

國立交通大學
工業工程與管理學系

碩士論文

網路服務與射頻辨識導向的
花卉追溯系統設計與實作

Design and Implementation of a Web Service and RFID
Based Flower Traceability System



研究生：林新凱

指導教授：梁高榮博士

中華民國九十六年七月

網路服務與射頻辨識導向的花卉追溯系統設計與實作

Design and Implementation of a Web Service and RFID

Based Flower Traceability System

研 究 生：林新凱

Student：Shin-Kai Lin

指 導 教 授：梁高榮

Advisor：Gau-Rong Liang



A Thesis Submitted to
Department of Industrial Engineering and Management
College of Management
National Chiao Tung University
In Partial Fulfillment of the Requirements
For the Degree of Master of Engineering in
Industrial Engineering and Management
June 2007
Hsinchu, Taiwan, Republic of China

研究生：林新凱

指導教授：梁高榮博士

國立交通大學工業工程與管理學系

中文摘要

本研究提出一套以網路服務為架構的模型來建構花卉追溯系統。在此架構之下，可以透過服務仲介者、服務提供者與服務需求者來追溯花卉商品的相關資訊。在此，服務的提供者為花卉供應鏈中的生產者、生產合作社、批發市場、承銷人或是零售商，而服務的需求者則是對於所有能夠追溯到的花卉資訊有興趣的需求者。服務仲介者是一個公開的網際網路導向系統，因此所有的服務提供者可以透過網路服務描述語言註冊它們的服務在仲介端之上，藉由這個方式來發佈它們的服務，而網路服務的內含為取得該服務提供者的花卉資訊。在這樣的環境下，任何的服務需求者能夠向服務仲介者查詢它們所需的服務，服務仲介者則回傳該服務的提供者資料，透過與服務提供者的溝通則可取得所需的花卉資訊。因此，本研究的成果在於設計與實作一套花卉資訊追溯系統的雛型。

關鍵詞：

網路服務

花卉追溯系統

網路服務描述語言

Design and Implementation of a Web Service and RFID Based Flower Traceability System

Student : Shin-Kae Lin

Advisor : Dr.Gau-Rong Liang

Department of Institute of Industrial Engineering & Management
National Chiao Tung University

Abstract

A web services framework has been proposed for constructing a flower traceability system. In this framework, flowers are traced among a service broker, service providers, and service requesters. Here a service provider may be a grower, a producers' cooperative, a wholesale market, a consignee, or a retailer in flower supply chains, and a service requester is anyone who is interested in tracing flowers. A service broker is an Internet-based computer so that all service providers can submit their own services based on Web Service Definition Language (WSDL) for tracing flowers. In operations, any service requester can ask a service from the service broker. Then the service broker is going to reply a list of service providers to the service requester who therefore retrieves the requested service from each service provider. As a research result, a prototype traceability system for flowers has been designed and implemented.

Keywords :

Web Services

Flower Traceability System

Web Service Definition Language (WSDL)

誌謝

本論文能順利完成，首先要感謝指導教授梁高榮教授的費心指導，不論是在研究的觀念上與系統的架構上，梁老師總是不辭辛勞的指導，使我獲益斐淺。此外還要特別感謝唐麗英老師與張永佳老師及洪一薰老師對本論文所提供的寶貴建議，使本論文的內容更加充實。

此外還要特別感謝高雄批發市場對本研究資料收集階段中解答許多問題，並安排許多實際參觀批發市場拍賣流程，讓本研究可以實際觀察拍賣現況。另外感謝學長小胖跟王炯棠對本研究給予許多的幫助，讓本研究可以順利完成。

在這兩年的日子中首先要感謝梁高榮教授在這段時間的照顧及關心。另外還要感謝實驗室的學長姐，小胖、小班、小先，以及同學，BE、老王、阿端、阿牛，以及學弟妹，音帆、Mia、粗皮、家宇，以及所有我身邊的朋友及家人，在研究所這段期間的陪伴與幫助。

本論文在撰寫期間曾參與漁業署的研究計畫「RFID 技術在漁業上的加值應用研究」，計畫編號為「94I812」、「漁產品產銷履歷資訊建置與整合」，計畫編號為「94I825」及農糧署的研究計畫「建置高雄花卉批發市場資料倉儲與大型冷藏庫庫存管理系統」，計畫編號為「95I510」，接受支助，特此感謝。



目錄

中文摘要.....	i
Abstract.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vii
表目錄.....	ix
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 問題界定.....	3
1.3 研究目的.....	4
1.4 系統架構.....	6
1.5 研究方法.....	8
1.6 論文架構.....	10
第二章 文獻回顧.....	11
2.1 花卉運銷體系.....	11
2.2 射頻識別技術.....	12
2.2.1 射頻識別技術標準.....	12
2.2.2 射頻識別標籤、標牌與編碼.....	14
2.2.3 射頻識別讀寫器.....	16
2.2.4 中介軟體架構與資訊服務系統.....	17
2.2.5 射頻識別技術的困境.....	18
2.3 網路服務技術.....	19
2.3.1 歷史探究.....	19
2.3.2 服務導向架構.....	19
2.3.3 可擴展標記語言.....	20
2.3.4 簡單物件存取協定.....	22
2.3.5 網路服務描述語言.....	24
2.3.6 通用描述、發現與整合.....	26
2.4 追溯系統.....	29
第三章 花卉供應鏈資訊儲存架構設計.....	31
3.1 花卉供應鏈運銷.....	31
3.1.1 產地的花卉貯存與運送.....	31

3.1.2 花卉批發市場作業流程.....	31
3.1.2.1 花卉批發市場的進貨流程分析.....	31
3.1.2.2 花卉批發市場的理貨流程分析.....	32
3.1.2.3 花卉批發市場的拍賣流程分析.....	32
3.1.2.4 花卉批發市場的分貨流程分析.....	35
3.1.2.5 花卉批發市場的領貨流程分析.....	35
3.1.3 零售據點的花卉貯存與運送.....	35
3.2 花卉批發市場資訊儲存架構設計.....	36
3.2.1 交易資料介紹.....	36
3.2.2 檔案轉移服務.....	37
3.2.3 三階正規化設計.....	39
3.3 供應商與承銷商的資訊儲存架構設計.....	40
3.4 花卉識別 Index 的設計.....	42
第四章 網路服務仲介端與伺服器端的設計	44
4.1 網路服務導向追溯系統的架構.....	44
4.2 網路服務仲介端的建置.....	46
4.3 Axis 的系統架構與運作模式.....	48
4.4 網路服務伺服器端的建置.....	50
4.4.1 伺服器端的建置流程.....	50
4.4.2 網路服務的內涵說明.....	51
第五章 網路服務客戶端的設計	57
5.1 射頻辨識讀寫功能的設計.....	57
5.1.1 射頻辨識硬體架構.....	57
5.1.2 射頻辨識軟體架構.....	60
5.2 追溯系統客戶端與仲介端溝通設計.....	62
5.3 追溯系統客戶端與伺服器端溝通設計.....	64
5.4 追溯系統客戶端整合入口的操作.....	66
第六章 結論與未來方向	69
6.1 結論與研究貢獻.....	69
6.2 未來發展方向與建議.....	71
參考文獻.....	72
附錄一 資料檔格式說明	73
附錄二 資料庫中資料表的欄位屬性說明	75
附錄三 網路服務伺服器端程式說明	78

附錄四 網路服務客戶端程式說明	81
附錄五 資料轉換服務	85
附錄 5.1 資料轉換服務 1.....	85
附錄 5.2 資料轉換服務 2.....	87
附錄 5.3 自動化排程.....	89



圖目錄

圖 1.1	荷蘭花荷拍賣場 RFID 使用狀況	2
圖 1.2	系統的可追蹤性與可追溯性	4
圖 1.3	網路服務架構圖	5
圖 1.4	追溯系統架構圖	6
圖 1.5	研究方法架構	9
圖 1.6	論文架構圖	10
圖 2.1	花卉運銷體系的關係圖	11
圖 2.2	EPC Network 架構圖	13
圖 2.3	顯示輸出和文件轉換架構	22
圖 2.4	SOAP Message 架構圖	23
圖 2.5	網路服務描述架構與網路服務描述語言對應圖	25
圖 2.6	重要資料節點的對應關係	27
圖 3.1	花市運送花卉貨車進卸貨過程	32
圖 3.2	彰化花市 LED 式拍賣鐘系統運作狀況	33
圖 3.3	高雄花市單槍投影式拍賣鐘系統運作狀況	33
圖 3.4	承銷人拍賣人機介面	34
圖 3.5	花卉拍賣資訊傳送過程	34
圖 3.6	自動分貨系統	35
圖 3.7	資訊轉換系統架構	36
圖 3.8	轉檔軟體	36
圖 3.9	交易資料原始檔案	37
圖 3.10	檔案轉移服務視窗	37
圖 3.11	與伺服器進行連線成功後視窗	38
圖 3.12	自動化設定視窗	38
圖 3.13	檔案上傳成功畫面	38
圖 3.14	花卉批發市場三階正規化資料庫架構	39
圖 3.15	供應商資料儲存架構關聯圖	40
圖 3.16	承銷商資料儲存架構關聯圖	41
圖 3.17	供應鏈的 Tradeoff	42
圖 3.18	批發市場拍賣序號識別碼	42
圖 3.20	建議的承銷商進貨編號識別碼	43
圖 3.21	標籤內所存取的條碼資訊	43
圖 4.1	追溯系統架構	44
圖 4.2	Microsoft UDDI 建置 Server 首頁	46
圖 4.3	UDDI 發行網路服務完成頁面	47
圖 4.4	Axis 用於伺服端的運作架構	48
圖 4.5	Axis 用於服務需求端的運作架構	49
圖 4.6	利用 AdminClient 執行 deploy.wsdd	50

圖 4.7	服務佈署成功驗證畫面.....	51
圖 4.8	批發市場網路服務內容程式與批發市場資料庫對應.....	52
圖 4.9	供應商網路服務內容程式與供應商日誌資料庫對應.....	53
圖 4.10	承銷商網路服務內容程式與承銷商日誌資料庫對應.....	53
圖 4.11	供應商提供的網路服務 WSDL 檔.....	54
圖 4.12	高雄批發市場提供的網路服務 WSDL 檔.....	55
圖 4.13	承銷商提供的網路服務 WSDL 檔.....	56
圖 5.1	RFID 系統架構圖.....	57
圖 5.2	IF5 射頻辨識讀取器.....	58
圖 5.3	射頻識別標籤規格.....	59
圖 5.4	射頻識別標籤內容.....	59
圖 5.5	追溯系統客戶端與射頻辨識讀取器溝通架構.....	61
圖 5.6	追溯系統客戶端 Socket 程式碼.....	61
圖 5.7	服務需求端與服務仲介端互動關係.....	62
圖 5.8	使用 UDDI4J 套件取得服務需求端所需資訊的程式說明.....	63
圖 5.9	客戶端與伺服器端溝通的程式碼.....	64
圖 5.10	批發市場 WSDL 的 getAucData 服務.....	65
圖 5.11	追溯系統客戶端程式介面.....	66
圖 5.12	追溯系統客戶端操作步驟圖.....	67
圖 5.13	追溯系統客戶端查詢結果頁面.....	67
附錄圖 1	網路服務程式碼流程.....	78
附錄圖 2	UDDI4J 程式流程.....	82
附錄圖 3	客戶端與伺服器端溝通的程式碼流程.....	84
附錄圖 4	供應人、承銷人及花卉資料的資料轉換服務流程.....	85
附錄圖 5	資料淨化.....	86
附錄圖 6	花卉批發市場交易資料轉換服務流程.....	87
附錄圖 7	系統更新轉換程序.....	89
附錄圖 8	資料轉換服務作業樹狀結構圖.....	90
附錄圖 9	自動化設定排程語法.....	90

表目錄

表 1.1	一維條碼與 RFID 之比較.....	4
表 1.2	追溯系統服務提供者註冊與佈署資訊.....	7
表 2.1	ISO14443 及 ISO15693 的特性比較.....	13
表 2.2	ISO18000 系列標準.....	14
表 2.3	射頻識別標籤的記憶體讀寫功能.....	15
表 2.4	射頻識別的頻率.....	15
表 2.5	第 1 級標籤(96 位元)的資料格式.....	16
表 2.6	Ubiquitous ID Center 的 ucode 資料基本結構.....	16
表 2.7	中介軟體的特色.....	18
表 2.8	SOAP 相關規範.....	23
表 2.9	SOAP Header 的屬性值.....	24
表 2.10	網路服務描述語言的主要元素.....	25
表 2.11	資料結構模型的元素.....	27
表 2.12	資料結構模型的屬性.....	28
表 2.13	可追蹤性之定義.....	29
表 3.1	供應商日誌基本資料表.....	40
表 3.2	供應商施肥紀錄表.....	40
表 3.3	肥料種類資料表.....	41
表 3.4	承銷商日誌基本資料表.....	41
表 3.5	承銷商日誌基本資料表.....	41
表 4.1	網路服務註冊資料.....	47
表 4.2	WSDL 與 Java 的伺服器端繫結規則.....	49
表 4.3	WSDL 與 Java 的客戶端繫結規則.....	49
表 4.4	所有網路服務內容整合.....	51
表 4.5	WSDL 文件設計要點與對應元素.....	53
表 5.1	IF5 讀取器的規格及支援性.....	58
表 5.2	BRI 的指令與說明.....	60
表 5.3	UDDI4J 主要 API 結構與功能對照表.....	63
表 5.4	追溯系統客戶端查詢結果表.....	68
附錄表 1	交易資料檔格式說明：(檔名範例 A.TXT).....	73
附錄表 2	供應人資料檔格式說明：(檔名範例 SU.TXT).....	73
附錄表 3	花卉品名資料檔格式說明：(檔名範例 PR.TXT).....	73
附錄表 4	承銷人資料檔格式說明：(檔名範例 BU.TXT).....	74
附錄表 5	拍賣資料表.....	75
附錄表 6	供應人資料表.....	75
附錄表 7	供應行政區資料表.....	76
附錄表 8	供應縣市資料表.....	76
附錄表 9	承銷人資料表.....	76

附錄表 10 承銷區域資料表.....	76
附錄表 11 花卉資料表.....	76
附錄表 12 花卉等級表.....	76
附錄表 13 花卉等級略表.....	77
附錄表 14 拍賣線資料表.....	77
附錄表 15 網路服務內容程式碼.....	78
附錄表 16 佈署服務程式的程式碼(Deploy.wsdd).....	80
附錄表 17 佈署服務程式的程式碼(Undeploy.wsdd).....	80
附錄表 18 客戶端透過 Socket 與 RFID 讀取器溝通的程式碼.....	81
附錄表 19 客戶端與仲介端溝通的 UDDI4J 程式碼.....	81
附錄表 20 客戶端與伺服器端溝通的程式碼.....	82
附錄表 21 新增、修改、停用判斷之程式碼與說明.....	87
附錄表 22 交易資料修正.....	88
附錄表 23 交易資料與供應商、承銷人、花卉品名資料表的比對.....	89
附錄表 24 資料轉換服務工作統計表.....	90



