP、WSDL 和 UDDI 所建立的服務導向機制。藉由將組織在網路上所提供的服務，以 WSDL 的標準描述文件註冊到 UDDI，以達到集中管理各種服務以及使用者容易搜尋的目的。

Mary Kirtland(2001) 於 Microsoft 對於 Web Services 定義為「Web Services 是可一套可被程式設計的應用軟體，並且可透過標準的 Internet 協定來存取，結合元件導向與 Web 呈現 Black-box 的功能，使得提供的服務可以被重新使用，而不用擔心此種服務如何被撰寫，用戶可以在任何平台任何語言實作使用，只要依循 Web Services 的介面即可。」Web Services 可以讓應用程式之間的溝通更為容易，並且透過這樣的一種方式可以使得分散在不同地點的應用程式，透過 Internet 而整合在一起，形成一個完整或是大型的資訊系統，有別以往侷限於單一主機上的資訊系統。

異質系統的整合並非時至今日才提出，其牽涉範圍泛括不同的機器實體、作業系統及應用程式，戚玉樑等(2003)提及各組織相關規範架構:Object Management Group（OMG）這個組織所推出的 Common Object Request Broker Architecture（CORBA）、Microsoft 所推出的 DCOM/COM+。此外 Sun 提出的 RMI 與 EJB，以及 Macromedia 的 Flash Remoting 等技術，奠定了 Web Services 的技術基礎。

2.2.1 Web Services 的特性

Web Services 提供一個嶄新的建構模式，將傳統的分散式系統所提供的服務標準化，並以 Internet 作為服務傳遞的媒介。透過服務，當你在本地端撰寫應用程式時，可以在自己的平台環境中直接引用該服務，依該服務所提供的介面而得以引用到最新的功能，雖然該服務提供在遠端的機器上，而使用該服務時就像是使用在自己平台上的元件。Web Services 可扮演一種在網路上運作的資源存取介面，讓所有在網路的應用程式能夠彼此互動。如果將 Web Services 導入組織，更可以降低系統整合與軟體開發的成本，並且促進運作流程的效率。

Web Services 架構最大的優點在於它突破了以往系統整合與開發上應用系統與程式元件間 Tightly-Coupled 的運作架構模式，Web Services 主要架構由三項核心技術所組成 UDDI、WSDL、SOAP 以及 XML 等，說明如下:

2.2.2 UDDI (Universal Description、Discovery and Integration)

UDDI 的是一種有關於 Web Services 的目錄註冊服務，其主要的目的是將提供者的 Web Services，藉由 UDDI 告知服務需求者有哪些服務提供，其功能猶如電話簿或是黃頁，需求者由 UDDI 所羅列服務選取合適者使用。UDDI 包含了兩個部份－服務描述及服務尋找的規範，是由 Ariba、IBM、Microsoft 等廠商組織共同制定的註冊以及儲存資料庫的標準，記載著提供 Web Service 的聯絡資訊，以及 WSDL 格式的 Web Service 使用操作資訊。UDDI 註冊資訊是由下列五種型態的資料結構所組成，圖 2-2 為各資料模型間的關係示意圖:

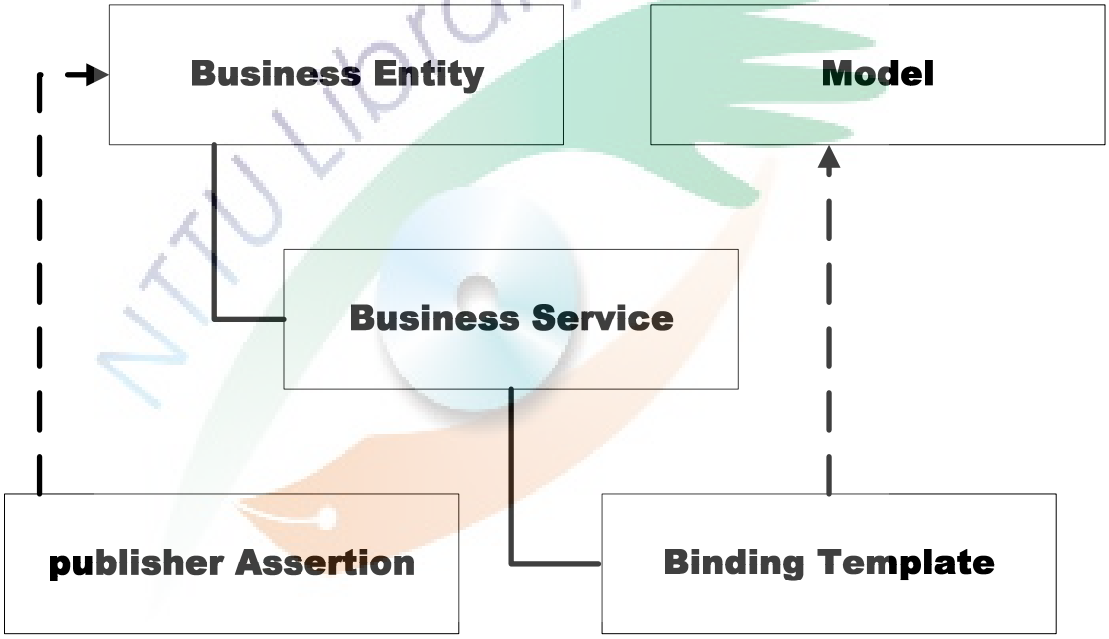

圖 2-2 資料模型間的關係示意圖

分別是：

1. 企業實體(business Entity)，是最上層的結構，描述企業資訊被註冊的實體， 其它的結構藉由參考這個結構產生關聯，服務的查詢者還可以根據以姓名分類或商業性質分類的方式來查詢欲使用的服務。
2. 企業服務(business Service) ， 被發佈的服務名稱與描述，目的就是用

來封裝一系列商業流程資訊。

3. 鏈結樣板(bindingTemplate)，描述服務相關資訊，封裝了通訊資訊、使用順序、路由選擇等資訊，包括存取服務的入口點位址。
4. 模型(Model)，是註冊資訊的核心，其最大功用就是描述資料，有關服務更具體的資訊都在 Model 之中說明，這個資料結構也支援最上層的搜尋。
5. 發佈者聲明(publisher Assertion)，依據明確的關聯型態，可以使兩個或兩個以上的企業實體結構之間產生關聯，例如部門關係。

2.2.3 WSDL (Web Services Description Language)

WSDL 是一種描述網路服務的技術語言，主要用來描述網路服務如何溝通，有什麼樣的服務內容等資訊，以 XML 形式呈現及定義服務介面，WSDL 涉及執行描述(Implementation Description)及介面描述(Interface Description)。介面描述部份將運用介面描述語言（Interface Description Language，簡稱 IDL）描述介面，將完整描述 Web Service 的介面名稱、參數名稱、參數型別。基本網路服務描述與 WSDL 的對應示意如下圖 2-3，其內容包含

- 服務型別（Type）：定義各 Element 實際對應之資料型態。
- 訊息（Message）：定義各輸入、輸出 Message 由哪些參數 Element 所組成。
- 服務流程（PortType）：Service 所有 Ports 提供之全部 Operations 集合，傳送端與接收端支援的作業種類。依據 XML 規格規定，每一個 portType 都是以其支援的作業進行定義，WSDL 指定四種基本作業型態， 分別是： 通知(Notification) 、單向(One-way) 、請求 Solicit-response) 、要求(Request-response)。
- 服務鏈結（Binding）：Binding 所使用的通訊協定，以及提供之 Operations，定義訊息格式與每種 portType 指定作業的協定。每個 Binding 根據每項作業的輸入、輸出與錯誤都有唯一的識別名稱。

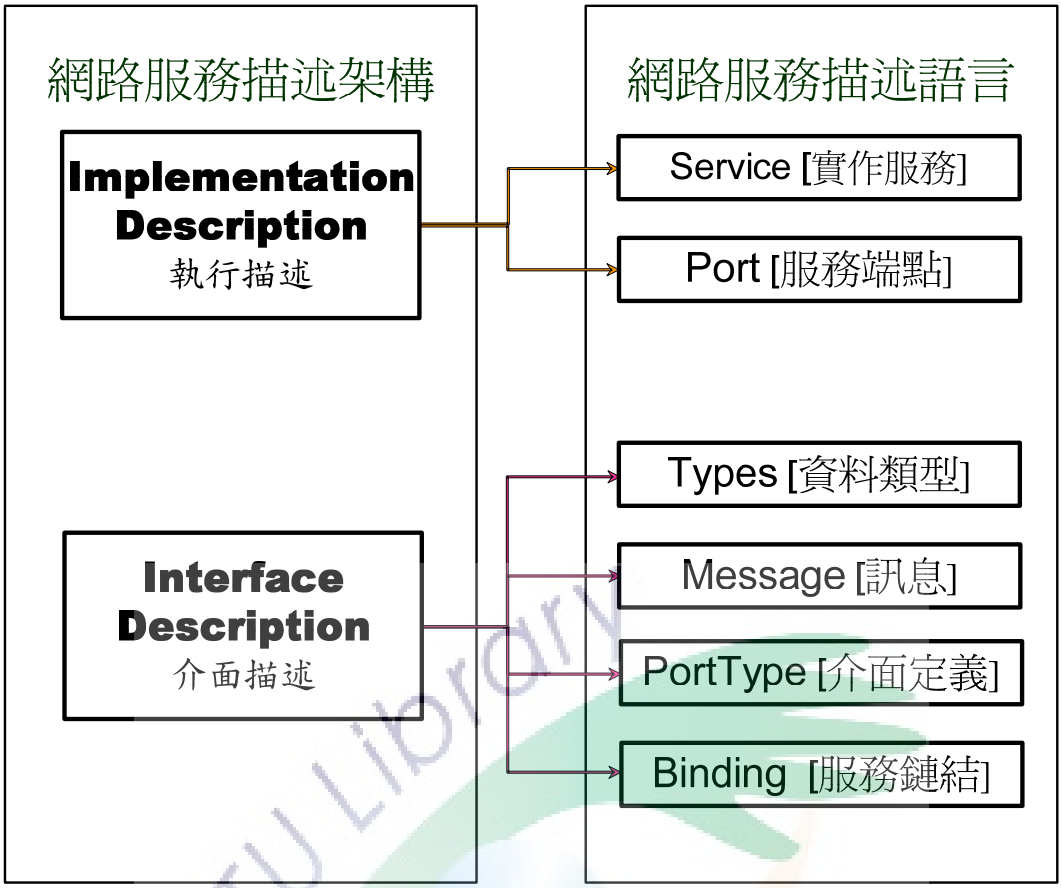

圖 2-3 基本網路服務描述與 WSDL 的對應示意圖

執行描述將描述 WSDL 文件內，有關介面的實做資料，包含描述服務的名稱、提供服務的廠商以及服務的網路位址，其內容包含

- <service>： 此 WSDL 文件所要描述的 Web Service 集合。
- <port>：訊息傳送端與接收端的網路實體位址，每個 Port 代表外界 Client 可和此 Service 溝通的進入點，一個 Port 會指定一個 Binding 的方式。
- <types>： 定義各 Element 實際對應之資料型態，端點之間交換訊息的資料型態，在網絡服務的架構之中，WSDL 扮演相當重要的角色，內容包含取得服務的方式與型態等，透過完整的 WSDL 可以協助使用者迅速的得到所需的服務與資訊。

2.2.4 SOAP (Simple Object Access Protocol)

SOAP 是屬於輕量級的協定，應用於分散式電腦環境中交換資訊，其作用是編譯網路服務所需的要求或回應後，再將編譯後的訊息送出到網路，簡單來說就是應用程式和用戶之間傳輸資料的一種機制。不同平台之間可以藉文字格式的方式，使得應用系統可以相互溝通；亦即是利用現存的網際網路架構讓應用程式之間可以彼此溝通，而不會被防火牆阻礙。

SOAP 是以 XML 為基礎的通訊協定，提供兩個電腦系統之間交換的架構與資料型別。

1. 封套(SOAP envelop)：SOAP 訊息中最外層的元素，構成 XML 訊息文件的根元素，且為必要元素，定義如何描述訊息以及如何處理訊息，一個整體的表示框架，可用於表示在訊息(message)中的是什麼，主要內容為一般命名空間及編碼命名空間，可以包含名稱領域(Namespace)的宣告以及其他額外的屬性宣告。

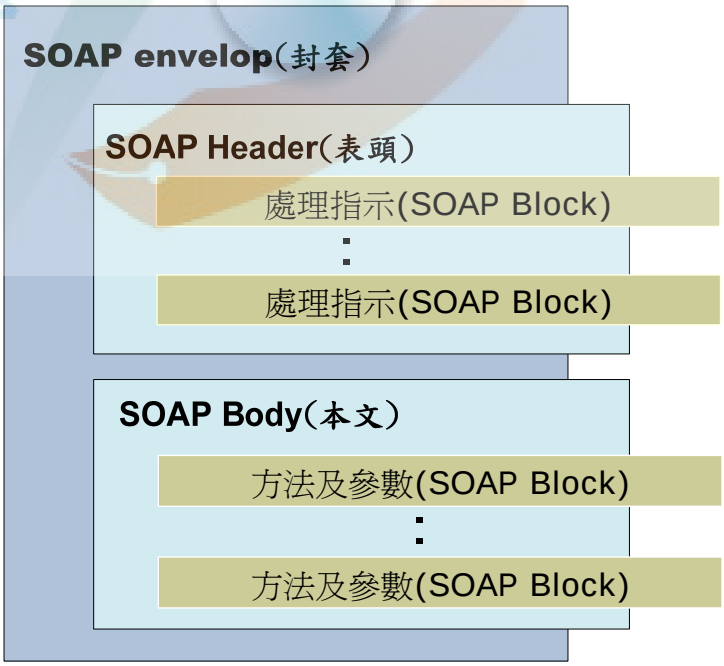

圖 2-4 SOAP Envelope 訊息架構

envelope 裡頭包含了三種格式，訊息架構如圖 2-4:

- 記錄了相關的 namespace
- Header 記錄了編碼的方式以及如何處理該 SOAP 訊息
- Body 包含了 SOAP 中真正的內容

2. 編碼規則(SOAP encoding rules)：定義了一編碼機制用於交換應用程式，定義的資料類型表示應用程式定義的資料格式。
3. 協定(SOAP RPC representation)：定義了一個用於表示遠端程序呼叫和回應的約定，表示遠端程序呼叫（RPC）及其回應。
4. 繫結協定(SOAP binding)：繫結（binding）交換的訊息。SOAP 可以與其他的協定結合使用，目前只規範 SOAP 與 HTTP 的結合使用作詳細的說明，定義了底層傳輸協定來完成在結點間交換 SOAP 信封的約定。

SOAP 的特性在於使用特定的封包結構，方便於把應用程式的資訊打包放進 XML 文件中，並且定義一些規則來處理這些文件，這個結構就是 SOAP Envelope，它包含了兩個元素：一個 Body 元素(必要)以及一個標頭元素(非必要），前者可放入與應用程式相關的資訊，通常一個 Body 元素可以有多個 body 記錄，而標頭元素是用來寫入與應用程式無關的資訊，如 SOAP1.1 的 actor 屬性，就可以是指定特定目標來要處理標頭記錄，也就是 SOAP 的路徑控制。

2.3 存取控制相關文獻

存取控制相關研究將針對存取控制的定義、方法、策略與以角色為基礎的存取控制（Role-Based Access Control ,RBAC）進行介紹與說明。

2.3.1 RBAC（Role-Based Access Control）

以角色為基礎的存取控制(Role-Based Access Control, RBAC)，(David Ferraiolo,2001)提及先後由多位學者提出之存取權限控管模組，再由 National Institute of Standards and Technology (NIST)組職加以匯編、

整理之後訂定出標準，稱為NIST RBAC。此機制以使用者的角色來判斷其存取權限，設定每個角色所能使用的資源及權限，每個使用者都屬於一個以上的角色，而使用者只要是屬於某個角色，就能取得這個角色所能取的各項資源。

它是在一般的存取控制政策中，在使用者(User)及權限(Permission)中，多增加了角色(Role)的元件，使用者是透過角色的中介元件來存取權限，如此可減化使用者存取權限的異動數量，達到減輕管理負擔之目的。在實際應用系統中，當使用者的數量不多時，管理者可以有效地集中管理的建立、啟用及權限配置。但隨著應用系統日益復雜，角色建立亦趨複雜，透過角色管理將可顯現效益。

2.3.2 RBAC 安全原則

在大規模分散式的環境下，系統幅員廣泛且角色多元，要如何管理它們之間的關係將會是很複雜的問題。若完全倚賴管理員集中式管理，會給管理工作帶來很大的困難，而基於角色的委託授權(Delegation)構想便是希望可以在分散式系統環境中去實現RBAC模型，讓使用者將權限委託給代理者以替委託者去執行某些任務，讓權限的管理可以更加靈活方便，並可以大大的減輕管理者的負擔。例如學校的老師、學生和行政人員等等皆為不同的角色，每一角色對於學校的資源使用權亦有所不同，對於權責的劃分，依據不同的政策（Policy）由管理人員制訂及修改每一角色，再依據所授予的權責行使責任與義務。

在Sandhu(1996) 所提出的基本模型概念，主體與權限是間接存有互動關係，透過角色來達成彼此的關聯性，角色的概念相當於企業組織的職務，代表著具有不同的工作功能。角色被定義為與特定工作有關的責任之集合，能表現特殊的職務分派(Assignment)，物件的存取授權是指定給角色，每個主體被授予可執行的存取控制工作不需一一指定，只要將主體指派到某一特