第一章 緒論

本章的目的是說明本論文的動機、研究內容、架構及目標，分六部份做介紹，分別為第 1.1 節「研究動機」，第 1.2 節「問題界定」，第 1.3 節「研究目的」，第 1.4 節「系統架構」，第 1.5 節「研究方法」，第 1.6 節「論文架構」。

1.1 研究動機

花卉生長之特性極易受外在環境的影響，且產品之脆弱性及損耗率均較其他農產品為甚。因此要達到花卉商品化程度，基本條件不外乎包括專業化經營，藉生產技術之提升及適當之田間管理以生產高品質之產品，同時並維持高品質的貯存與運送，方能確保商品價值。

除了生產、運送部份，銷售也是花卉商品化重要的一環。花卉公司、供應人與承銷人三者間活絡的互動關係造就花卉市場旺盛的生氣與活力，連帶促使花卉交易資訊的產生、處理、傳播、分析等受到國人的重視，且成為國內花卉產業重要的經濟性活動。目前花卉供應鏈中各端點的資訊並沒有完全透明化，各端點的使用者無法對其進行直接的追蹤(Tracking)與追溯(Tracing)，因此本研究建立一套追溯系統將供應鏈上各端點的資料透過使用者圖形介面呈現出來。

透過追溯系統，在生產、運送部份可提升品質保證、降低風險、提升供應鏈整體效益。在銷售部份，承銷人可藉追溯歷史資料傳達出過去供應人供貨的統計資料供做參考。供應人則可以依據承銷人需求進行線上供貨，供銷雙方的交易是直接、透明的，同時大量、長期的線上下單採購過程，所降低的交易成本與營運管銷費用，更是供銷雙方直接獲利的最大來源。不僅如此，顧客在向承銷人購買花卉時，可透過追溯系統得到過去花卉交易資訊，提升顧客購買欲望及選擇的多樣性。

可追溯性是藉由記載的識別證明進行追溯歷史、應用或是實體位置的能力，並提供了產品所在地與相關處理之狀態資訊能見性。除了能見性外，一個線上追溯功能創造了允許追溯元件與處理最終產品之重要歷史的記載[26]。本追溯系統除了具有追溯功能，也帶有追蹤的功能。而追溯系統的導入受到產業類別等因素的影響，對花卉供應鏈而言，建立追溯系統有兩個關鍵因素，分別為產品資訊傳遞技術與跨組織的整合資訊分享架構。

在產品資訊傳遞技術方面，每一個產品都有其單一的識別碼，而且每個識別碼存在其唯一性，過去在載具方面採用的是一維條碼，一維條碼受環境、距離等因素的限制。例如台北花市目前條碼標籤每日的使用量，切花約 5000 張、盆花約 300 張，條碼標籤機燙頭與標籤紙的損耗相當高，不僅市場成本較大，在損耗同時也喪失了該筆資料內容，追溯系統便無法追溯到其資料。除此之外一維條碼資訊密度較低、資訊容量較小、沒有錯誤糾正能力、保密防偽性較差及使用可靠性較差，同時讀取數量一次只能一個、讀取距離也較短，在運送與讀取過程中有其不方便性。

為克服這些缺點，本研究嘗試將射頻辨識技術(Radio Frequency Identification Technology, RFID Technology)[1][18]導入花卉供應鏈中，以射頻識別標籤取代傳統的一維條碼。射頻識別標籤有著高度的環境忍耐力及重覆的讀寫率。除此之外由於技術的成熟，射頻辨識技術的標籤、讀取器等設備價格不再像過去如此昂貴，對於高單價的花卉產品成為超值的載具。射頻識別技術更可以透過無線通訊，一次將多個識別辨識標籤自動讀取完成，幾乎不需借助人力，

提升作業效率、減少人力支出。

近年來射頻辨識技術的成熟與商品化，由於美國 Wal-Mart 零售系統要求其百大供應商必須全面將商品貼上 RFID 電子標籤，目前所有的 Wal-Mart 供應商都採用射頻辨識技術的方式出貨。據 Wal-Mart 內部評估，Wal-Mart 在一年內省下的成本約高達 83.5 億美元。市場研究機構 Gartner 的報告也指出，相關硬體及軟體的建置將會在 2007 年加速發展，預估到了 2010 年 RFID 的全球支出金額可達 30 億美元。此外，美國市場研究機構 AMR Research 亦預測 RFID 在 2013 年產值可達 \$210 億美元，年成長率約 44.2% [5]。射頻辨識技術不僅是提供一個能夠改善供應鏈管理與大幅降低運作成本的一種獨特物件識別方式，更能夠結合其他檢測與網路技術來追溯物件與自然環境，產生一種更靈敏的電子化架構。利用射頻辨識技術提昇企業競爭力已成為全球主要產業的主流，並視為本世紀最重要的前十大技術。

荷蘭花卉拍賣場(Bloemenveilingen Flora Holland)是目前全世界最大的花卉拍賣場，年營業額超過 20 億歐元，每天都會有大約四千個承銷商從世界各地來花卉拍賣場使用 26 個拍賣鐘進行拍賣。荷蘭花卉拍賣場自 1996 年也開始導入射頻辨識技術於花朵和拍賣車管理，在拖車底部與花卉拍賣盤上裝有射頻辨識標籤。荷蘭花卉拍賣場將射頻辨識技術應用於四個部份，包括在拍賣場中 30 條必經之路中獲得標籤資訊、從拖車(Trolley)中獲得標籤資訊、解決供應鏈中瓶頸問題、保證花卉的新鮮度 [20]。圖 1.1 是目前荷蘭花卉拍賣場使用 RFID 的狀況。

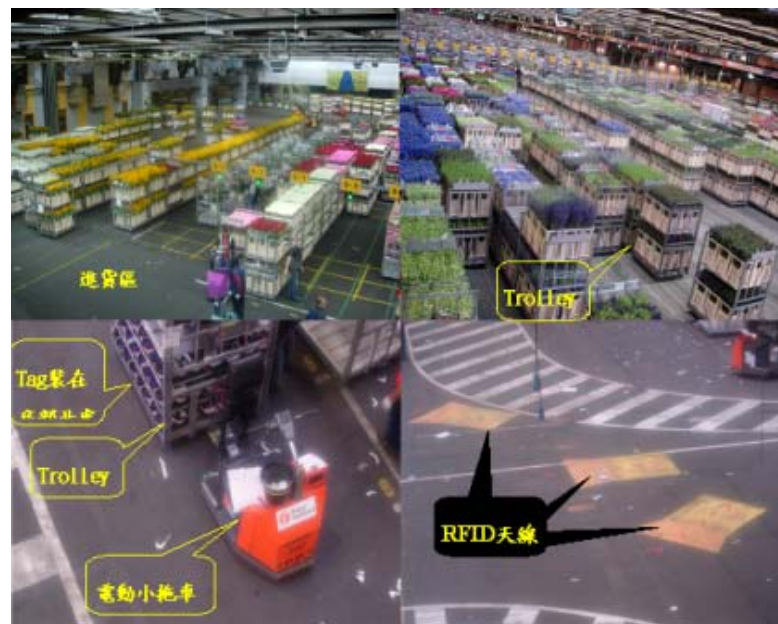


圖 1.1 荷蘭花荷拍賣場 RFID 使用狀況 [20]

追溯系統的跨組織的整合資訊分享架構部份，由於資料分散於供應鏈各個節點之中，追溯系統必須得到各作業節點中的資料才能進行追蹤與追溯。對於分散式架構而言，取得與傳遞資料過程中以遠端程序呼叫(Remote Procedure Call, RPC)所進行的服務最為重要，一方面需顧及資料的安全性、傳遞的迅速，另一方面還得顧及資料的完整性。而網路服務(Web Service) [2][23] 是目前支援透過網際網路進行遠端程序呼叫最熱門的軟體服務。服務的相關細節利用網路服務描述語言來描述，透過標準的網路通訊協定，以 XML 格式組成的簡單物件存取協定訊息處理溝通(通訊底層、進行傳送的动作、向通用描述、發現與整合進行搜尋或是註冊動作)與辨認程式碼的問題，更有實現分散式架構動態整合、平衡負擔、單元升級等優點。

1.2 問題界定

臺灣地區生活水準提高，花卉需求日增，國內批發市場功能日趨重要，目前國內五座花卉批發市場採用現代化的電腦拍賣系統，大幅提升交易之公平性。然而供應商自行或透過共同運銷直接運至批發市場交易之比率，由過去之 1.5% 大幅提高為 92.6%，而直接銷售給販運商之傳統運銷方式，則相對式微。

但是舉凡批發市場的進貨、理貨作業及拍賣後之分貨、提貨作業仍需許多人力，不僅作業效率難以提昇，且提貨也容易錯誤。就進貨作業而言，進貨貨車通常都趕忙卸貨離開，點貨驗收通常都以點件為主。然而貨車卸貨不一定是按照進貨資料表進貨，而同一張進貨資料表的花卉也可能由不同的貨車運送，在處理貨物件數比對等進貨工作時往往會出差錯，造成延誤的狀況。不僅如此在點貨與搬運過程中所運用的大量人工處理，便有可能產生錯誤，例如筆誤、輸入操作失誤，特別是對於分級制度精細的花卉產業。

而花卉供應鏈載具的部分，目前搭載的載具為一維條碼(code 39)。光是台北批發市場目前每日條碼的使用量，切花約 5000 張、盆花約 300 張，在大量的應用之下產生了些許問題。由於一維條碼只能儲存 15 個文、數字，因此必須具有網路或資料庫等配合方有作用。除了儲存內容的限制，一維條碼還受限於環境與距離等因素。不單單如此，條碼標籤機燙頭與標籤紙的損害率相當高，該標籤紙有所破損時，該筆花卉商品的資料將隨之消失，再也追溯不到該筆花卉資料。此外一維條碼的資訊密度較低、沒有錯誤糾正能力、保密防偽性較差，同時讀取數量一次只能一個、讀取距離也較短，在運送與讀取過程中有其不方便性。

而資訊交換的部分，花卉供應鏈並沒有一個公開資訊讓使用者了解供應鏈中的狀況，供應鏈中的資訊並沒有透明化，無法進行追溯或追蹤等作業。舉例來說，顧客購買花卉的同時，無法得知該筆花卉出自何地、經由哪個批發市場賣給承銷商等資訊。

總結我國現行的花卉供應鏈的制度有幾項缺失，分述如下：

1. 花卉供應鏈中搭載的載具為一維條碼，產生許多的人力消耗、作業的不確定性以及各方面外在環境的限制，例如一維條碼的儲存空間較小、損毀率較高等。
2. 供應商不僅僅出貨給一家批發市場，而一家批發批發市場也不僅出貨給同一家供應商，因此同一個生產批號並不一定具有相同的拍賣序號，如果標籤中只存有單一序號，在進行追蹤或追溯時可能會發生問題。
3. 供應鏈沒有透明化，因此各端點的使用者無法追蹤及追溯到供應鏈中各層級的狀況，無法應用其資訊。
4. 承銷業者與顧客之間的需求與互動比較不常被注意。

在第 1、2 點的缺失部分，期望透過射頻辨識技術的導入，改善缺失。第 3、4 點則期望透過追溯系統的能見性，改善資訊沒有透明化的窘境。因此本研究將問題界定於花卉供應鏈中流程的改善、資料的保存以及資料交換的架構建置。

1.3 研究目的

綜合 1.2 節中所提出的問題，本研究提出一個以網路服務技術與射頻辨識技術為導向的花卉追溯系統，針對的使用對象，舉凡從供應商到批發市場業者再到承銷商甚至顧客等供應鏈上的使用者皆可使用，主要的追蹤與追溯過程如圖 1.2 所示。

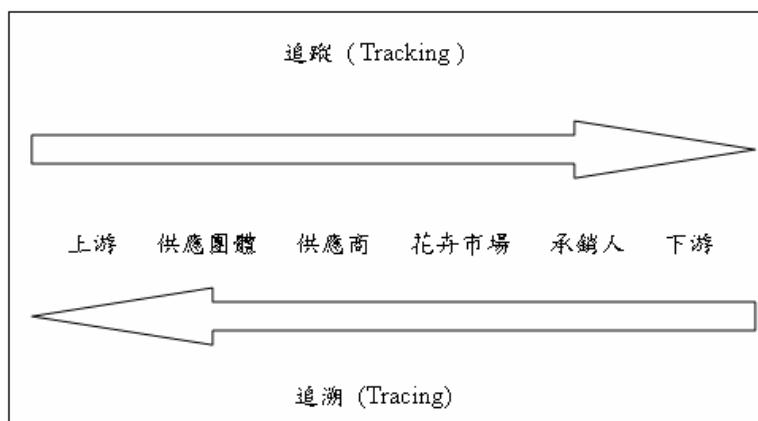


圖 1.2 系統的可追蹤性與可追溯性

提出追溯系統主要的目的為強化風險管理以及回應客戶端的資訊需求。強化風險管理的部分主要是透過系統追溯產品的製造過程及運送過程，此外透過掌握物流的動向，防止商品的遺失；回應客戶端資訊需求，是改善過去資訊封閉、供應鏈沒有透明化的窘境，透過追溯系統的追溯及追蹤功能，達到說明的責任。過去這方面的問題並沒有受到大眾的重視，但是有些資訊對於使用者是很好的參考資訊。以顧客而言，像花卉等級便是一個很重要的參考資訊，透過資訊的取得，就算是對於花卉產品知識不足的顧客也可以購買到合適的商品。

追溯系統的產品傳遞技術採用射頻辨識技術取代現存的一維條碼，主要目的是透過射頻辨識技術的優點改善一維條碼的缺點。在 1.2 節中已經簡單的說明一維條碼在花卉產業中的不足性，在表 1.1 中，說明一維條碼與 RFID 的差異。

表 1.1 一維條碼與 RFID 之比較

	一維條碼	RFID 標籤
儲存資料量	低	高
加密性	不可加密	可加密
同時讀取數量	一次只能讀取一個	可同時讀取多個
讀取方式	人力	自動
方向性	需近距離對準條碼	遠距離但會受金屬等干擾
讀寫性	唯讀	可讀可寫
對環境的忍受力	低	高

在眾多的產品資訊傳遞技術之中，採用射頻辨識技術的主要原因是 RFID 具有高度的環境抵抗力、讀取距離較遠、可同時讀取多個的優點。對追溯系統而言，高度的環境抵抗力是絕

對不可或缺的，一旦標籤損毀所帶來的問題不單只有標籤的成本，追溯系統也再也追溯不到標籤的資料，連帶著整個追溯系統呈現不完整的狀況。讀取距離遠與可同時讀取多個是射頻辨識技術最大的特點，透過射頻辨識的無線通信技術，一次能將多個標籤自動讀取完成，幾乎不需要借助人力。相較於一般條碼必須藉由人工將其置於讀取器前，一項一項進行讀取，節省大量的人力與時間，更使得錯誤率降低。不僅如此透過射頻辨識技術和其他系統的結合，例如自動搬運機，在物品的搬運可以減少許多不必要的勞力浪費。