定角色，主體可依所扮演的角色去執行所有該角色的所被授予權限的存取活動。角色的概念與傳統群組(Group)概念不同，主要是在於群組的對象是人的集合，而角色的對象除了人的集合外，還包括了訂定的權限許可、角色之間承繼的關係、職務間權責合理配置，以及個體互動的特定條件。

2.3.3 NIST RBAC 模式

RBAC 架構以美國國家技術標準局(National Institute of Standards and Technology, NIST)的RBAC 為標準是目前最為完整的架構，此標準主要是由存取控制領域的兩位大師 Ferraiolo & Sandhu(2000)提出RBAC 建議標準。以下為NIST 的RBAC 標準七個基本元件。

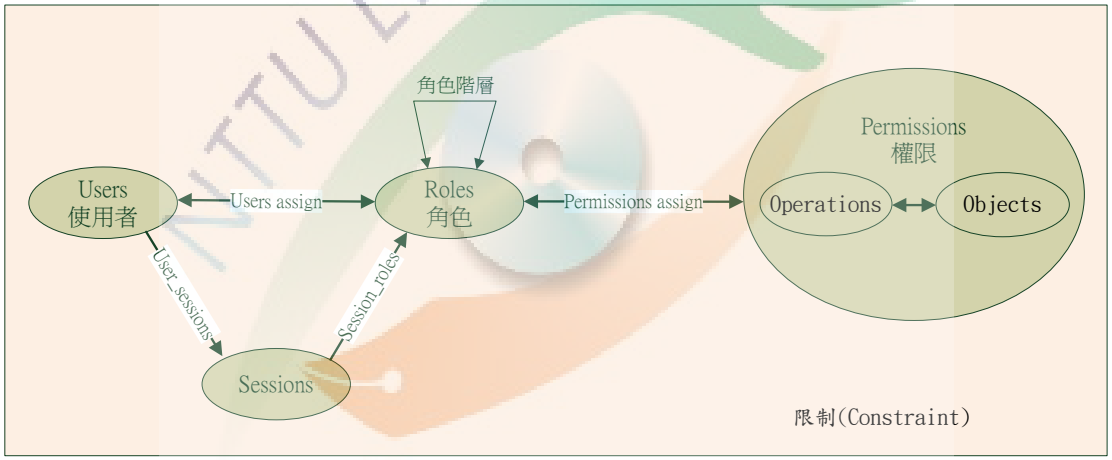

圖 2-5：RBAC 模式圖

(1) 使用者(User)：直接與系統進行互動的人或程式。

(2) 角色(Role)：存取控制機制中扮演的角色，用以描述使用者對應到此角色可被賦予的權限，每一個人可以是不同角色，同一角色可由不同的多人所擔任。

(3) 權限(Permissions)：指存取控制機制中對於物件的權限，包含操作與物件，在NIST 的RBAC 標準中，特別去定義操作與物件。

(4) 操作(Operations)：指對物件的操作能力，包含寫入、讀取與擁有權等，

此外也可以包含抽象的操作動作，如借款、貸款。

(5) 物件(Objects)：指存取控制機制可供存取的物件，可能是檔案、文件或是執行程式等相關的物件。

(6)連線(Sessions)：使用者對應至可使用的角色集合(Active Role Set)的過程，使用者透過Sessions 對應到被允許的角色，RBAC 中使用者可以對應到多個角色，依狀況來轉換角色，操作適當的權限，以達到最少權限的功能，每一人可能同時使用多種角色的權責執行，但對於每一使用者利用一個角色行使使用權時，僅有一個使用連線處理。

(7)限制(Constraint)：用來規定角色彼此之間關係，如階層關係、角色互斥的關係。

2.4 資料安全相關文獻

2.4.1單向雜湊函數

單向雜湊函數(One-Way Hash Function)許建隆、楊松諺(2005)提到可以將任何長度輸入的訊息，經過運算之後輸出一個固定長度的輸出訊息，可以應用在完整性、鑑別性、不可否認性及數位簽章等資訊安全功能。常見的單向雜湊函數有MD4、MD5、SHA、SHA-1等，單向雜湊函數有以下兩個特點：

(1)單向性(One-Way)

將任意明文透過單向雜湊函數運算之後產生一固定長度的輸出，但想要從任何經單向雜湊函數處理過的訊息，要推算出原本的明文，在合理的時間範圍以及有限的資源限制下是不可行的。

(2)無碰撞性(Collision-Free)

將任意的兩個明文輸入，經過單向雜湊函數運算之後，不會產生相同的雜湊值，意即任何兩份不同文件的雜湊值是不會相同的。

2.4.2 SSL通訊協定

SSL（Secure Socket Layer）為目前網際網路上安全通訊的國際標準與通用傳輸加密之通訊協定，目前最新版本為 SSL 3.0 ， SSL 是1994年由 Netscape 公司所提出，現今市面上大多數的標準瀏覽器都有支援 SSL 通訊協定。

SSL 連線主要是建立在應用層與 TCP 層之間，如圖3-1所示，因此在應用層的應用程式可以不需經過太多修改，即套用 SSL 機制以保護傳遞資料的安全。SSL 協定主要包括四個協定，分別是交握協定（Hankshake Protocol）、警告協定（Alert Protocol）、改變加密方式（Change Cipher Spec Protocol）與記錄協定（Record Protocol），其中，交握協定、警告協定與改變加密方式位於相同一層，用來作為認證身分與資料加密前金鑰交換之用；而記錄協定則位於交握協定、警告協定與改變加密方式之下方，用來定義連線雙方資料傳遞的格式，包含資料分拆、壓縮與加密規格。

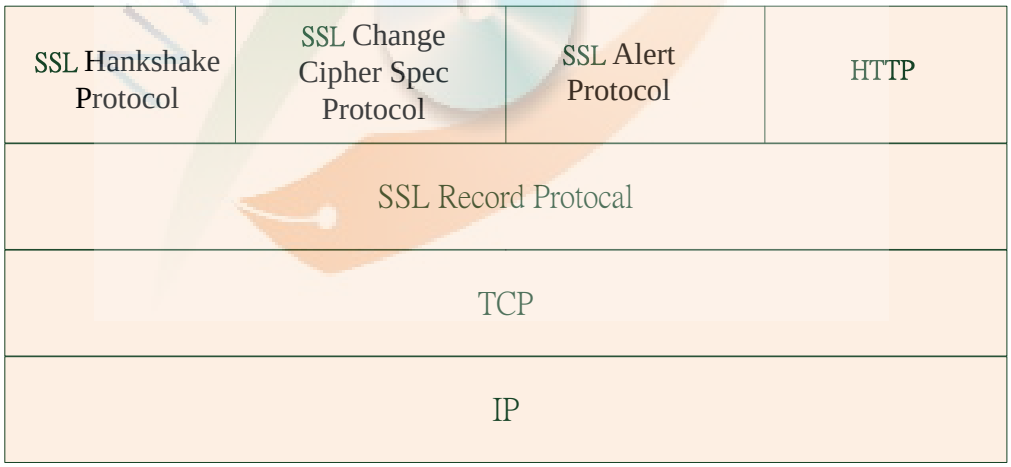

圖2-6 SSL通訊協定架構示意圖

2.4.3 加解密

加密（Encrypt）係指將明文資料（Plain Text）轉變為密文資料（Cipher Text）之處理過程；解密（Decrypt）係指將密文資料（Cipher Text）反轉為明

文資料（Plain Text）之處理過程。在密碼學的領域，通常會應用加解密方式來達到秘密通訊的目的。加解密系統依金鑰的特性，可區分為對稱式（Symmetric）系統及非對稱式（Asymmetric）系統。

對稱式系統係指加密端與解密端使用同一把金鑰（Secret Key）進行加解密；而非對稱式系統係指加密端與解密端透過成對的公開金鑰（Public Key）與私密金鑰（Private Key）進行加解密，其中，私密金鑰僅被解密端所擁有，如圖2-6所示。