在跨組織的整合資訊分享架構部分，完整的取得分散式架構下各節點的資料最為重要。本研究使用網路服務技術主要的目的是透過網路服務的簡易性及服務性取得完整的資料。網路服務遵循一個以服務為導向的架構，架構圖如圖 1.3。

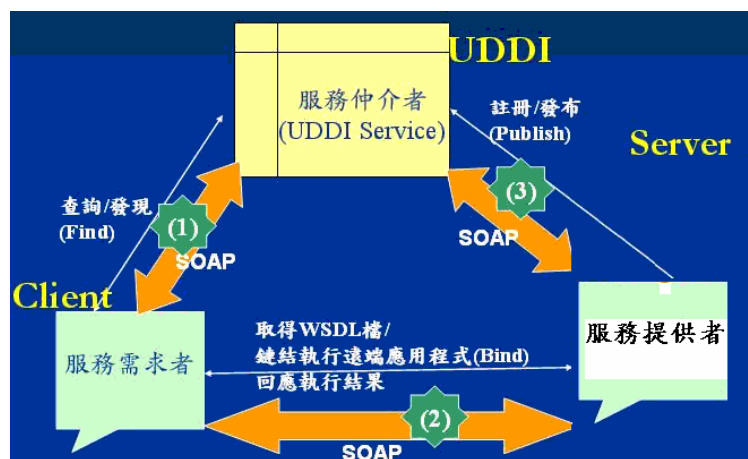


圖 1.3 網路服務架構圖

簡單來說本研究中網路服務所扮演的角色為一般三層式架構下第一層的客户端與第二層中介軟體的角色，使用者透過網路服務到資料庫查詢資料。與一般的中介軟體不同的是依服務需求者不同的參數到服務仲介者查詢所需的服務，依不同的服務會向不同的資料庫進行存取，不再是一對一的關係，讓整個系統有很大的彈性。

網路服務基本架構主要由三者組成：服務提供者(Service Provider)、服務需求者(Service Requester)、服務仲介者(Service Broker)。服務提供者透過網路服務描述語言文件(Web Services Definition Language, WSDL)[25]將提供的服務規格書註冊在服務仲介端；而服務仲介者使用通用描述、發現與整合(Universal Description, Discovery and Integration, UDDI)[22]技術，是網路服務架構中的第三方仲介，功能是告知服務需求者某項服務的提供者在網路上的正確位址；而服務需求者，透過服務名稱以簡單物件存取協定(Simple Object Access Protocol, SOAP)[24]訊息送至服務仲介端查詢適合的網路服務，再透過服務位址以及產品代碼與服務提供者所提供的服務進行溝通。

在本文中，服務提供者是供應鏈中花卉批發市場的業者、供應商與承銷商，每個業者建立屬於自己的網路服務，而服務內容是存取該業者資料庫中的日誌資訊。而服務仲介者為 UDDI 註冊中心，告知服務需求者某項服務的位址；而服務需求者，也就是供應鏈上的各端點(包括供應商、批發市場、承銷商、顧客等)。

1.4 系統架構

在本節大略的介紹整個追溯系統的架構，而追溯系統主要的架構是透過讀取的條碼到相關的資料庫查詢，並將資料顯示於使用者圖形介面中，架構圖如圖 1.4。本系統結合射頻辨識技術與網路服務技術，射頻辨識技術為供應鏈所搭載的載具，供應商的生產批號、批發市場拍賣序號及承銷商的進貨編號都儲存在射頻辨識標籤中；網路服務技術則是扮演遠端程序呼叫的角色，透過網路服務的中介，可以依條碼特性及提供者的不同對所需的資料庫進行存取。

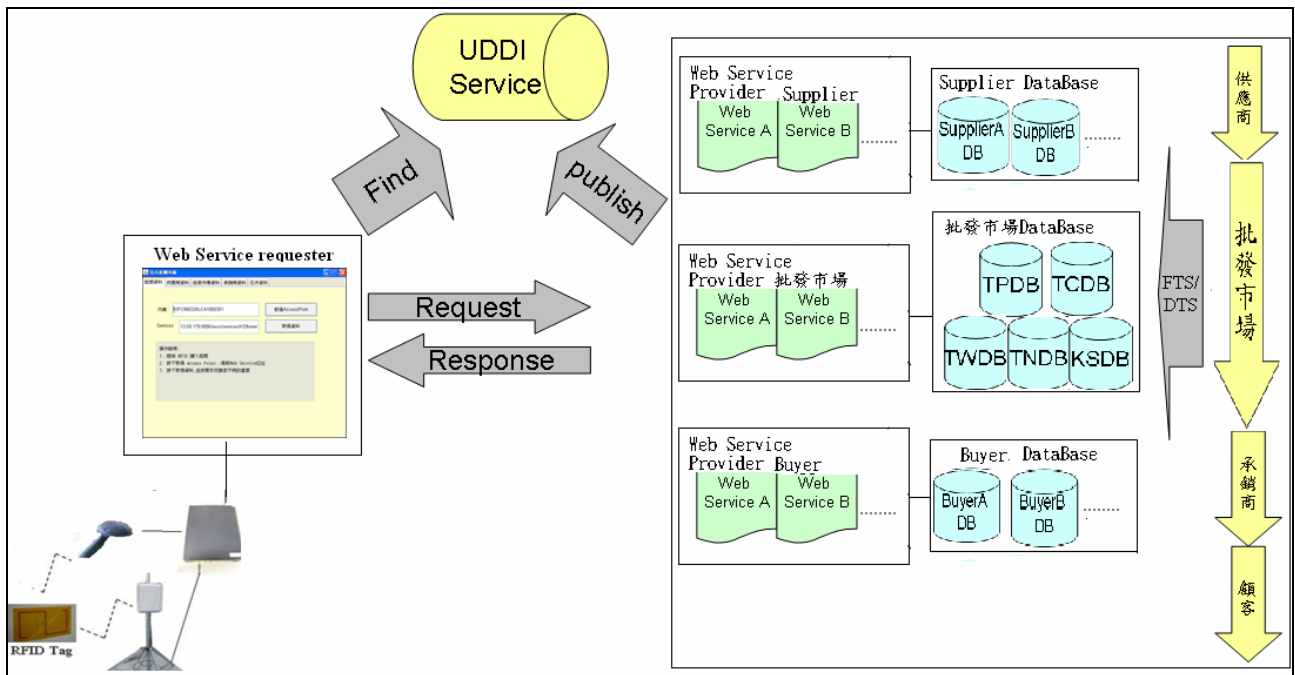


圖 1.4 追溯系統架構圖

整個系統建置過程可以分成前端與後端。後端分成資料庫和網路服務仲介端與伺服器端的部分。本研究中的花卉供應鏈主要包含供應商、批發市場、承銷商，供應鏈中各業者建立自己的資料庫紀錄花卉的產銷日誌，並建立各自的網路服務，服務的內容為存取各業者資料庫中的日誌資料。批發市場資訊儲存架構的部分，由於目前批發市場共有五座，因此依個數建立五個資料庫，包括台北、台中、台南、彰化、高雄花卉資料庫。各批發市場每日會透過檔案轉移服務(File Transfer Service, FTS)與資料轉移服務(Data Transfer Service, DTS)將當日拍賣資料匯入資料庫，進行更新。各供應商及承銷商建立自己的資料庫，紀錄每筆花卉商品的產銷資料。追溯系統後端的資訊儲存架構在第三章作詳細的介紹，另外也包含結合射頻辨識技術的花卉供應鏈運銷過程。

後端網路服務部分，各個供應商、批發市場、承銷商在網路服務伺服器端佈署自己網路服務，而每個服務內容是分別對自己的資料庫進行存取。接著再分別將各自的網路服務註冊到網路服務的仲介端，也就是圖中的 Publish，讓使用者可以透過仲介端找尋到適合的網路服務。而供應商與承銷商註冊的服務名為他們的代碼，批發市場則以他們的縮寫為主。表 1.2 為系統後端各個網路服務提供者從仲介端註冊到伺服器端佈署的對應表。網路服務仲介端的註冊與發佈則在第 4.2 節中介紹，伺服器端的建構與佈署在第 4.4 節中介紹。

表 1.2 追溯系統服務提供者註冊與佈署資訊

供應鏈業者	服務提供者	仲介服務名	伺服器署名	資料庫	關聯圖
供應商	供應商 A	LCA18	SupplierA	供應商 A 產銷日誌資料庫	圖 3.15
	供應商 B	LFQ99	SupplierB	供應商 B 產銷日誌資料庫	
	⋮				
批發市場	台北花市	TPFC	TPflower	台北花卉批發市場資料庫	圖 3.14
	台中花市	TCFC	TCflower	台中花卉批發市場資料庫	
	台南花市	TNFC	TNflower	台南花卉批發市場資料庫	
	彰化花市	TWFC	TWflower	彰化花卉批發市場資料庫	
	高雄花市	KIFC	KSflower	高雄花卉批發市場資料庫	
承銷商	承銷商 A	20935	BuyerA	承銷商 A 產銷日誌資料庫	圖 3.16
	承銷商 B	30919	BuyerB	承銷商 B 產銷日誌資料庫	
	⋮				

系統前端的操作主要分成五個部分，首先透過讀取器取得標籤的資料(包括服務名稱、供應商生產批號、批發市場拍賣序號、承銷商進貨編號)。接著透過服務名稱向 UDDI 註冊中心查詢適合的網路服務，如圖 1.3 中的 Find。第三步透過仲介端回傳的批發市場網路服務位址，將條碼序號傳送給伺服器端，對批發市場資料庫進行查詢，取得該筆花卉在批發市場中的日誌資料及供應商代碼與承銷商代碼。最後分別透過供應商代碼與承銷商代碼到 UDDI 註冊中心查詢該供應商與承銷商的服務位址，透過他們的網路服務及供應商生產批號與承銷商進貨編號取得這筆花卉的供應商日誌與承銷商日誌。

以一個例子說明。透過讀取器讀到標籤條碼為『096AAA001032KIFC960326LCA1800301096000231』，『096AAA001032』為供應商的生產批號，『KIFC』為批發市場所提供網路服務的服務名稱，『960326LCA1800301』為批發市場的拍賣序號，而『096000231』為承銷商的進貨編號。透過『KIFC』這個服務名稱向仲介端搜索，得到服務個體『KSflower』所註冊的『KIFC』服務，而其服務位址為『http://140.113.59.178:5800/axis/services/KSflower』。透過此服務位址，傳送批發市場的拍賣序號至伺服器端對高雄資料庫進行服務，得到這筆花卉在批發市場中的日誌資料，及這筆商品的供應商代碼為『LCA18』、承銷商代碼為『30919』。接著透過這兩個服務名稱向仲介端查詢，透過其網路服務得到該筆花卉在供應商 A 及承銷商 B 的產銷日誌，所有的資料透過使用者圖形介面顯示出來。例如此花卉商品的供應商為李玉燕小姐，供應地區為南投草屯，採取冷藏運輸,保鮮預冷處理；經過的批發市場為電子化交易、提貨管制監控的高雄批發市場，花卉等級 A3；而承銷業者為高雄縣的蘇榮雄，採取分級包裝、保鮮處理的方式，而該花卉商品為代碼 FR602 的玫瑰比浪卡。除了這些資料外，還有其他資料供追溯系統使用者參考，如表 5.4。系統前端的建構在第五章詳細的介紹。

1.5 研究方法

本論文的研究步驟一共分為七個步驟，如圖 1.5，分述如下：

1. 確認研究目的與範疇

本論文首要步驟是確認花卉供應鏈上的缺失與現行架構是否有需要進一步改善的地方，並衡量個人能力以及實際需求，規劃未來的研究目的。

2. 相關基本技術研究

確認研究目的後，針對所有可能需要的知識與技術進行了解與分析，並確認系統建構的可行性與完整性。除此之外，更尋找相關的研究了解當中的應用。

3. 廠商訪談

為了使理論研究與實作不至於脫軌，本研究實地訪問了高雄花卉批發市場，詳細了解並記錄供應鏈中各工作人員所遭遇到的問題。

4. 作業流程中技術的導入

規劃整個供應鏈中，射頻辨識技術的導入點，以及整個供應鏈在技術導入後的流程規劃和所提升的價值。

5. 資訊儲存架構規劃與建立

分別針對供應鏈上各業者建立各別之三階正規化資料庫，將整個供應鏈資訊流的資料儲存在其中。

6. 網路服務技術伺服器端、伺服器端及客戶端使用者圖形介面的建立

在網路服務技術架構下，透過『網路服務描述語言』文件描述各項提供的服務；而傳送的訊息則透過『簡單物件存取協定』加以包裝後在網路上傳輸；在網路服務仲介端利用『通用描述、發現與整合』企業註冊主機登錄資料。

7. 結論與建議

系統進行測試後，本研究將整個建構作業製成文字報告，以供往後相關人員進行開發與參考之用。

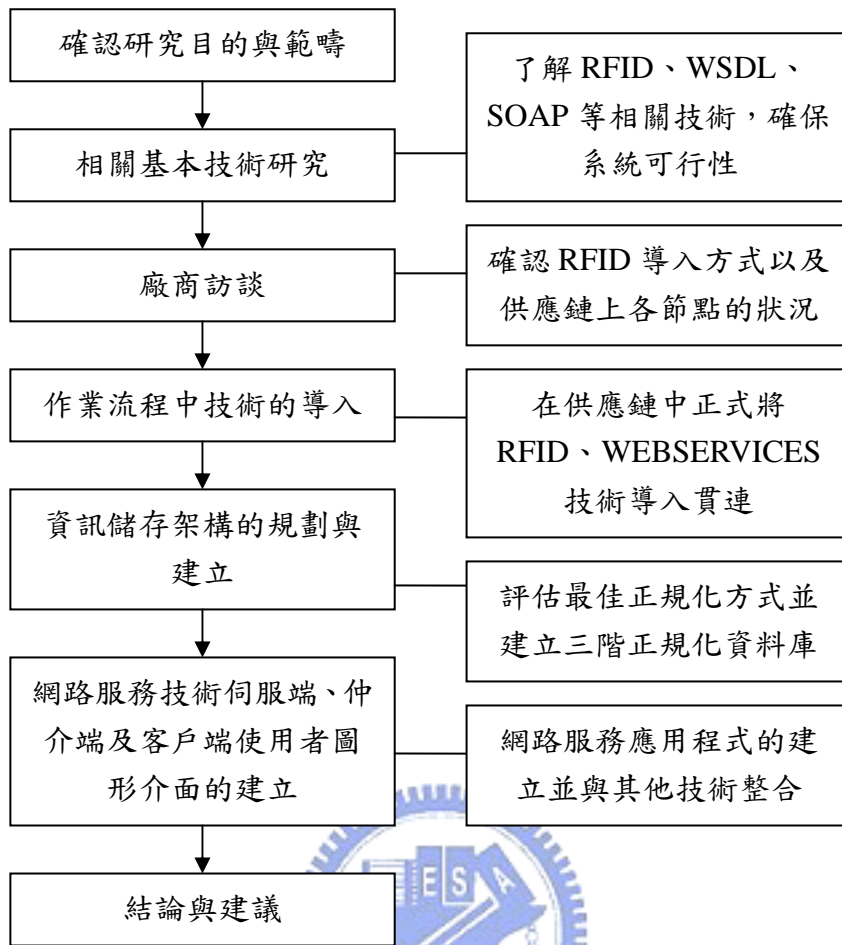


圖 1.5 研究方法架構

1.6 論文架構

本論文的內容包含六個章節，內容大綱如下。論文架構如圖 1.6 所示：

第一章：『緒論』，說明本論文之研究動機與背景、問題界定、研究方法以及論文的整體架構。

第二章：『文獻回顧』，探討本論文中所運用的參考資料、觀念與相關的研究內容。

第三章：『花卉供應鏈資訊儲存架構設計』，描述整個供應鏈過程並透過花卉供應鏈中各端點藉由射頻辨識技術所傳回的資料建立其關聯式三階正規化資料庫。

第四章：『網路服務仲介端與伺服器端的設計』，分為服務伺服器端與仲介端兩項。對仲介端而言，說明仲介端的操作使用，並在註冊端登錄；對服務伺服器端而言，針對服務伺服器端程式使用網路服務套件，建構網路服務應用查詢系統架構的伺服器端，建置方法主要由建置流程規劃、伺服器端程式設計以及佈署方法三個要素所組成。第五章：『網路服務客戶端的設計』，透過網路服務套件建立與伺服器端溝通的應用程式，並利用射頻識別辨識技術與服務需求端予以整合。

第六章：『結論』，結論與未來研究方向。

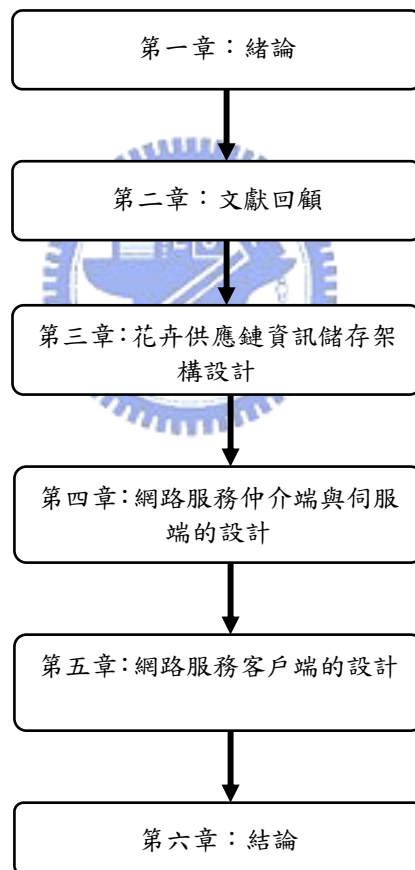


圖 1.6 論文架構圖

第二章 文獻回顧

本章的主要目的在於介紹花卉運銷體系及基本學理研究與技術規格說明，針對系統中所需要的技術進行探討。本章共分四節，內容為第 2.1 節「花卉運銷體系」，介紹花卉市場的運銷體系，並說明花卉供應鏈與花卉市場的關係。第 2.2 節「射頻識別技術」，對於射頻識別技術從技術標準、標籤、讀寫器以及後端中介軟體架構進行規格介紹。第 2.3 節「網路服務技術」中說明服務導向架構，並逐一介紹網路服務技術中的細節，包括可擴展標示語言、簡單物件存取協定、網路服務描述語言、通用描述、發現與整合等技術規格第。第 2.4 節介紹「追溯系統」。

2.1 花卉運銷體系

以台灣花卉運銷體系而言，由生產者到消費者中間所含的階層大致分類如下：1.生產者 2.農會運銷班 3.合作社、場 4.產地販運商 5.拍賣、批發、零批市場 6.市場承銷商 7.零售商(花店、小販)8.消費者(飯店、餐廳、一般消費者)[10]，其花卉運銷體系關係圖如圖 2.1 所示。

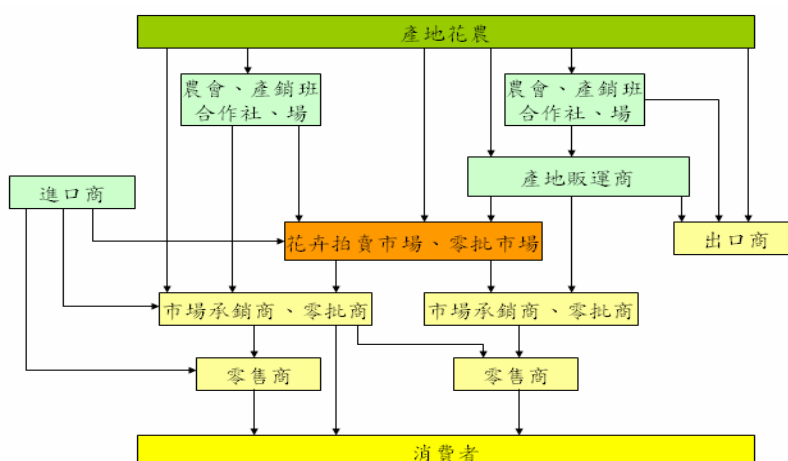


圖 2.1 花卉運銷體系的關係圖[10]

花卉不同於其他產品，其易受外在環境的影響，且本身有極高的耗損率，所以從生產端到消費者間的運送及保存過程中必須特別注意花卉產品的保鮮與保存。而花卉批發市場目前採用的荷蘭式拍賣由於交易時間短以及連續拍賣的特性，使得價格合理化，更讓花農在花卉產品交易上可有效解決花卉保存與保鮮的問題。此外，對消費者而言，組織化的批發市場除了提供多樣化的產品外，一次買齊的機制可降低因為運輸而增加的交易成本。由生產者至運銷者再到消費者三者形成完善的貯存與保護，對花卉之品質維護助益良多。

2.2 射頻識別技術

RFID 為 Radio Frequency Identification 的略稱，是一種利用電波進行感應式偵測數據載波的自動辨識技術，可以和嵌有記憶型 IC 晶片與小型天線的電子標示標籤進行無線通信，辨識出各別的讀寫資料及序號，隨著射頻辨識技術的導入達到降低成本、提昇了作業的準確度等價值。例如，在客戶的工廠裏生產訂製的車輛時，以往都是採用條碼來管理車輛的工程。然而，由於必須將條碼讀寫機貼近車輛來讀取，工程中停車的時間過長，讀取時也曾出現人為的疏失。

射頻識別相關技術可分為射頻識別設備與射頻識別整合服務兩大類。射頻識別設備是指射頻識別標籤、射頻識別讀寫器和相關應用系統；而射頻識別整合服務，是指將上述產品與原有資訊系統加以整合以提供加值服務。射頻辨識技術分別在以下小節介紹，第 2.2.1 節為「射頻辨識技術標準」；第 2.2.2 節為「射頻識別標籤、標牌與編碼」；第 2.2.3 節為「射頻識別讀寫器」；第 2.2.4 節為「中介軟體架構與資訊服務系統」；第 2.2.5 節為「射頻識別技術的困境」。

2.2.1 射頻識別技術標準

為了能使各種讀取器可以讀取到各種標籤的資料，並使讀取器可以順利連接到各套作業系統，因此射頻辨識技術需要制定其標準。目前射頻識別規格主要有電子化產品碼(Electronic Product Code, EPC)與國際標準組織(International Organization for Standardization, ISO)及泛在識別中心(Ubiquitous ID Center--UID)等三大標準組織。[21]

EPCglobal[17]為 2000 年由美國麻省理工學院(MIT)結合寶鹼(P&G)、金吉列(Gillette)、及美國編碼協會(Uniform Code Council, UCC)等，在 EAN.UCC 國際組織轄下發起成立的一個委員會。初時稱為 Auto ID Center，後又更名為 EPCglobal，並與 EAN.UCC 所推廣的全球商業編碼原則相結合，延伸其應用。EPCglobal 提出的是一個新的 Electronic Product Code(EPC)架構，EPC 系統亦是採用性能適合 Logistics Management 的 UHF 頻率，而 EPC 內容是一組數字。它在供應鏈中是某項產品的獨一代表，被儲存在 Tag 中，其資料包含此項產品製造的時間、製造地點、與製造的廠商等資料，藉著 RFID 的 Reader 將這組數字傳到互通之電腦網路中，電腦網路可將此數字所代表的該項產品所有資料提供給廠商。

EPC Network[13]系統架構整體的運作是透過標準的商品識別碼與網路應用系統的整合，以連結商品資訊。目的是讓上游廠商貨品流通至下游廠商時，隨著物品移動，沿途由射頻識別讀寫器發射無線電波感應物品的標籤，後端資訊系統可以透過網際網路即時查詢與存取物品資訊或傳遞該項物品的相關資訊於相關廠商的資訊體中。在整套機制的運作之下，其所傳輸的物品資訊，不只包括物件的基本資料例如：重量、外觀、材質等，還可以追溯至原料生產，下至終端配送狀況，並可以詳述物件活動路徑與生產過程。圖 2.2 說明 EPC Network 系統架構。

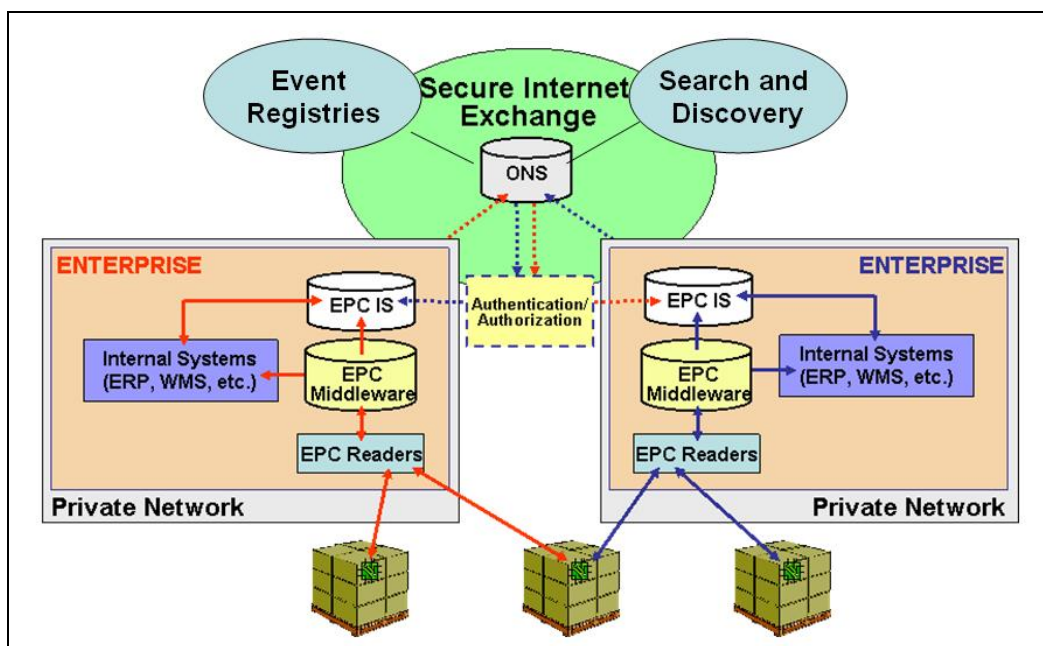


圖 2.2 EPC Network 架構圖[14]

國際標準組織(ISO)則是聯合國標準制定的推行組織，同時結合國際電子科技委員會(International Electrotechnical Commission, IEC)合作設立聯合技術委員會(Joint Technical Commission 1, JTC1)，並在這個委員會之下設立許多工作小組來發展此國際標準。

目前 RFID 業界通用的 ISO 標準有 ISO 14443「近旁型智慧卡(Proximity Coupling Smart Cards)」標準和 ISO 15693「近距型智慧卡(Vicinity Coupling Smart Cards)」標準。ISO 14443 界定在 10cm 內的距離以非接觸式的方式讀取智慧卡運作標準，一般之大眾運輸票價卡皆為此類；ISO 15693 規定讀取距離為長達 1 公尺之運作標準，它的簡單設計令其讀取器成本遠比 ISO 14443 低。由於使用方便又有追蹤功能，故 ISO 15693 主要用作進出控制等用途。它的缺點是其溝通速度只有 26K bits。表 2.1 為 ISO14443 及 ISO15693 的特性比較。

表 2.1 ISO14443 及 ISO15693 的特性比較

類別	ISO 14443	ISO 15693
操作頻率	13.56MHz	13.56MHz
讀取距離	近旁型智慧卡(Proximity Cards)	近距型(Vicinity Cards)
晶片類型	微控制器(MCU)或記憶體佈線邏輯 (一種無 MCU 的電子電路)	記憶體佈線邏輯
記憶體容量	64~64K	256~2KB
讀寫能力	可讀取和寫入資料	可讀取和寫入資料
資料傳輸率(kbps)	高達 106, 可提高至 848	高達 106
反碰撞功能	有	有

ISO 18000 Series Item Management 則為物流系統無線通信技術的首項國際標準。有鑒於供應鏈管理的重要性，ISO 特別由 ISO/IEC 小組規劃了一系列的 18000 標準，以規定 Item

Management 之 RFID 溝通語言(Air interface)準則。目前 ISO 18000 系列包含以下幾項重要之標準如表 2.2 所示。其中最重要的是 18000-6 之規定，其規範之頻率 860-930 MHz 為 Logistic Management 之最佳選擇，已成為國際 Supply Chain RFID 應用技術的重要標準。而 ISO 18000-6 的基準所定義的規範較 EPC 系統少，因此較 EPC 系統有更多的應用範圍。因為 ISO 主要只規範溝通語言的標準，不考慮標籤及閱讀器的資料內容及實際應用，亦不考慮其基礎設施的架構(如網絡技術及資訊應用平台)，其規格下的標籤只是單純的資料傳送媒體。

表 2.2 ISO18000 系列標準

Part	規定內容	應用領域範圍
18000-1	全球可接受頻率之 Air Interface Communication(AIC)一般參數	
18000-2	135MHz 以下之 AIC 參數	適合短距離之紙類標籤，如門禁卡
18000-3	13.56MHz 之 AIC 參數	記憶體佈線邏輯
18000-4	2.456GHz 之 AIC 參數	長距離之讀取範圍應用
18000-5	5.8GHz 之 AIC 參數	審議中止
18000-6	860~930MHz 之 AIC 參數	適用於 Logistic 及 Asset Mgt 之最佳方案
18000-7	433.92 MHz 之 AIC 參數	與 18000-6 相似，唯讀、速度慢且易受其他電訊器材干擾