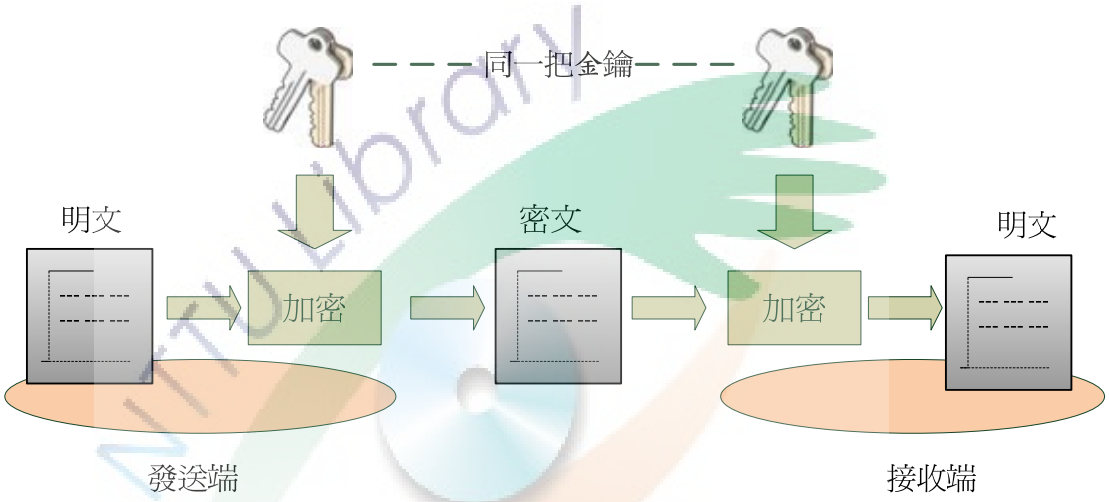

圖2-7 對稱式加密系統示意圖

非對稱式密碼系統（Asymmetric Cryptosystem），又稱為公開金鑰密碼系統，係指明文在加密與解密的過程使用兩把不相同的金鑰。在非對稱式密碼系統裡，加密與解密所使用的金鑰是在同一階段產生且具有對應關係的兩把金鑰，一把稱為公開金鑰（Public Key），係指可以公開給大家知道的金鑰；另一把稱為私密金鑰（Private Key），係指持有者自行持有且不能讓其他人所知道的金鑰，而透過公開金鑰去推導出秘密金鑰是相當困難。在非對稱式密碼系統裡，發送方與接收方不用事先協議與持有加解密金鑰，發送方在發送明文前，將明文經過接收方的公開金鑰加密以產生密文，並且傳遞密文給接收方，而接收方在接收

密文後，即透過本身持有的私密金鑰解密以還原明文，如圖2-7所示。

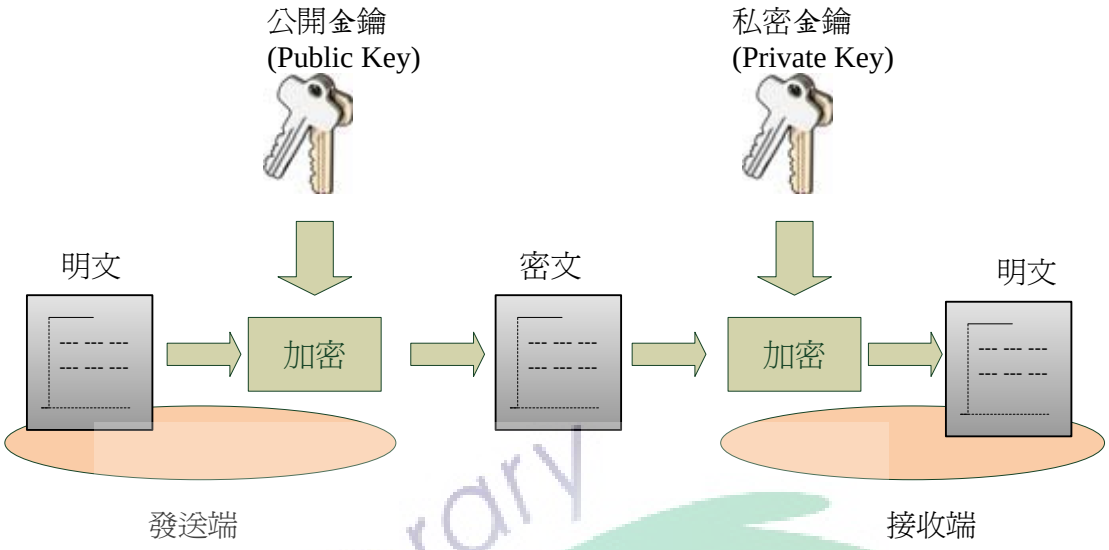

圖2-8 非對稱式加密系統示意圖

第三章　研究設計

本章將介紹本研究所提之認證、授權之流程架構，第一節介紹設計概念，接著第二節介紹系統架構，第三節說明系統運作方式。

3.1 設計概念

校園內資訊系統發展蓬勃，隨著應用服務增加而日益複雜的網站結構，造成各個服務網站間的資訊無法有效率相互交換，致使這些服務系統必須一再地建置基本功能，諸如帳號、密碼驗證及服務權限控管等系統功能。

植基於 SOA 機制的校務資訊系統，如圖 3-1 所示，將各系統的授權認證功能，以服務需求的角度彙整成基礎服務需求，服務供應端相對提供帳密認證、角色授權等基礎服務。

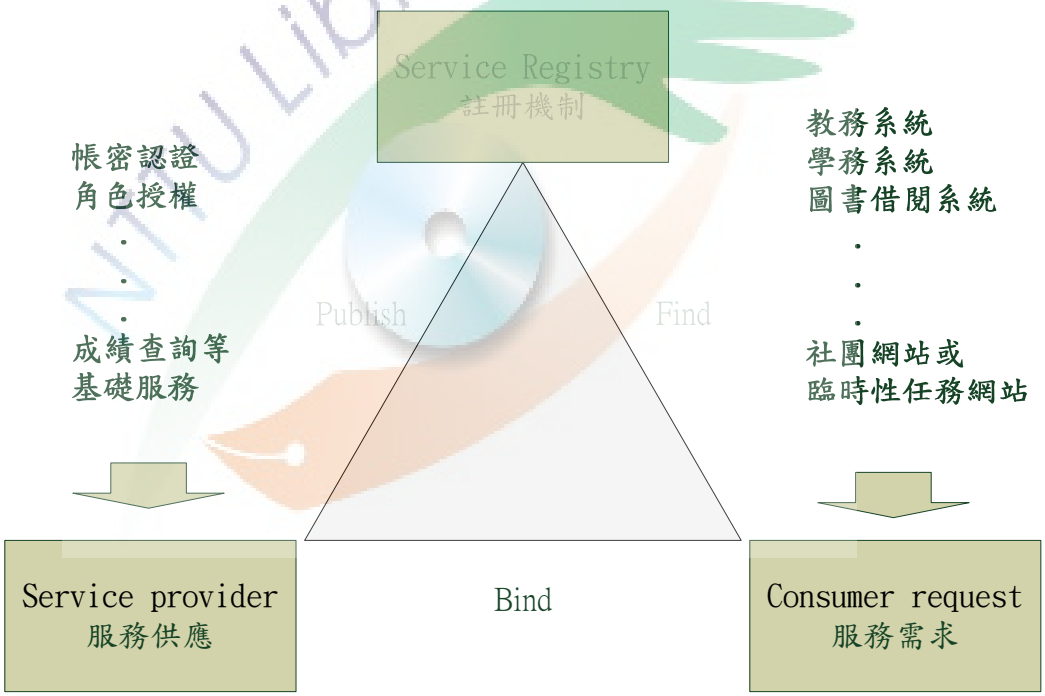

圖 3-1 植基於 SOA 機制的校務資訊系統示意圖

3.2 系統架構簡介

一般系統建置，對使用者帳號及權限模組，每開發一套系統就需將使用者帳號建立一次，隨校務系統應用服務增加，對系統開發者、管理者、使用者的負擔

都是與時漸進，有鑑於此本研究所提出的服務架構，所要加強的地方就在於：

1. 建立認證標程序

   當系統建置者採用本認證服務來建構其系統服務時，必須依循資訊交換標準，由服務主機向認證主機建立註冊，繼而使用認證服務。

2. RBAC 的權限控管機制

   在服務架構中，依循以角色為基礎的存取控制(RBAC) 作為權限控管機制，在 RBAC 的制度之下，一個使用者可能同時屬於多個角色，一個角色則對各項服務皆有不同的使用權限。

3. 採用 Web Services

透過 URL 指定存取 internet 上任何一台電腦提供的應用程式服務，由於它不受作業平台的限制，適合用於本研究所提出的服務架構。在 Web Services 的開放架構之下，本服務架構將能夠脫出平台的限制來提供各項訊息交換服務。任何使用本服務架構的服務網站建置者都能夠自由選擇他/她所喜歡的開發作業平台。本服務架構有五個主要的參與者，針對各個成員和運作分別作深入的說明，請參考圖 3-2：