主導日本 RFID 標準研究與應用的組織是 T-引擎論壇(T-Engine Forum)，該論壇已經擁有成員 475 家成員。值得注意的是成員絕大多數都是日本的廠商，如 NEC、日立、東芝等，但是少部分來自國外的著名廠商也有參與，如微軟、三星、LG 和 SKT。T-引擎論壇下屬的泛在識別中心(Ubiquitous ID Center--UID)成立於 2002 年 12 月，具體負責研究和推廣自動識別的核心技術，即在所有的物品上植入微型晶片，組建網路進行通信。UID 的核心是賦予現實世界中任何物理物件唯一的泛在識別號(Ucode)。它具備了 128 位(128-bit)的充裕容量，提供了 340x 1036 編碼空間，更可以用 128 位元為單元進一步擴展至 256、384 或 512 位。Ucode 的最大優勢是能包容現有編碼體系的元編碼設計，可以相容多種編碼，包括 JAN、UPC、ISBN、IPv6 地址、甚至電話號碼。Ucode 標籤具有多種形式，包括條碼、射頻標籤、智慧卡、有源晶片等。泛在識別中心把標籤進行分類，並設立了多個不同的認證標準。

2.2.2 射頻識別標籤、標牌與編碼

射頻識別標籤是由具有類比(Analog)、數位(Digital)與記憶體(Memory)功能的晶片，以及依不同頻率、應用環境而設計之天線所組成。如果依電子標籤是否附有電池來區分，可分為被動式及主動式射頻識別標籤。其中被動式射頻識別標籤的能源是由讀取器提供，所以標籤上不附加電池，因此體積小、使用期限較長，但是讀取的距離較短。對主動式射頻識別標籤而言，其標籤是附加電池的，系統另外增加喚醒裝置，平時標籤是處於休眠的狀態，當標籤進入喚醒裝置的範圍時，喚醒裝置利用無線電波或磁場來觸發或喚醒標籤，標籤這時才進入正常工作模式，開始傳送相關資訊。由於本身具備工作所需之電源所以傳輸距離較長，但是

相對具有體積較大、需更換電池及成本較高等缺點。

射頻識別標籤依據記憶體讀寫功能可以區分為三種，包括唯讀(Read-Only)、僅能寫入一次多次讀取(Write-Once Read-Many)、可重複讀寫(Read-Write)，其細節如表 2.3。

表 2.3 射頻識別標籤的記憶體讀寫功能

功能	說明
唯讀	標籤晶片內的資訊出廠時已固定，使用者僅能讀取標籤晶片內的資訊而無法進行寫入或修改的程序。成本較低，一般應用於門禁管理、車輛管理、物流管理、動物管理等。
僅能寫入一次多次讀取	使用者可以寫入或修改標籤晶片內資料一次，和唯讀標籤相同也可進行多次讀取。成本較高，一般應用於資產管理、生物管理、藥品管理、危險品管理、軍品管理等。
可重複讀寫	使用者可以透過讀取器進行標籤內晶片資訊之讀取與寫入，資料可以視需要附加或重新寫入。成本最高，一般應用於航空貨運及行李管理、客運及捷運票證、信用卡服務等。

在頻率的部分主要可以區分為低頻(Low Frequency)、高頻(High Frequency)、超高頻(Ultra High Frequency)、微波(Microwave)四種，細節如表 2.4。

表 2.4 射頻識別的頻率

頻率	說明
低頻	使用的頻段範圍為 10KHz~1MHz，常見的主要規格有 125KHz、135KHz。一般這個頻段的電子標籤都是被動式的。低頻的最大的優點在於其標籤靠近金屬或液體的物品上時能夠有效發射訊號，不像其他較高頻率之標籤，在這種情況下訊號會被金屬或液體反射回來，但缺點是讀取距離短、無法同時進行多標籤讀取以及資訊量較低。一般應用於門禁系統、動物晶片、汽車防盜器和玩具等。
高頻	使用的頻段範圍為 1MHz~400MHz，常見的主要規格為 13.56MHz。這個頻段的標籤主要還是以被動式為主。和低頻相較，其傳輸速度較快且可進行多標籤辨識，一般應用於圖書館管理、產品管理等。
超高頻	使用的頻段範圍 400MHz~1GHz，常見的主要規格有 433MHz、868~950MHz。主動式和被動式的應用在這個頻段都很常見，被動式標籤讀取距離約 3~4 公尺左右，傳輸速率較快，而且因為天線可採用蝕刻或印刷的方式製造，因此成本較低。雖然在金屬與液體的物品上的應用較不理想，但由於讀取距離較遠、資訊傳輸速率較快，而且可以同時進行大數量標籤的讀取與辨識。目前已成為市場的主流，未來將廣泛應用於航空旅客與行李管理系統、貨架及棧板管理、出貨管理、物流管理等。
微波	使用的頻段範圍為 1GHz 以上，常見的主要規格有 2.45GHz、5.8GHz。微波頻段的特性與應用和超高頻段相似，讀取距離約為 2 公尺，但是對於環境的敏感性較高，一般應用於行李追蹤、物品管理、供應鏈管理等。

對於 RFID 標籤的規格而言，目的是制定資訊交換時的標準資料格式。在 EPC 的部份推出六個等級的電子化標籤。第 0 級標籤是被動式唯讀標籤，記憶體有 64 位元；第 1 級標籤是被動式可讀寫標籤，但是只能寫入一次，記憶體 96 位元；第 2 級標籤是被動式可多次讀寫標籤，記憶體有 128 或 256 位元；第 3 級標籤為半主動式標籤，內建電池可供應電力；第 4 級為主動式標籤，標籤內建電池，記憶容量比其他等級的標籤更大；第 5 級為主動式標籤，規範標籤與標籤之間的自我溝通方式，此等級的標籤可以取代讀寫器的功能。目前產品最常見的是第 1 級標籤，其資料格式如表 2.5。

表 2.5 第 1 級標籤(96 位元)的資料格式

Element	Header	EPC Manager	Object Class	Serial Number
Bits	8	28	24	36
Values	0-256	0-268,435,456	0-16,777,216	0-68,719,476,736

ISO/IEC 15963 標準由分配等級、固定 ID 發行者登錄編號、序號三者組成。其中只有分配等級固定為 8 個位元長度，其他兩項的編號長度由發行團體自行決定，但是發行團體必須保證 ID 登錄編號為獨一無二的號碼。

UID 的 Ucode 資料長度為 128bits，視需求可擴展至 256、384、或 512bits 等長度。表 2.6 為日本 JAN code 為例的 128bits ucode 資料結構，包含了 12bits 的 code identifier、52bits 日本國內所使用的編碼(JAN code)、與 64bits 的 unique Identification(unique ID)。

表 2.6 Ubiquitous ID Center 的 ucode 資料基本結構

Element	Code Identifier	JAN code	Unique Identification
Bits	12	52	64

2.2.3 射頻識別讀寫器

在 RFID 系統中，讀寫器是利用射頻技術將標籤中的資料讀出，或將標籤所需要儲存的資料寫入標籤的裝置，讀寫器讀出的標籤資料通過電腦網路進行資料管理和資訊傳輸。讀寫器的構成一般由天線、射頻模組、讀寫模組構成。

讀取器製造商可分成三種類型，第一種希望能夠設定標準成為市場領導者的 RFID 晶片商，製造出完整的讀取器之後，針對讀取器製造出關鍵的 RFID 晶片。其目的是希望能夠設定標準成為市場領導者。例如德州儀器有提出貨車卸貨台傳送帶的讀取器解決方案，這些解決方案是針對 Procter & Gamble、BMW 或物流配銷商等公司設計的。這類廠商包含德州儀器、飛利浦半導體、與西門子等；第二種為中型 RFID 零組件的製造商，此類廠商向 RFID 晶片組廠商購買晶片之後，利用組裝能力，自行完成完整的 RFID 系統。其透過自行能力製造出來的 RFID 系統各有其差異點，因此服務包含了技術支援、安裝與測試整體系統等服務。其目標市場包含 Wal-Mart、Metro 與 Tesco 等零售大型通路商與配銷中心公司。這類廠商包含 TAGSYS、Matrics 與 Intermec 等；第三種為普通的小型製造商，透過跟大型的晶片商與中型的 RFID 板製造商拿貨，並配合自己生產的外殼而生產出讀卡機。因此他們的目標市場是以

使用手持式與行動 RFID 讀卡機的廠商與業者，差異化在於行動相關技術。雖然他們提供技術支援服務，但是卻沒能力提供完整顧問式服務。此類廠商包含法國的 Athelia、英國 Blackro 與 AIR LIQUIDE 等。

2.2.4 中介軟體架構與資訊服務系統

射頻辨識中介軟體與資訊服務系統扮演射頻辨識標籤和應用程式之間中介的角色，從應用程式端使用中介軟體所提供一組共通的應用程式介面 (API)，即能連到讀寫器，讀取到標籤上資料。如此一來，即使儲存標籤情報的資料庫軟體或後端應用程式增加或改由其他軟體取代等情況發生時，應用端不需修改也能處理，省去多對多連結的複雜維護問題。

射頻識別中介軟體從架構上可分為以應用程式為中心及以架構為中心兩種，以應用程式為中心為設計概念是透過讀取器廠商提供之 API，直接撰寫該讀取器讀取資料的程式，並傳送至後端系統的應用程式或資料庫，藉此達成與後端系統或服務串接之目的。而這種中介軟體架構只能達成單點對單點的連接，僅適用於企業內部單一商業應用系統。隨著企業應用系統的複雜度增高，企業無法負荷為每個應用程式撰寫各自的程式，企業可以考慮採用廠商所提供標準規格的射頻識別中介軟體。如此一來，即使儲存標籤情報的資料庫軟體改由其他軟體代替，或讀寫標籤的讀取器種類增加等情況發生時，應用端不做修改也能處理。

對於中介軟體的架構也逐漸致力於標準化，EPC 中介軟體架構建構網際網路的基礎上，通過管理軟體系統(Savant)、物件名稱服務(Object Name Service, ONS)與實體標記語言(Physical Markup Language, PML)實現全球的供應鏈產品資訊追蹤。

Savant 是位於讀取器和企業應用系統之間的中介軟體，主要負責收集、儲存讀取器發射的資訊並採取相對應行動。Reader 所收集的 EPC 碼會先傳送給 Savant，依據這樣的資訊，Savant 向散落各處的 ONS 提出詢問，由 ONS 找尋對應該 EPC 碼的產品資料位址，再回傳答覆給 Savant。Savant 主要功能有資料校對、讀寫器協調、資料過濾、事件處理。對於資料校對功能，可以對於 RFID 標籤資料的誤讀進行錯誤校正；對於讀寫器協調功能，可以避免重疊通訊區域的相異讀寫器讀取重複的 RFID 標籤資料；對於資料過濾功能，Savant 可以設定過濾條件，決定何種資訊需要傳送向其他供應鏈的資訊系統；對於事件處理功能，能夠維護事件資料庫的儲存，並且對特定事件設計對應處理動作。

標籤資訊用實體標記語言(PML)是一種以 XML 為基礎的規範，用以說明商品資訊格式，因此電腦可辨別接收進來的 EPC 碼是屬於哪家公司的產品，也可以確定該產品是在哪個工廠製造。延伸應用上，如當有瑕疵品出現時，可以很輕易的追溯問題發生點在何處。而 PML 主要是為物理實體的遠程監控和環境監控提供一種通用的描述語言，用來統一格式包裝 EPC 資料，以便資料傳遞時，能以有一致的資料分類方式，方便系統查詢、或統計。

物件命名服務：類似於網路環境下的 Domain Name Service(DNS)，提供 EPC 碼的位置資訊，當 Savant 需要查詢保存該商品資訊的伺服器網路位址，ONS 伺服器類似一個電話簿的功能，提供 EPC 碼與 EPC 資訊伺服器對照的功能。

而中介軟體的特色如表 2.7 所示。

表 2.7 中介軟體的特色

特色	說明
獨立架構 (Insulation Infrastructure)	中介軟體獨立並介於讀取器與後端應用程式之間，能夠與多個讀取器以及多個後端應用程式連接，以減輕架構與維護的複雜性。
資料流 (Data Flow)	中介軟體具有資料的蒐集、過濾、整合與傳遞等特性，以便將正確的物件資訊傳到企業後端的應用系統。
訊息流 (Process Flow)	採用程序邏輯及儲存再轉送 (store-and-forward) 的功能來提供循序的訊息流，具有資料流程設計與管理之能力。
標準 (Standard)	EPC 為中介軟體提出一個標準，透過無線射頻辨識標籤由讀取器讀入的資訊，並透過這套標準應用到電腦或是應用系統中。

2.2.5 射頻識別技術的困境

雖然射頻識別技術充滿發展性，但是現階段而言，仍有諸多困難尚待克服。最主要的瓶頸在於價格[14]、干擾性、批量讀取精確度[1]、隱私權的疑慮[27]四項。

在價格方面，除了標籤、讀取器及軟體開發的成本也包含許多客製化的應用成本。由於所有使用者有不同的應用材料、不同的天線設計需求、不同的使用環境、不同的國家規範以及不同的資訊應用需求等等，往往需要許多系統整合與客製化系統開發的成本。而射頻識別標籤的價格是由 IC 晶片、天線與外型封裝的成本所累加，目前的價格仍算是偏高，價格的降低是射頻識別突破的首要目標。

對於干擾性而言，RFID 運作主要是基於無線電波傳送的原理，但當無線電波遇到金屬或是液體時，訊號的傳導會產生干擾與衰減，因而影響到資料讀取的可靠性與準確度。一些特殊環境中的應用情境，如將 RFID 標籤貼於裝飲料的鋁罐外，或是金屬外殼的電腦設備上，都會遭遇到這個問題。而目前設備供應商以標籤特殊封裝的技術，來解決這個問題，將訊號干擾的影響程度減至最小。

而對批量讀取精確度而言，若批量的正確讀取率無法達到 100%，則必須人工重新檢驗，如此一來就喪失射頻識別技術自動化讀寫辨識的優勢。讀取精確度主要受到三項原因影響：射頻識別標籤與讀寫器的距離太遠、射頻識別標籤與讀寫器的方向不正確與同時多數標籤讀取之 Tag 訊號會有碰撞問題。以讀寫器的距離太遠問題而言，這是由於批量讀取時，每個物品與讀寫器的距離不一，因為排列的關係導致有的物品擺放位置超過了讀寫器的有效距離；射頻識別標籤與讀寫器的方向不正確主要原因也是批量物品混雜擺放於物流箱，如果標籤天線與讀取器天線之方向性正交，會有無法吸收能量或無法接收到訊號之問題；在多數標籤讀取之 Tag 訊號會有碰撞問題方面，已有一些解決方案技術的產生，如防碰撞技術(Anti-collision)與二進位搜尋技術(Binary Search)等。

在隱私權的疑慮方面，由於射頻辨識技術具有追蹤物品的功能，尤其是在消費者商品的使用上。當消費者於超市中購買商品時，商品上的資訊恐會被有心人士刻意收集而侵犯到個人的隱私權，人權團體針對這項質疑，使 RFID 的大量應用埋下了變數。未來尚需各國主管機關在隱私保護的相關法令訂定遵循的方向，特別針對 RFID 使用上隱私權的保護，以一些流程或機制來保障，才能促使業者開發相關確保隱私的資訊技術，並提高一般民眾使用具備 RFID 技術的商品與服務的意願。

2.3 網路服務技術

在網路普及的年代中，愈來愈多人想要利用網路來達成一些需求，但是資料又是分散在各地、各個不同的網站上。為了解決遠端、客戶端與伺服器端的資料溝通，因此有了網路服務技術[2] [23]的誕生。