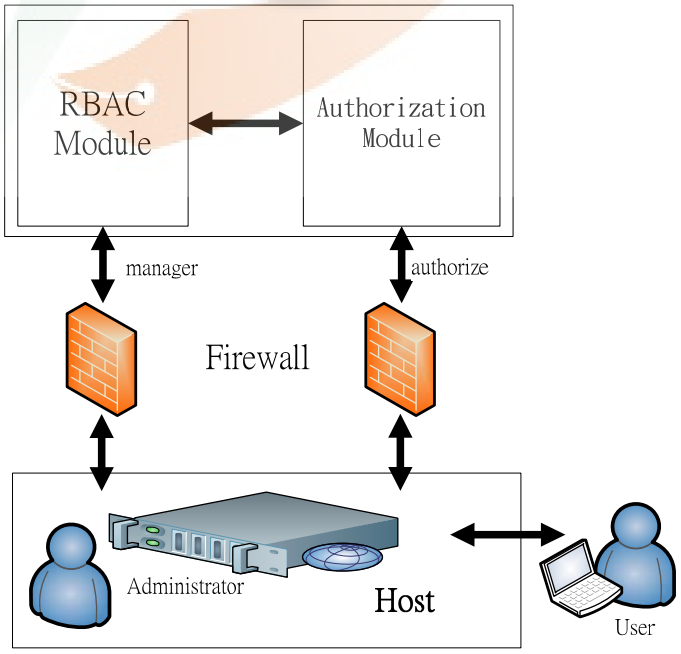

圖 3.2 系統架構和運作說明

1. Host Administrator（主機的管理者）

主機的管理者須先向認證主機提出註冊，儲入其網路位置等資訊，以為提出服務需求。並向 RBAC Module 管理編輯所屬功能及其使用成員。

2. RBAC Module（角色存取控制模組）

角色存取控制模組中存放各已註冊服務主機之權限管理機制，包括每個帳號及其角色所對應的權限資訊，讓系統架構下的服務主機，能夠由認證模組來查詢使用者所持有的角色，是否擁有其所要求之資源的使用權限。

3. Authorization Module(認證模組)

Authorization Module 負責透過 web services 提供認證服務。它的工作內容包括：對有註冊的服務主機依其網路位置等資訊提供認證服務，Host 將使用者登入帳號/密碼提交驗證，Authorization Module 將驗證結果回覆。

4. Host(服務主機)

在本服務架構下提供使用者各項服務的服務主機，這些主機本身的認證資訊必須記錄在 RBAC Module 中。並且，這些服務主機的管理者須先提出註冊，如此方能建立可信任的關係，以利訊息交換。Host(服務主機)的型態不侷限於 WEB 服務或視窗應用程式。

4. Users

即一般的使用者，泛指任何使用本服務架構下之 Host(服務主機)的使用者。

3.3 運作方式

在架構中，使用者、Host(服務主機)、認證模組與 RBAC Module 的互動步驟為：

(1) 使用者透過 Host(服務主機) 進行登入動作。

(2) Host(服務主機) 將使用者所輸入的認證資料送至認證模組。認證模組根據 Host(服務主機)所傳來的使用者認證資訊，進行使用者身分認證動作，以取得該使用者的資料。若使用者所輸入的認證密碼無誤，則繼續進行授權步驟。若使用者輸入的資料有問題，那麼就直接進行步驟 5，並傳回錯誤提示訊息。

(3) 認證模組向 RBAC 模組索取使用者於該 Host(服務主機)的使用權限。

(4) RBAC 模組將該使用者的權限資料傳回給認證模組。

(5) 認證模組將使用者登入等一連串動作之執行結果回報給 Host(服務主機)。

(6) Host(服務主機)依據認證模組所傳回的資訊決定該使用者，是否有權使用系統所提供之服務。

3.4 系統設計

本節說明實作面所需考量問題，並以 UML 循序圖描述概觀，如圖 3-3。

情境簡述如下:

1. 使用者開啟登入驗證視窗，並輸入帳密資訊。
2. 檢查帳密資訊是否吻合。
3. 若帳密確認，擷取使用權限，含括該帳號角色權限。
4. 將權限資訊回覆。
5. 將權限資訊反映於該主機系統。

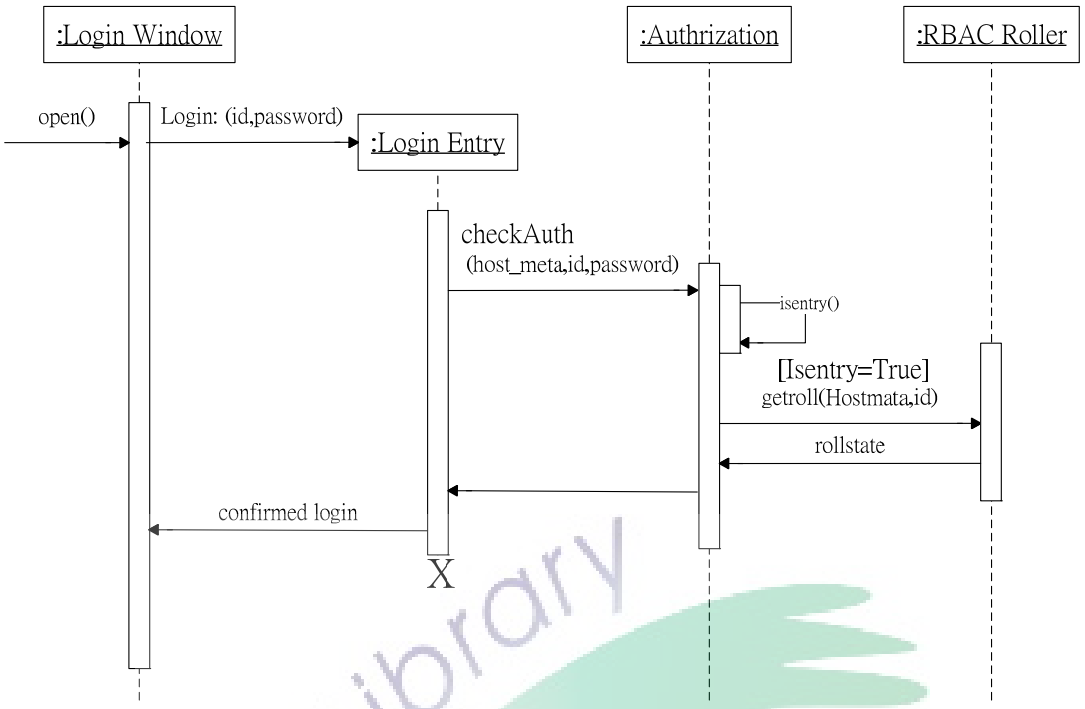

圖 3-3 系統運作 Sequence Diagram

第四章 模擬系統建置

4.1 系統描述

本研究機制主要包含四個主要成員，分別包括使用者 (User)、服務伺服器(Service Host)和在驗證伺服器之內驗證模組(Authentication Module)及授權模組(Authorization Module)。本研究機制將驗證伺服器資料庫分成兩部份，一為驗證模組所屬之帳號資料庫，此資料庫僅放置使用者 ID、密碼、姓名、職生屬性等資料；另一為授權模組所屬之角色資料庫，此資料庫放置角色、權限與功能之對應資料。

本研究機制運作情境假設為使用者向服務伺服器(Host A)登入，服務伺服器(Host A)將帳號密碼向驗證伺服器之驗證模組驗證進行身分驗證。驗證伺服器確認使用者身分，若身分驗證成功，即再進行至授權模組所屬角色資料庫取得該使用者之角色集合並回傳至服務伺服器(Host A)，透過此角色集合至進行存取控制。本研究機制架構與流程如圖 4-1 所示：

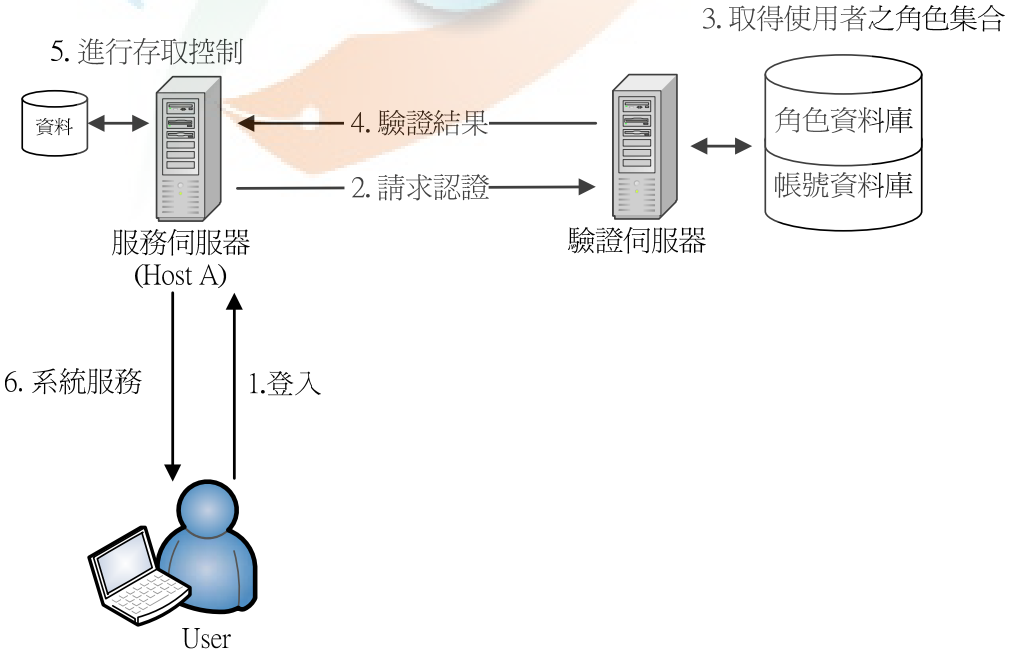

圖 4-1 本研究架構流程圖

4.2 系統運作階段

本研究機制共分為5個階段，分別是。論文的架構與相關細節分述如后：

1. 註冊階段

服務伺服器(Host A)的系統管理員，透過驗證伺服器之授權模組管理介面進行註冊與授權角色資料庫運作，對於學員生使用者資料新增於帳號資料庫，並設定角色給該使用者；對於權限異動的資料，則修改該學員生於角色資料庫；對於學員生離職或畢業的資料，則註記該學員生於帳號資料庫。

2. 服務請求階段

使用者向服務伺服器(Host A)提出服務請求，服務伺服器(Host A)將使用者的帳密作一次單向雜湊函數後，向驗證伺服器提出服務需求，以進行身分驗證及取得權限集合。

3. 驗證階段

驗證伺服器之驗證模組於收到服務伺服器(Host A) 帳密，會進行比對驗證確認使用者，是否為該服務伺服器(Host A)使用成員之一，驗證成功後繼而進入角色取得階段。

4. 角色取得階段

以該帳號尋找於服務伺服器(Host A)所授予角色，集合各角色權限，並回傳給服務伺服器(Host A)。

5. 進行存取控制階段

服務伺服器(Host A)將依該使用者帳號之角色權限進行存取控制。

4.3 RBAC 管理系統

本研究機制提供建議之 RBAC 授權管理系統，透過資料庫與系統功能來完成建置 RBAC 授權管理系統，採用統一介面來管理與設定授權中心與各服務伺服器之使用者、角色、權限與功能資料。

4.3.1 資料庫規劃

本研究機制將授權資料庫分成授權中心所屬之角色資料庫（Role based DB on Authorization Center）與服務伺服器所屬之角色資料庫（Role based DB on Service Server）。授權中心所屬之角色資料庫將存放使用者、單位、角色資料；服務伺服器所屬之角色資料庫將存放角色、權限與功能資料。本研究機制之 ER-Diagram ，請參考圖 4-2：

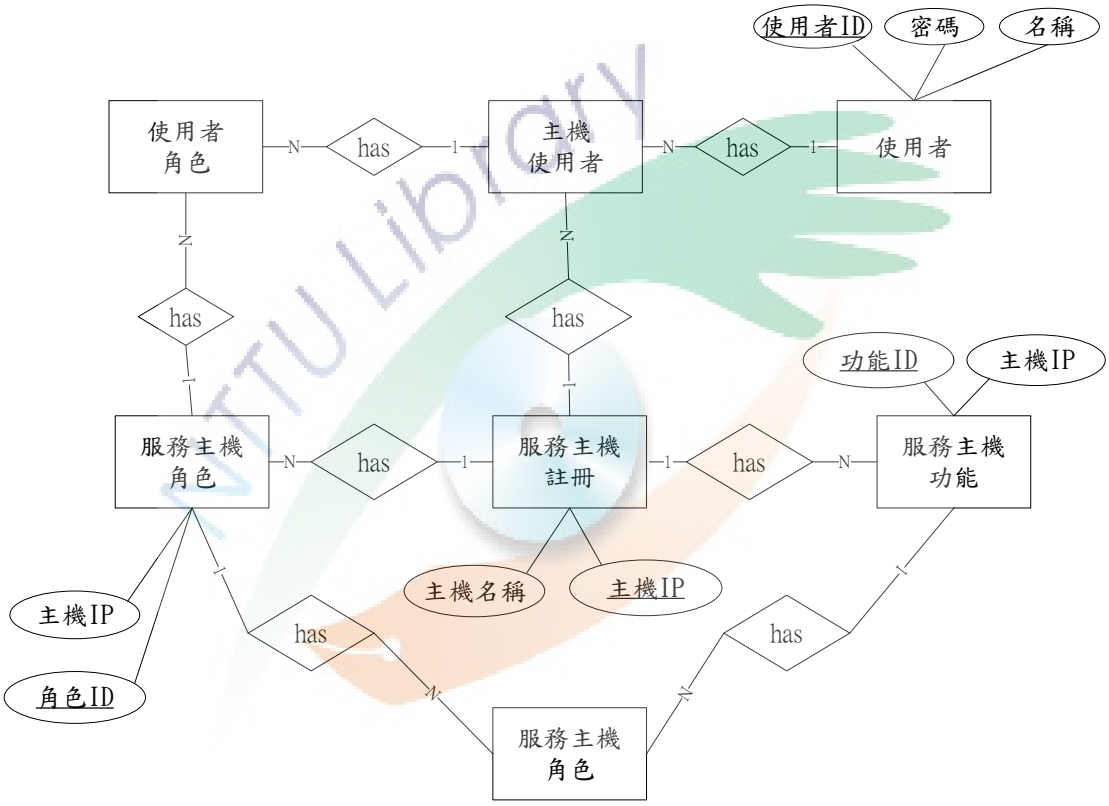

圖 4-2 資料庫 ER-Diagram

4.3.2 系統雛型

本研究依所建議之 RBAC 授權管理系統，進行系統雛型設計，並提供「登入畫面」、「Host 帳號管理」、「Host 功能管理」、「Host 角色管理」、「角色功能

管理」、「使用者基本資料」、「Host 使用者範圍」與「使用者角色資料管理」之畫面，並以情境模擬說明:

假設理工學院要辦理產學講座，對象為院內教職員生，希望能透過院內首頁，進行報名以做統計，長遠更希望建置參與講座履歷，以往建置流程首要就須建立教職員帳號密碼，讓各使用者逕自報名，並於與會當日現場簽到，在進行人工彙整，成效往往不彰，且對院內教職員生多了一組帳號密碼的管理負擔，若開發管理人員稍有不慎，將易使個資洩漏，註冊加入本架構服務，將可省卻此建置步驟及降低資安風險。

首先，院內首頁主機(服務伺服器 Service Host) 向驗證伺服器之授權管理系統申請註冊得到一組密碼，繼而登入系統如下

1. 登入畫面：

提供給各系統管理者進行登入 RBAC 授權管理系統，如圖 4-3 所示：

IP 須與該服務伺服器相吻合，輸入密碼即可登入。

圖 4-3 登入畫面

2. Host 帳號管理

授權管理系統管理者在此建立各服務伺服器(Service Host)資料，包括 IP、密碼等，如圖 4-4。每個服務伺服器登入時，將只編修自己資料，院內首頁主機名稱為理工學院 WEB 主機。