網際網路服務是指一個支援透過網際網路遠端呼叫的軟體服務。服務的相關細節利用網路服務描述語言來描述，並透過標準的網路通訊協定，以 XML 格式所組成的簡單物件存取協定訊息處理溝通(通訊底層、進行傳送的动作、向通用描述、發現與整合進行搜尋或是註冊動作)與辨認程式碼的問題。在這樣的運作架構裡，主要包括三項關鍵技術：『簡單物件存取協定』、『網路服務描述語言』與『通用描述、發現與整合』。

網際網路服務標準使用到的技術及應用分別在以下小節中介紹：第 2.3.1 節說明網際網路服務歷史探究；第 2.3.2 節說明服務導向架構；第 2.3.3 可擴展標示語言；第 2.3.4 節說明簡單物件存取協定；第 2.3.5 節說明網路服務描述語言；第 2.3.6 節說明通用描述、發現與整合。

2.3.1 歷史探究

分散式系統的相關技術，從大型主機的時代後，漸漸走上分散式的系統架構。而 Server 與 Client 的溝通管道上，首先有 Remote Procedure Call (RPC)、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+/.NET Remoting、Sun 的提出的 RMI 與 EJB，以及 Macromedia 的 Flash Remoting 等技術。這些技術都是由 RPC 衍生而來，而 RPC 是利用網路硬體與通訊協定作為技術基礎，透過代理人進行要求與回應的动作，進而完成資訊的傳遞。

但是在實際發展過程中，以上這一些技術紛紛產生一些問題，例如 CORBA 的標準太過鬆散，以致於在伺服器端與客戶端間，會有安裝版本不易互通之缺點；微軟的標準雖然可以用 VB、C# 等不同的語言實作出來，但卻只能在自家的作業系統平台上操作；昇陽的 RMI 與 EJB 雖然可以橫跨不同作業系統平台，但是卻只能用 Java 語言溝通。從上面的分析後可以發現，現有的分散式系統技術存在一些問題，第一個就是不同的系統必須透過專有的通訊協定建立連結關係；第二個是需要不同的資料處理技術交換資料；第三個就是無法將技術擴展到協定結構鬆散的網際網路上。為了解決上面三種問題，國際上幾家資訊科技大廠，包括 Microsoft、IBM、Sun、HP 共同推出了網際網路服務的標準。宣稱可以利用 http 通訊協定，解決通訊協定複雜的問題，利用 XML 的方式來處理資料，因此有了網際網路服務的誕生。

2.3.2 服務導向架構

服務導向架構 (Service Oriented Architecture, SOA) [15] 是一種新興的系統架構模型，主要概念是針對企業需求組合而成的一組軟體元件。在官方定義中提到，服務導向架構是一種為了達到企業內部與企業之間跨系統整合之鬆散耦合目標的一種架構，而組合的元素則包含軟體元件、服務與流程三部分，當企業面對外部要求時，流程定義外部要求的處理程序；服務包括特定步驟的所有程式元件，而元件的功能是負責執行工作的程式。例如企業面對顧客訂單要求時，流程制定訂單的處理程序，服務則包括物料查詢、詢價與報價作業及接單作業

等，元件則是指真正完成前述的查詢或作業等工作的程式。

服務導向架構已成為現今軟體發展的重要技術，透過服務導向架構可以讓異質系統整合變得容易、程式再用度也提高，不必自行開發或擁有所有程式元件。發展者可以視需要組合網路上最好的服務，讓軟體開發人員更專注於企業需求的滿足，而不僅僅是資訊技術的開發。此外服務導向架構不受限於特定廠商的產品功能或是平台，達到真正的開放性。

服務導向架構具有四項特性：分散式架構、鬆散耦合的介面、開放的標準與流程角度。對分散式架構而言，服務導向架構的組成元件是由許多分散在網路上的系統元件組合，例如網路服務技術就是運作網際網路的 HTTP Protocol 來相互連結的服務導向架構；對鬆散耦合介面而言，因為傳統的系統架構是將應用系統功能需求切割成相互關聯的模組、物件或元件，發展者要花費極大的心力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設計，就成為一件困難的事情。而服務導向架構的作法則是以界面標準來組合系統，只要符合界面要求，內部零組件的設計反而沒有強制規定或限制。因此內部零組件可以任意替換，大幅提高系統變更的彈性；而對開放性標準而言，因為過去的軟體元件平台如 CORBA、DCOM、RMI、J2EE 採用各自專屬的協定作為元件連結的規範，使得不同平台的元件無法相通。因此服務導向架構採用完全開放的標準協定，可避免不同平台開發的程式互相整合的困擾；對流程角度而言，建構系統時，先了解特定工作的流程要求，並將其切割成服務界面(輸入與輸出格式)，如此其他的發展者就可以依據服務界面規格來開發程式，透過選擇合適的元件組裝完成。

網路服務技術是目前主要用來實現服務導向架構的介面標準，透過網路服務架構提供訊息處理的框架。而網路服務架構的發展，可以分為三個階段：簡易型網路服務、組成型網路服務、產業協同型網路服務。就簡易型網路服務而言，此階段主要以網路服務元件的開發為主，透過基本服務例如資訊查詢或非交易型服務作為網路服務發展的起點，可視為基礎建設的佈置與開發。而第二階段的組成型網路服務延續第一階段的基礎元件，透過組裝與串流的方式形成更複雜的功能，作為企業內系統整合、企業內應用系統服務化與企業間的服務。而就產業協同型網路服務而言，企業間透過網路服務技術進行即時、動態的商業協同運作。交易的協商與確認透過即時與自動化處理於整個供應鏈的運行，而協同合作的行為就發生在這個供應鏈的兩兩互動中，可視為是鏈對鏈的服務互動。

2.3.3 可擴展標記語言

可擴展標記語言(eXtensible Markup Language)[7]是 1998 年由 W3C 所公佈的，至今仍然沿用 XML Recommendation 1.0 規範，更新的版本 1.1 已接近完成。XML 是以文字為基礎(Text-based)的規範，具體了人機可讀(Human-Machine Readable)的特性，因此不涉及平台的問題，而且 XML 是以明文(Plain-text-based)的方式編譯，並允許使用者自己可以定義特定用途的文件格式。而這些文件格式經正式的律定，可以有依序的被參考應用。XML 文件的內容可以被自己的標記(Tag)，來解釋給外界知道，具有自我解釋(Self-Describing)的功能。

XML 描述了文件的結構、文件的語意，但沒有描述 XML 的格式，一個 XML 的文件形成了一個樹狀的結構。由於 XML 只是單純地用來描述資料，因此以瀏覽器開啟閱讀一份 XML 文件並無意義，必須加上其他技術支援與配合，才能達到資料的建構、展示與交換的目的。而這些技術包含文件資料驗證，定義 Document 的格式、顯示輸出和文件轉換、連結其它的

資源。

在文件資料驗證，定義 Document 的格式部份包括 DTD (Document Type Definition)和 XML Schema。DTD 就是 SGML 的語法檢查，XML1.0 版也支援此驗證方式，DTD 能夠幫助 XML 剖析器來解譯 XML 文件的內容。而 DTD 雖然可以定義 XML 標籤和架構，不過使用的語法不同，而且缺乏擴充性，所以一些軟體廠商如 Microsoft 等就提出了一種不同的檢查系統，也就是 XML Schema。XML Schema 語法檢查擁有擴充性，可以直接擴充別人現成定義的 XML Schema，XML 文件的標籤內容都是文字字串，但是 XML Schema 可以定義標籤的資料型態。雖然在撰寫上仍然為文字內容，不過在撰寫程式讀取 XML 元素時，可以取得元素的資料型態。

關於顯示輸出和文件轉換，包括 CSS(Cascading Style Sheets)和 XSL (Extensible Stylesheet Language) /XSLT /XSL-FO [26]，如圖 2.3。CSS 並不是全新的技術，目前版本為 2.0 版，HTML 文件可以使用 CSS 重新定義標籤樣式，XML 文件也一樣適用，CSS 可以用來定義 XML 標籤顯示的樣式，將一致的外觀應用到所屬網站上。XSL 本身便是一項 XML 的應用，直接架構在 XML 的語法之上。一共分做兩部分：第一部分負責將 XML 原始碼轉換為另一種格式，也就是「XSLT」(Extensible Stylesheet Language Transformation)，使用 XSLT 可以將 XML 文件的樹狀架構轉換成另一個新的架構。它是一種樹狀結構導向的轉換語言，能夠將 XML 文件轉換成文字、HTML 或其它 XML 文件。XSLT 使用「XPath」(XML Path Language)在 XML 文件找尋資料，它的語法可以用來指出文件架構或資料的位置；第二部分稱作 XSLT/XSL-FO [Extensible Stylesheet Language Formatting Object]，XSL-FO 允許使用者對內容頁數、版面設計、樣式資訊等做詳細的指定，達到精確設定其外觀式樣並可以將其列印出來。XSL-FO 的標記十分複雜且冗長，而要產生 XSL-FO 檔最實際的方法就是用 XSLT 來產生一份來源文件，之後再將該文件作為輸出的媒介。由於只有少數的工具支援 FO，因此 FO 的發展並不如 XSLT 那般快速。

在連結其它的資源的部份，包括 XPoint (XML Pointer Language)和 Xlink (XML Linking Language)，XML Pointer Language 能夠連結 XML 文件本身或其他的文件，它並不是用來搜尋文件，而是用來在文件內定址，以便快速的找到所需標籤或內容，主要為 Xlink 的定址方式。而 XML Linking Language 能夠連結一系列相關文件內容的資源，這些資源可以為元素、內容和部分內容，除了支援超連結的單一連結外，也支援更複雜的連結方式，而這兩種語言皆屬於 XML 草案階段的規格，目前大部分的剖析程式並未支援。

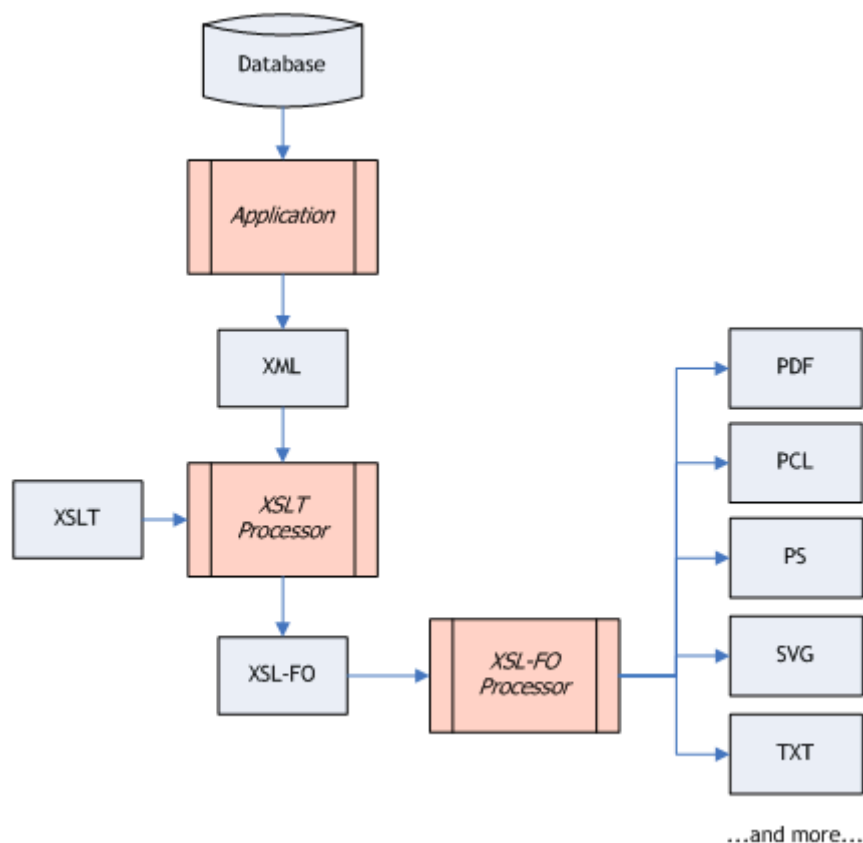


圖 2.3 顯示輸出和文件轉換架構[26]



2.3.4 簡單物件存取協定

簡單物件存取協定[24]並沒有特定的程式語言模型或是語意，純粹是規範一種機制。而簡單物件存取協定的基礎目標是在網際網路環境中，傳遞標準格式的文字訊息，讓各種不同的服務在任何系統平台之間，以多種傳輸協定例如 HTTP、SMTP、FTP 等，進行傳輸溝通而達到異質環境中系統整合的目的。簡單物件存取協定傳遞的文字訊息，使用 XML 文件主要是著眼於 XML 雖然可以作為功能強大的資料格式，但是 XML 本身卻不能對於資料傳遞方式有所規範。因此造成使用的雖然是 XML 此種開放性的標準，但卻因為每一家軟體公司所使用的傳遞方式不同，而使得 XML 的開放性受到很大的限制。而目前 SOAP1.2 共有十份相關的規範[24]，如表 2.8 所示。

其中四分文件已於 2003 年 6 月正式公佈為正式規範，分別是：SOAP VERSION 1.2 PART0: PRIMER SOAP VERSION 1.2、PART1: MESSAGING FRAMEWORK、SOAP VERSION 1.2 PART2: ADJUNCTS、SOAP VERSION 1.2 SPECIFICATION ASSERTIONS AND TEST COLLECTION。

表 2.8 SOAP 相關規範[24]

SOAP 1.2 相關規範和名稱	修訂時間	狀態
SOAP Version 1.2 Part:Primer	2003/6/24	Recommendation
SOAP Version 1.2 Part1: Messaging Framework	2003/6/24	Recommendation
SOAP Version 1.2 Part2: Adjuncts	2003/6/24	Recommendation
SOAP Version 1.2 Specification Assertions and Test Collection	2003/6/24	Recommendation
SOAP Version 1.2 Attachment Feature	2002/9/24	Working Draft
SOAP Version 1.2 Email Binding	2002/7/3	Note
SOAP Message Normalization	2003/10/8	Note
SOAP Transmission Optimization Mechanism	2003/7/21	Working Draft
SOAP Optimized Serialization Use Cases and Requirements	2003/11/12	Working Draft
SOAP Version 1.2 Usage Scenarios	2003/7/30	Note

SOAP 主要是由四大部分所組成，分別是 SOAP Encoding Rules、SOAP RPC Representation、SOAP Binding、SOAP Binding。SOAP Encoding Rules 是指編碼規則定義了特定的應用程式之間，若要進行資料轉換時，所需的資料類型交換方式，編碼規則是有關於資料如何被編碼及序列化的一組定義集；SOAP RPC Representation 定義了一個用於遠端程序呼叫及回應的協定，也就是如何利用這些現有的通訊協定如 HTTP、SMTP、FTP 等來發送 SOAP 訊息；SOAP Binding 定義了將 Web Services Addressing Core 中的抽象屬性與 SOAP 訊息結合的方法；SOAP Envelope 定義了一個完整的架構，包括訊息的內容是什麼、誰該處理它、甚至於它是強制性或是選擇性接受。每個在 SOA 架構下的 Find、Publish、Binding 都是依靠 SOAP 訊息來傳遞，無論是用於呼叫或回應，SOAP 訊息本身就是一份 XML 文件，而 SOAP Envelope 是 SOAP Message 的根元素，而且為必要元素，對外代表著這份文件是 SOAP Message。每份 SOAP Message 為一份 XML 文件，那代表一定會有一個根元素就是 Envelope，並含有兩個子元素，分別為 SOAP Header 及 SOAP Body。SOAP Message 的 XML 架構，如圖 2.4 所示。