圖 4-4 Host 帳號管理

3. Host 功能管理

理工學院 WEB 主機將訂定幾個子系統及其功能，如公告發行、讀取、刪除等功能，如圖 4-5。

圖 4-5 Host 功能管理

4. Host 角色管理

理工學院 WEB 主機將訂定幾個子系統管理角色，如圖 4-6，以為授權管理。

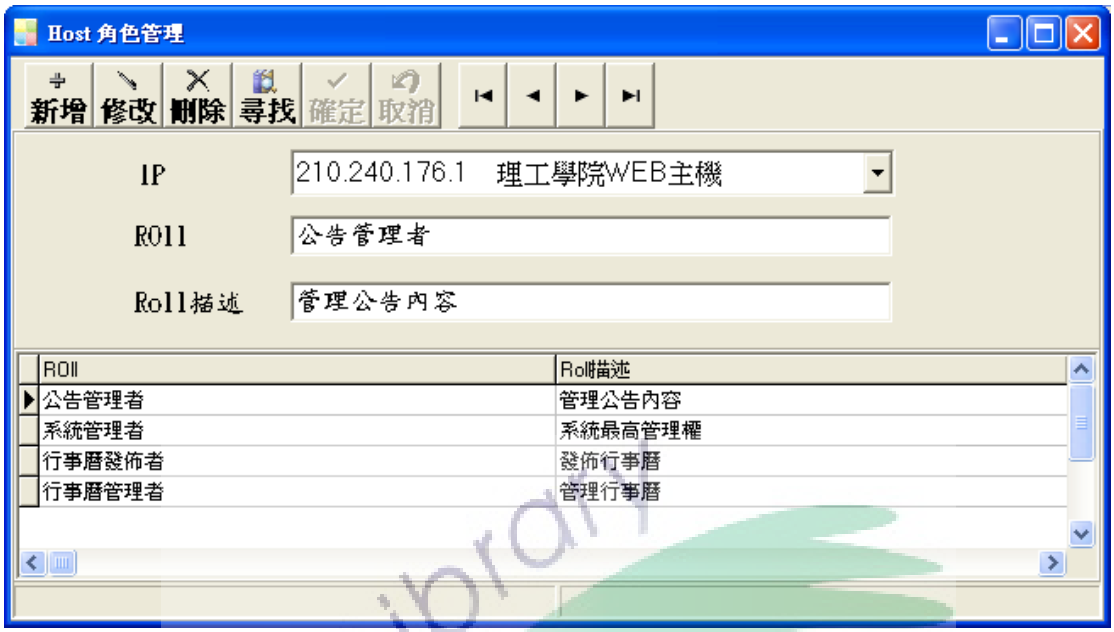

圖 4-6 Host 帳號管理

5. Host 角色功能管理

理工學院 WEB 主機角色功能管理由角色面來看，分別將先前所建立的角色，逐一賦予其功能，如圖 4-7。

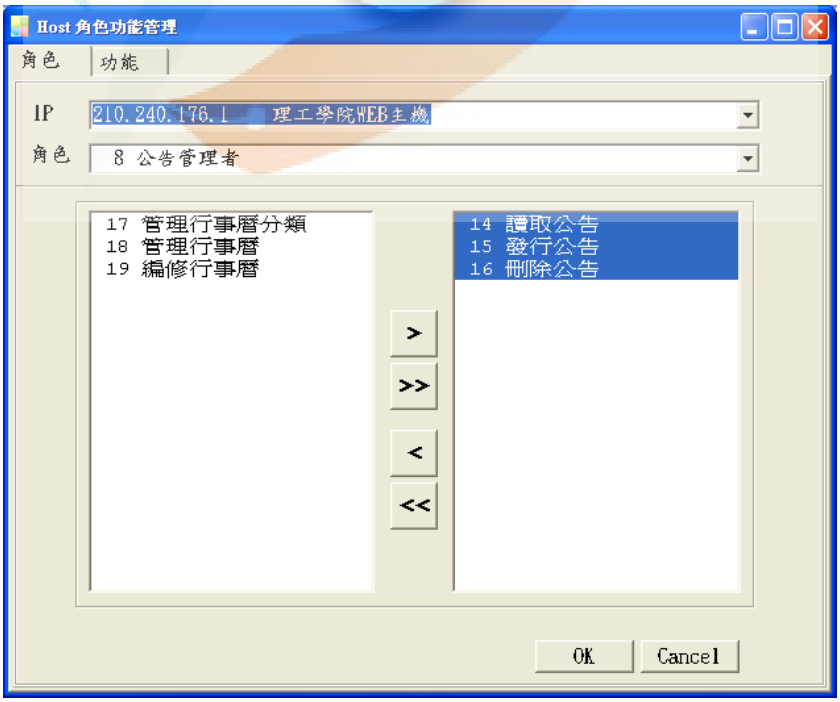

圖 4-7 Host 角色功能管理(角色)

反之由功能面來看，具有刪除公告權限者，將只有公告管理者及系統管理者，如圖 4-8。

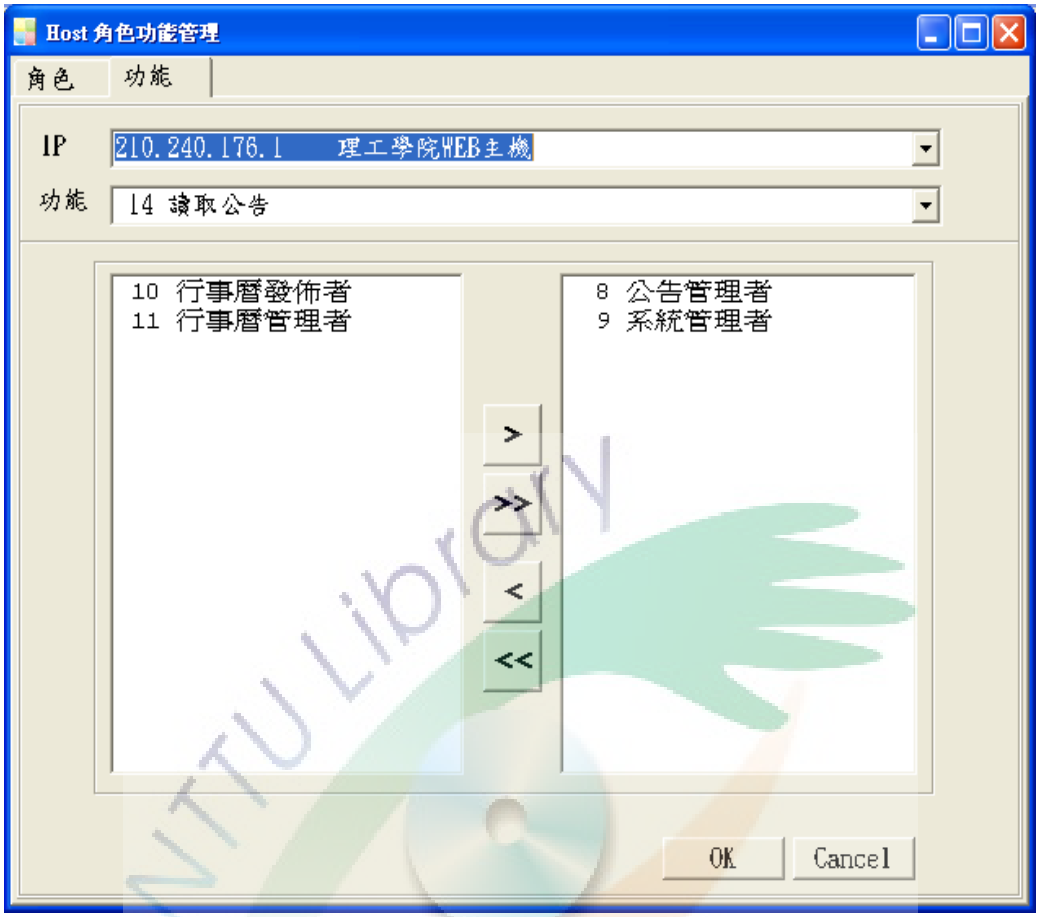

圖 4-8 Host 角色功能管理(功能)

6. 使用者基本資料

RBAC 管理者使用，其內資料含括所有系統之帳號、密碼、姓名及學生或教職員屬性，各服務伺服器(Service Host)無法進入，如圖 4-9。

7. Host 使用者範圍

RBAC 管理者使用，其內資料為定義各服務伺服器(Service Host)之教職員生使用者。理工學院 WEB 主機當選取院內教師及學生為基本使用者，，如圖 4-10。

圖 4-9 使用者基本資料

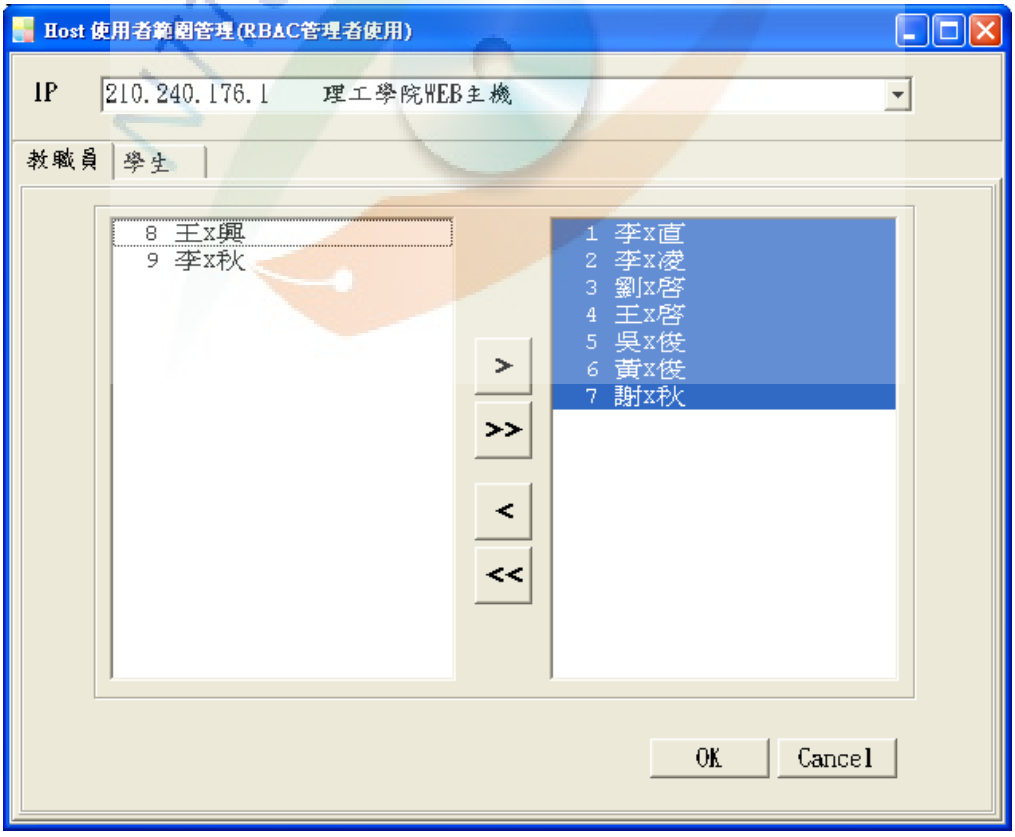

圖 4-10 Host 使用者範圍

8. 使用者角色資料管理

理工學院 WEB 主機以不同面向，分別將前述角色、功能，指派於教職員及學生，如圖 4-11，將管理公告內容之權，附予兩位代表學生。

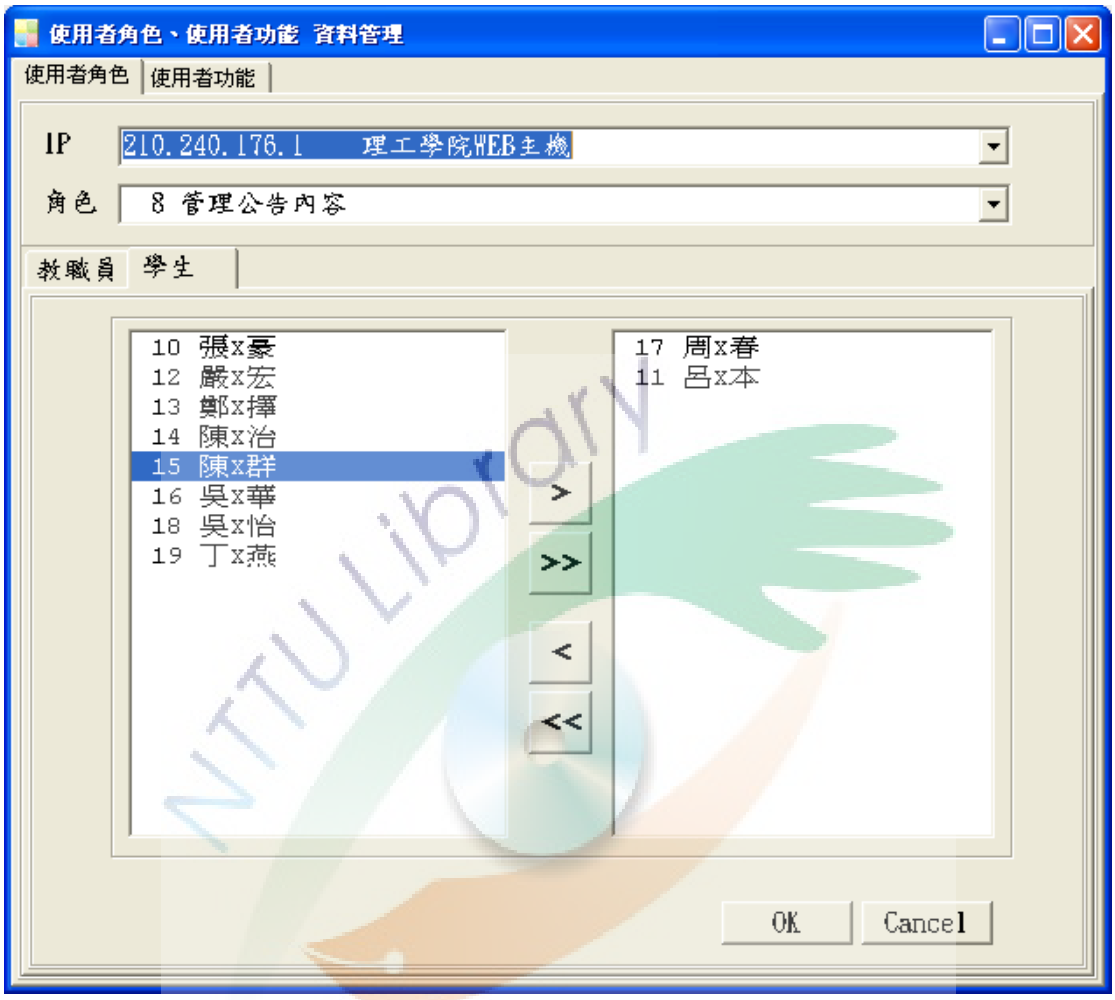

圖 4-11 使用者角色資料管理

4.4 服務請求雛型

在開發系統過程中，透過服務所在 URL 位置，直接引用該服務之 WSDL，即可直接呼叫系統所提供之驗證、授權服務

第五章 研究結論與未來展望

本章節針對研究過程提出結論，以及探討未來的研究方向。

5.1 研究結論

近年來，網際網路的快速且多元化的發展，讓網路購物、電子拍賣、線上社群與企業入口網站…等網際網路服務變得相當的盛行，也愈來愈多的組織利用網際網路的技術來建置企業內外相關網站。本研究機制的運作，免除需要記憶多組帳號的困擾與安全上的風險。同時，以角色為基礎的存取控制（RBAC）提供有效控管、簡化管理與易於維護之特性，將提升存取控制的管理效率。

每一套系統各自擁有獨立的資料庫及使用者驗證機制，如使用者想要進入任一套系統時，即必須記住該系統之帳號與密碼，經通過該系統之身分驗證後，才得以進入該系統。以使用者的角度而言，使用多套系統就必需記住多組帳號與密碼，將造成不便與困擾；以管理者的角度而言，各系統擁有所屬的資料庫與驗證機制，將造成管理上的困難。本研究機制在兼顧使用者經驗、安全性及易於部署的考量下，在安全性方面，採用非對稱式加解密、單向雜湊函數來強化系統之安全性；在使用者驗證方面，採集中式驗證之模式；在平台限制方面，則不受系統平台限制。

所有使用者和網站及網站和網站之間的資料傳遞方式均以 HTTP 通訊訊協定傳輸，透過使用者、服務伺服器、驗證模組、授權模組四種角色與註冊階段、服務請求階段、驗證階段、角色取得階段及進行存取控制階段五個階段來完成集中與 RBAC 之授權機制。

以角色為基礎的存取控制（RBAC）是比較新的存取控制策略，利用角色的概念以作為授權的依據，如在使用者與權限之間加入了角色，讓使用者與權限變成是間接的關係，再透過角色來達到使用者與權限的關連性。以角色為基礎的存取控制（RBAC)可以簡化管理者的管理程序，以變得簡便且具有彈性，如把角色賦予給使用者，再把權限賦予給角色來執行，當使用者的職務或是任務有所改變的

時候，只需變更賦予給使用者的角色，就可以改變此使用者所可以執行的權限，讓權限的管理更為的簡單，另外，角色可以對應到組織的職務，讓權限的控管可以更接近實際企業運作的安全需求。

5.2 未來研究方向

未來方向，建議可朝向下列方向進行：

(1) 由於單一帳密若不慎遭偽騙竊取，損失風險將提高，建議考量結合實體自然人憑證之輔證擴及使用端，以確保使用者身分。

(2) Web Services 跨平台技術未來將可應用在更多領域，以及提供更多元的服務項目，將是另一項值得更深入探討的主題。

參考文獻

■ 中文部分

[1]比爾・蓋茲(1999)。數位神經系統。(樂為良譯，1999)。台北市，商周出版。

[2]王昭嵐、朱斌妤，“大學校務行政電腦化推行概況與問題實證研究：行政人員電腦知能與態度分析”，中華管理評論，Vol.3, No.2, pp.135?150，2000。

[3]許建隆、楊松諺，「密碼學實務」，初版，台北，碁峰資訊股份有限公司，2005 年。

[4]簡西村,"MSDN 服務導向架構(Service Oriented Architecture)應用專欄",http://www.microsoft.com/taiwan/msdn/columns/soa/SOA_overview_2004112901.htm，2004

[5] 戚玉樑、彭淑芸、張琪瑩與賴德優，Web Services 探索與應用，台灣台北，全華科技，民國九十二年五月

[6]謝文全(2002)。學校行政(八版一刷)。台北市，五南。

■ 英文部分

[1] Ben Margolis ,"SOA for the Business Developer: Concepts, BPEL , and SCA","Texas : MC Press Online LP",2007.

[2] David Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn and Ramaswamy Chandramouli, “Proposed NIST Standard for Role-BasedAccess Control”, ACM Transactions on Information and System Security, Vol. 4, No. 3, August 2001, 224–274

[3] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn and R. Chandramouli, “Proposed NIST Standard for Role-Based Access Control,” *ACM Transactionson Information and System Security*, vol. 4, no. 3, pp. 224-274, 2001.

[4] MacKenzie, C. Matthew., et al.(2006), “Reference Model for Service

Oriented Architecture 1.0”, OASIS Public Documents

[5] Mary Kirtland, “A Platform for Web Services”,(Microsoft Developer Network), January 2001

[6] Michael Huhns,and Munindar P. Singh,"Service-Oriented Computing:Key Concepts and Principles",IEEE Internet Computing,Vol. 9, No. 1,JAN/FEB 2005

[7] Nicolai Josuttis,"SOA in Practice",O'Reilly, 2007

[8] Norbert Bieberstein and Sanjay Bose and Marc Fiammante and Keith Jones and Rawn Shah, "Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap",(Hardcover - Nov 4, 2005)

[9]R. S. Sandhu, E. J. Coyne and C. E. Youman, “Role-Based Access Control Models,” *IEEE Computer*, pp. 38-47, 1996.

[10] Thomas Erl. "Service-Oriented Architecture: Concepts, Technology, and Design. ", "Prentice Hall PTR", 2005.