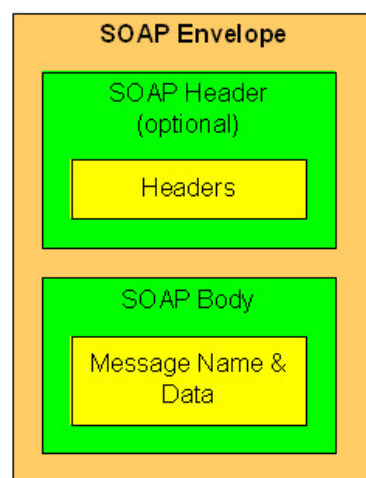


圖 2.4 SOAP Message 架構圖[24]

SOAP Header 為非必要的元素，它定義了一些屬性，包括誰該處理該項訊息及此份訊息是「選擇性」或「強制性」處理等，類似信封套的角色，其作用是協助寄件者能依相關資訊送交到最終的收件者。但如果 SOAP Message 存在 Header 元素的話，那麼必須為 SOAP Envelope 的第一個子元素。SOAP Header 有四個屬性，其屬性值如表 2.9 所示。

表 2.9 SOAP Header 的屬性值[24]

屬性名稱	說明
encodingStyle	用於說明 SOAP 訊息序列化所採用的編碼規則，『http://www.w3.org/2003/05/soap-encoding』為編碼規則的定義集。
mustUnderstand	用來指明 SOAP 訊息的接收者是否一定要處理 Header 項，屬性值有 true 與 false 兩種，預設值為 false。
Role	用於賦予 SOAP 節點單一或多重 SOAP 角色。在 SOAP 文件接收與轉傳過程中，各個 SOAP 節點可根據 role 屬性敘述，決定是否擁有讀取 Header 元素中某個區塊內容的權限或是將該 SOAP 訊息繼續傳送到下個節點。
Relay	用來指定某些特定 SOAP Header 區塊需要被傳遞而不需要處理。屬性值有 true 與 false 兩種，預設值為 false。

SOAP Body 為必要的元素，類似信中信紙的角色，裡面包含所要傳送的資料，所有在 Body 下的子元素都稱為是 Body Entity，這些子元素的內容就是所要傳送的訊息，當 Body 元素發生錯誤時，錯誤訊息由 SOAP Fault 定義。而 SOAP 封包方面主要是採單向傳送，而 SOAP Message 依靠結合接收端的應用程式完成單向傳遞(呼叫/回應)。

2.3.5 網路服務描述語言

網路服務描述語言[25]由 IBM 與 Microsoft 所共同制定，以 XML schema 來界定 Web Services 應含的內容。因此網路服務的描述最後會以 XML 文件形式與以儲存，且以.wsdl 來當作它的副檔名。

當服務需求者透過通用描述、發現與整合找到所需要的服務時，為了讓服務需求者使用該服務，服務提供者必須為所提供的服務撰寫使用說明書，而網路服務描述語言就是網路服務使用說明書的文件格式。服務需求者可以依據網路服務描述語言文件，利用轉換工具例如昇陽公司的 WSDL2Java、webMethod 的 Glue 等產生與網路服務描述語言文件相對應的程式碼，經過針對需求的小部分調整設定，便可以使用該項服務。使用網路服務描述語言文件讓網路服務應用程式能以一種標準方法來描述自己擁有哪些能力，以便讓互動更容易進行。想要瞭解這項網路服務，便需要瞭解該網路服務之網路服務描述語言的定義、目的以及結構等。

網路服務描述語言內容主要分為六個主要元素：Port、Service、Types、Message、portType、Binding[6][12]。Port 與 Service 屬於實作描述，而 Types、Message、portType、Binding 則屬於介面描述，其對應關係如圖 2.5 所示，細節如表 2.10。

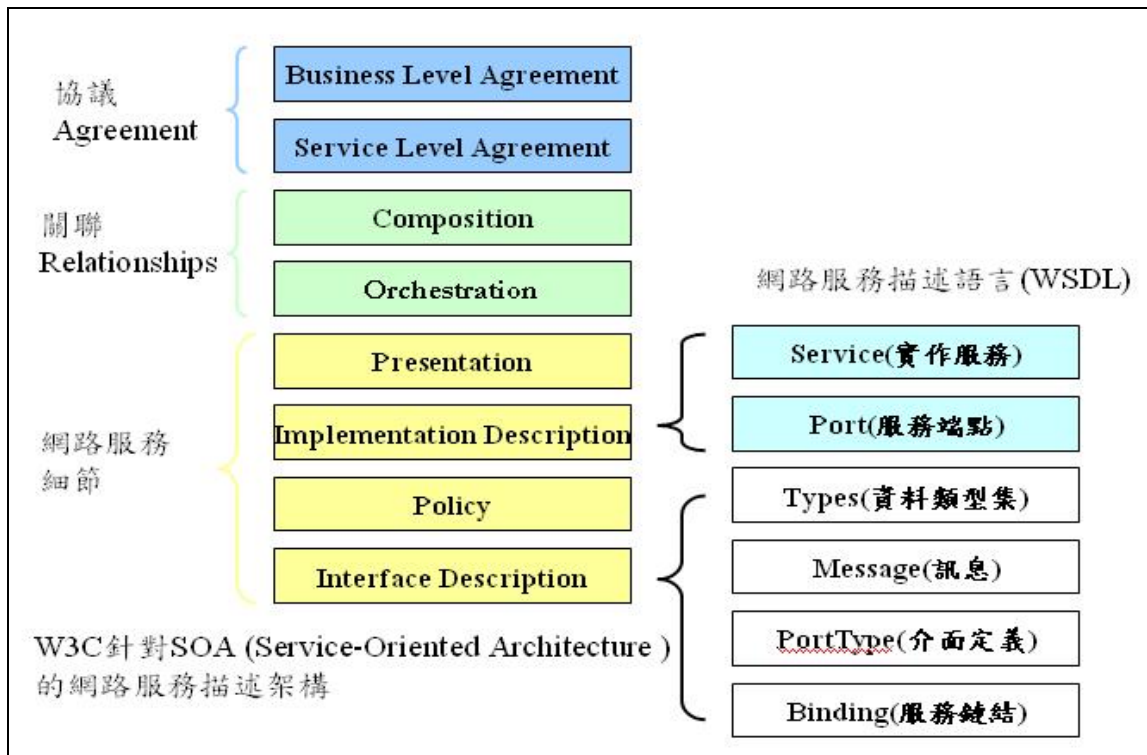


圖 2.5 網路服務描述架構與網路服務描述語言對應圖[6]

表 2.10 網路服務描述語言的主要元素[6]

屬性名稱	說明
Port	描述實際在進行資料傳輸時，提供服務的端點位置(URI)，而該 Port 代表外界可以和此 Service 溝通的一個進入點。
Service	具體描述一個 Web Service 所有作業端點的佈署細節，除提供服務的 Port 外也包含整份 WSDL 服務的服務名稱。
Types	定義訊息交換時的資料結構，在網路服務上，所謂的服務是透過資料交換來達成其目的，一般性資料結構如果可以引用，則不需重覆定義，若需擴充的資料項目，則在此定義。
Message	訊息代表網路服務的細節抽象概念，藉由表達輸入、輸出或錯誤等操作用訊息說明哪些資訊需要傳輸。
portType	主要是定義各項網路服務作業，每個作業中分別定義 Message 的資料輸入與輸出項，且用抽象的方式指出該網路服務究竟能做些什麼，因此常作為檢視網際網路描述語言的起點。而 WSDL 規範了四種主要的端點服務傳輸方式，分別為 One-way、Request-response、Solicit-response、Notification。
Binding	主要具體描述在 Binding 部分的 PortType 物件的鏈結，清楚定義由 PortType 服務流程及 Message 訊息段所提到的資料元素部份。而 Binding 因使用的協定不同，可分成三種方式：SOAP Binding、HTTP Get/Post Binding、MIME Binding。

2.3.6 通用描述、發現與整合

通用描述、發現與整合[22]的緣起主要是因為過去每當一個新的商業實體要進入應用體系，應用體系的鏈結必須反覆的重新整合，所以需要一個良好的註冊機制架構，而且必須超過傳統的目錄架構，透過 UDDI 註冊機制，商業實體能夠以標準化的方法建立註冊資訊，並描述本身和其所提供的服務，並且能夠讓登錄註冊的商業實體，透過分散式的方法去互通和分享資訊。IBM 及 Microsoft 等業者於 2000 年 9 月共同催生 UDDI 1.0 的計劃，發展至今已發行到第三版。

通用描述、發現與整合在網路服務架構扮演註冊服務的角色，而它使得商業實體能夠發現彼此、網路上的互動方式能有統一的標準遵循、透過共通性的註冊資訊，能夠分享商業實體的資訊及其提供的服務所需的相關介面資訊。

針對運作架而言，通用描述、發現與整合主要有三項功能：公佈(Publish)、尋找(Find)與鏈結(Bind)。對「公佈」而言，是指網路服務提供者可以在通用描述、發現與整合主機上進行註冊的動作，讓其他應用程式透過標準 XML 介面取得；對「尋找」而言，服務需求者在通用描述、發現與整合主機搜尋適當的服務位址；對「鏈結」而言，是指服務需求者在搜尋到服務之後，將應用程式連接至網路服務。

對於通用描述、發現與整合的註冊資料就是用 XML 格式來界定一致的註冊資訊，有些技術文件將其分成三部份：關於商業資訊的白皮書(White Page)，依據標準商業法的黃皮書(Yellow Page)，網路服務資訊細節描述的綠皮書(Green Page) [22]。而白皮書用於登錄商業實體的一般資訊。如：公司名稱，聯絡資訊(住址，E-mail，電話等)，已知的企業標識等；黃皮書是依通用描述、發現與整合分類別來登錄商業實體，目的是讓服務需求者可按分類尋找到目前已登錄且符合要求之商業實體，如工業分類索引，服務和產品分類索引，地理分類索引等；綠皮書用來描述網路需求者，如何與服務提供者進行交易的技術細節資訊。如：電子商務的交易規則，商業實體所提供有關其網路服務的描述與技術資訊，網路服務技術規格的參考資料，以 URL 取用網路服務的參考格式等。

通用描述、發現與整合的註冊資訊是使用 XML Schema 來定義，共包含六項。其中商業實體(BusinessEntity)封裝了白皮書與黃皮書中的資訊，讓服務的使用者能夠取得服務提供者的相關資訊，此外能透過姓名分類或商業分類的方式來查詢欲使用的服務；商業服務(BusinessService)包含一系列有關商業流程或分類目錄的網路服務，並將其描述組合在一起；服務鏈結(BindingTemplate)封裝了通訊資訊、使用順序、路由選擇、負擔平衡等資訊，用來對應上一層的商業服務，而商業服務與服務鏈結共同組成綠皮書所有的規範資訊；技術模型(tModel)描述資料的具體資訊，例如輸出入參數型態、使用之通訊協定、使用之安全機制等，為 UDDI 的核心；商業實體關聯性(PublisherAssertion)說明商業實體間的相關性；訂閱(Subscriptions)提供客戶端有條件的登錄及訂閱相關資訊，以利異動的及時更新。這些資料節點的對應關係如圖 2.6 所示，而資料結構模型的元素如表 2.11，屬性如表 2.12。

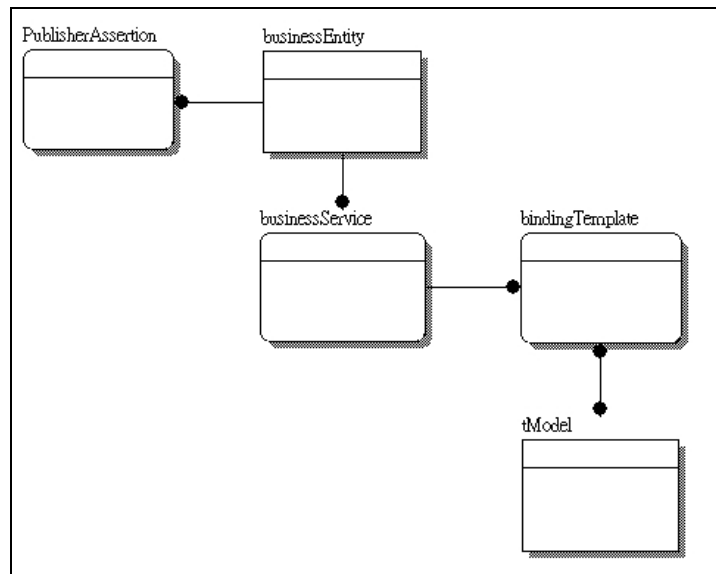


圖 2.6 重要資料節點的對應關係[28]

表 2.11 資料結構模型的元素[6]

資料結構模型	元素	說明
businessEntity	name discoveryURL description contacts businessServices identifierBag categoryBag	商業實體中的必要元素，紀錄商業的名字。 提供查詢企業的 URL 等資料。 封裝與該服務提供者的簡短描述。 封裝與該服務提供者聯絡用的資訊。 該企業所能提供之網路服務之名稱集合。 紀錄商業的識別名稱及其值。 紀錄此服務提供者的分類資訊。
businessService	name description categoryBag bindingTemplate	商業服務的名稱。 商業服務的描述。 記錄商業服務的分類資訊。 記錄與此商業服務相對映的服務鏈結資訊。
bindingTemplate	tModelInstanceDetails description accessPoint(address) hostingRedirector	描述 tModel 的細節。 服務鏈結的簡短描述。 記錄服務鏈結所描述的網路服務存取位址。 當 accessPoint 無法運行之備案。
tModel	name description overviewDoc(WSDL) identifierBag categoryBag	tModel 的名稱。 tModel 的描述。 封裝與 tModel 相對映之描述或使用說明。 tModel 附屬識別字的名稱與對應值。 分類資訊之名稱及其對應值。
publisherAssertion	fromKey toKey keyedReference	指向已註冊之商業個體的索引值。 指向已註冊之商業個體的索引值。 說明兩商業個體之間是以何種關係來描述。

表 2.12 資料結構模型的屬性[6]

資料結構模型	屬性	說明
businessEntity	businessKey operator authorizedName	註冊中心自動賦予，指向商業個體的索引值。 控管註冊資料的原始註冊中心的認證名稱。 實際發佈商業個體資料的授權企業。
businessService	serviceKey businessKey	記錄商業服務的索引值。 記錄上一層商業個體的索引值。
bindingTemplate	bindingKey serviceKey	記錄服務鏈結索引值，由註冊中心直接配給。 記錄上一層商業服務的索引值。
tModel	tModelKey operator authorizedName	記錄 tModel 的索引值，由註冊中心直接配給。 管控 tModel 的註冊中心的名稱。 發佈 tModel 的授權企業。
publisherAssertion	KeyName keyValue tModelKey	KeyValue 屬性的名稱。 關聯關係的代碼。 指定所使用之關係。



2.4 追溯系統

近年來供應鏈的興起，各個企業漸漸著重在於自身核心技術上，將非核心的技術委外給專業的製造商，並直接出貨給下游，以降低成本並贏得競爭優勢。在產品到達消費者之前，將會透過許多的廠商，不管是提供原料、加工製造亦或是轉手販售等。這樣的情況導致產品的品質與管理成為一個問題，為了有效的收集、儲存、利用分佈於各廠商的產品資訊，建立追溯系統是最好的一種方式。不僅如此，食品安全的問題越來越受到消費者的重視[29][30]。更於 2005 年歐盟《食品法》規定食品資訊必須具有可追溯性，並成立食品追溯紀錄管理辦法，未來進口食品若不遵守該辦法，將不得進入歐盟市場。目前可追蹤性並沒有一致的定義，多普(KJ van Dorp)[31]在 2002 年整理了可追蹤性的相關定義，如表 2.13 所示。

表 2.13 可追蹤性之定義[31]

作者	定義
美國資源管理協會 APICS (1992)	對可追蹤性有兩部分的想法：(1)提供所定義之貨物的行進所在位置之屬性；(2)在一批次的生產中，零件、製程和原料的登記與追蹤。
Beulens 等 (1999)	可追蹤性是為證明運送貨物或服務的履歷符合規格之能力。然而，如果著眼於追蹤與追溯時，可追蹤性指在特定商業交易後，顧客與供應商仍之約束關係。
國際標準組織 ISO (1994)	可追蹤性是藉由記載的識別證明進行追溯歷史、應用或是實體位置的能力。ISO 敘述可追蹤性關係到產品與零件的來源、產品的製造歷史和交貨後產品的分佈與位置。且根據 ISO 的記載可追蹤性包含了轉換輸入到輸出之資源與活動的部分。
Jansen (1998)	產品追蹤與產品追溯之間有所區別。產品追蹤來自於產品價值或任務，用於能找出產品的所在位置；產品的追溯來自於例外的處理，用於確定不良品質的來源。
Kim 等 (1995)	可追蹤性稱為產品的血統的證明，其包含了(1)追溯的本質、(2)定義單一識別、(3)血統的圖形化標記所註釋的可追蹤關係。
國際製造執行系統協會 MESA (1997)	可追蹤性提供了產品所在地與相關處理之狀態資訊能見性。狀態資訊包含了誰對它作業、組成、原料、投料量、供應商、批次、序號、流通產品的條件、警告、產品的修正或例外。除了能見性以外，線上追蹤功能創造了允許追蹤元件與處理最終產品之重要歷史的記載
Moe 等 (1998)	可追蹤性是可以追蹤產品批量與每個產品批量的歷史資料之能力，歷史資料來自其產品在收成、運送、儲存、加工、配銷和販售各步驟間的流動或是各步驟的內部程序。
Rijn 等 (1993)	可追蹤性涉及(1)製作過程、(2)定義原料批量之識別、(3)追蹤所在位置與品質的資訊、(4)追溯原料來源與利用的資訊。
Van Twillert (1999)	追蹤與追溯可以被細分為追蹤部分與向前及向後可追蹤性部分。追蹤部份由當物品在供應鏈生產流通之所在位置的確認所構成。向前可追蹤性是關於一起生產的物品在供應鏈中所在位置之確認。向後可追蹤性是用於不良品問題來源的確定。
Weigand (1997)	追蹤與追溯被認為是一個現代化的工具，它可以洞察產品的來源與供應鏈中個別作業所使用的物品。
Wilson and Clarke (1998)	食品可追蹤性可以被定義為(1)描述產品收成歷史的資訊必需性質 (2)描述從收成之後一直到顧客手中任何處理資訊與加工資訊之必需性質。

產品安全是供應鏈導入追溯系統的一項因素，但是對於企業來說產品安全只是其中的一項，追溯系統對於產業的價值，共可列舉如下：

1. 提升供應鏈整體的效率：導入了追溯系統，所有的廠商都得進行流程的自動化與商品資訊的統一，藉此可提升在製品的品質管理。
2. 降低風險：追溯系統可以提供供應鏈中完整的資料，倘若產品具追溯性，則可釐清是否為該企業所生產之產品造成損害。
3. 確認產品安全：追溯系統最主要功能在於追蹤上游及追溯下游，當產品發生問題時能夠很快且完整的回收全部的貨物，不僅如此，更能查出是供應鏈中哪個部分出現問題。
4. 強化工廠中製造管理的功能：依照原料點收、原料的計量與分裝、投入原料至生產線等各項主要工程來收集資訊。
5. 消費者的信賴感：透過可追蹤性，消費者能夠得到產品的更多資訊，產品的售後服務將優於從前。

追溯系統的導入受到各個產業的類別、導入目的等因素的影響，因此各個產業導入的模式並沒有一定的規範。對花卉供應鏈來說，著重的是上游的生產、原料資訊到中游的拍賣、管理過程再到下游的出貨，因此本研究將可追溯性應用在花卉產業上主要是可以確認供應鏈中的狀況，提供供應鏈的資訊給資訊需求者，此外提升整個供應鏈中品質的控管。



第三章 花卉供應鏈資訊儲存架構設計

本章主要描述無線射頻辨識技術在花卉供應鏈中扮演的角色以及完整的資訊流過程，從花卉批發市場的交易物流作業中將交易資訊透過檔案轉移服務傳到伺服器端，再藉由資料轉移服務將資料轉入關聯式三階正規化資料庫，完成花卉供應鏈各端點資料自動化存取的動作。本章共分為三個小節，第 3.1 節為「花卉供應鏈運銷」；第 3.2 節說明「花卉批發市場資訊儲存架構設計」；第 3.3 節為「供應商與承銷商的資訊儲存架構設計」。

3.1 花卉供應鏈運銷

花卉產品的運銷由生產者轉移到消費者手中，期間必須考量產品特性在整個運銷過程中的變化及市場交易制度與方式。自產地做最適切的準備，再由安全迅速的運送過程，保持產品新鮮完整的送達市場。經由市場以最快速、合理的價格分銷至承銷人，最後送達消費者手中，最後到達消費者手中。整個台灣花卉運銷體系在 2.1 節已有討論，在本節更詳細的描述花卉貯存和運送方式。第 3.1.1 節說明「產地的花卉貯存與運送」；第 3.1.2 節說明「花卉批發市場作業流程」；第 3.1.3 節說明「零售據點的貯存與運送」。

3.1.1 產地的花卉貯存與運送

花卉採收，由田間即需考慮延續其水份之供給，採收於盛水之容器中。裝運至集貨據點後，經過分級篩選，依等級花卉別裝入桶裝容器中，吸足水份並做保鮮處理。送上裝載之台車後，運送至冷藏庫，並依花卉種類特性以適度之低溫預冷。出貨時，結合冷藏運輸，對花卉進行保鮮處理及預冷處理，並將花卉裝箱，且在箱子上貼上無線射頻識別標籤，送上台車及冷藏運輸車，以確保由採收至運輸過程之品質保護。標籤內容包括花卉別、供應商、產地等資訊。

3.1.2 花卉批發市場作業流程

花卉批發市場的五大交易作業流程分別是進貨、理貨、拍賣、分貨以及領貨作業，為花卉市場中最重要的交易流程，共分五個小節分述將無線射頻辨識技術導入花卉批發市場後其細部流程。第 3.1.2.1 節說明「花卉批發市場的進貨流程分析」；第 3.1.2.2 節說明「花卉批發市場的理貨流程分析」；第 3.1.2.3 節說明「花卉批發市場的拍賣流程分析」；第 3.1.2.4 節說明「花卉批發市場的分貨流程分析」；第 3.1.2.5 節說明「花卉批發市場的領貨流程分析」。

3.1.2.1 花卉批發市場的進貨流程分析

進貨是指把從產地運來的花卉由貨車司機轉送至批發市場的流程[8]。花市進卸貨狀況如圖 3.1 所示。當貨車進入花卉批發市場之後，透過讀取各個箱子上的射頻辨識標籤之花卉種類，分別將各箱花卉依其種類、等級放置於所屬拍賣線的卸貨區。由於貨車通常都趕忙卸貨離開，因此點貨驗收通常以點件為主，而抽驗花卉把數與品質等拆箱工作在理貨作業時進行。進行點件工作時，進貨人員直接對照進貨資料表與讀取器所接受到的花卉種類個數及箱數，便可了解進貨的箱數與種類是否正確，確認無誤後將資料傳送至 OP 處理室，進行資料登錄的工作。

與傳統透過進貨資料表的人工點件比較起來，透過射頻辨識技術使得整個進貨過程變得十分迅速。而且常常在貨物件數比對時，由於貨車卸貨不一定是按照進貨資料表卸貨，同一張進貨資料表的花卉也可能由不同貨車運送，透過射頻辨識技術更使得整個錯誤率與不確定性降低許多，也減少許多人力的浪費。



圖 3.1 花市運送花卉貨車進卸貨過程

3.1.2.2 花卉批發市場的理貨流程分析

理貨作業是指對不同產地的花卉拆箱進行抽驗花卉把數與品質等工作，再依其類別整理成不同的拍賣批次並編列拍賣序號[8]。編列拍賣序號使得整個拍賣作業的進行更加方便，每一筆拍賣序號都是獨一無二的，而該筆拍賣序號也只限於當天買賣。對各家花市而言，編碼的方式不盡相同，也無統一的規格。目前編碼的方式是以各家花市方便為原則，除了高雄花卉批發市場是採用產地理貨的方式，台北、台中、台南、彰化等四家花卉批發市場的編列方式較為接近。

高雄花卉批發市場主要是以產地理貨的方式，主要是以供應人序號再付予一個流水號跟子序號，為一筆拍賣序號。以高雄花市的其中一筆拍賣序號為例，AG001001，而 AG001 為供應人代號，001 是依每個供應人所供應的花卉件數往下編列的流水號。其他四家花卉批發市場主要是以四碼及五碼的編碼方式，再加上一個子序號顯示供貨中的件數，例如 01345，前五碼代表立式容器花卉產品序號。

全部進貨資料拍賣序號編碼完成後，登錄的資料會顯示在進貨動態資料顯示板上。顯示資料包含各種花卉今日到貨統計及市場價格行情等，接下來花卉會依拍賣序號依序搬運堆疊上台車，送至拍賣暫存區等候拍賣。

3.1.2.3 花卉批發市場的拍賣流程分析

拍賣作業是指依拍賣順序及開價原則將不同的拍賣批次依序推出並進行拍賣的流程[8]。而於拍賣作業開始前，花卉批發市場會將每日各種花卉的進貨量進行統計，並將資訊顯示在進貨看版上，以提供承銷人作為競價的參考指標。接著各箱花卉由工作人員依序從台車

或拍賣暫存區推上輸送帶之上，每一個輸送帶的前台工作人員共有四名，分別是：助理拍賣員、第一輔助拍賣員、第二輔助拍賣員及主拍賣員。拍賣開始之後，一箱箱的花卉依序往前推進，助理拍賣員透過裝置在輸送帶入口處的助拍電腦，依序將各箱花卉的拍賣序號、特徵等訊息輸入電腦，再傳入主拍電腦中。一旦主拍賣員叫用該序號後，資訊處理部門立即將該拍賣序號的進貨資料顯示在主拍賣員的終端機及拍賣現場的拍賣鐘上，此時該筆序號的花卉會推進到第一輔助拍賣員前。第一輔助拍賣員會將箱子打開，接著推進到主拍賣員前時，第二輔助拍賣員會取出最底層的一把花卉高舉給承銷人觀看，並由主拍賣員以口述的方法給承銷人作參考，隨後主拍賣員會依個人經驗定價供承銷人競標。例如依據線上花卉品質與預設定價，迅速開出起價或是經由電腦依據前三天平均價自動建議起價。

而各家的拍賣鐘亦有所不同，典型的 LED 式拍賣鐘以彰化花市為例，如圖 3.2 所示；單槍式拍賣鐘以高雄花市為例如圖 3.3 所示。



圖 3.2 彰化花市 LED 式拍賣鐘系統運作狀況



圖 3.3 高雄花市單槍投影式拍賣鐘系統運作狀況

於拍賣過程中，主拍賣員通常遵循開高走低的拍賣模式以高於平均價格約 50%左右開

價，電腦會自動依預設的降幅調降單價，直到承銷人按下按鈕後，電腦才會自動停止降價並扣除該承銷人所購之件數[8]。如果該序號仍有剩餘件數存在，系統採用自動跟價模式，此時鐘面的跟價燈會隨之點亮，在設定的時間內承銷人可以此價格繼續跟價。接著電腦會再自動調降，直到最低價格或該序號全數賣完為止，剩餘的則列為殘貨。

對成交的花卉而言，工作人員會將標示成交條碼的射頻辨識標籤黏貼於貨品包裝箱上，其上記載日期、拍賣序號、子序號等資料，子序號是依該拍賣序號下第幾件成交而賦予，其後並將該貨品輸出至堆貨區，等候分貨作業。

對承銷人而言，需在進入拍賣館前結清所有的應繳納之帳款，在承銷人入座後，隨後將手中的承銷卡插入座位旁的讀卡機內，完成入席的動作。隨著花卉依序送出，承銷人可透過桌面上的拍賣人機介面，如圖 3.4，決定以什麼價格、多少數量以及購買什麼花。

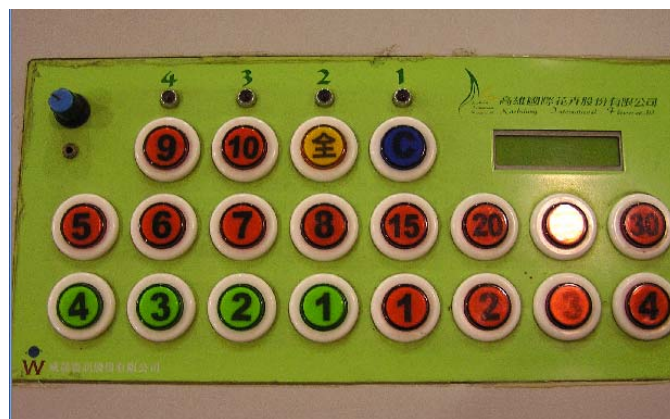


圖 3.4 承銷人拍賣人機介面

整個拍賣過程相當的迅速，透過前場自由席的拍賣系統，快速且有效率的進行直到全部貨品售完為止，而全部貨品也會自動登陸並同步轉移至後場拍賣資訊管理系統，進行結帳作業與列印結帳清單作業。此外承銷人只需以個人承銷卡刷卡便可查詢到個人之成交件數、購買金額、與交易行情等即時資訊。完整的拍賣資訊傳送過程如圖 3.5 所示。

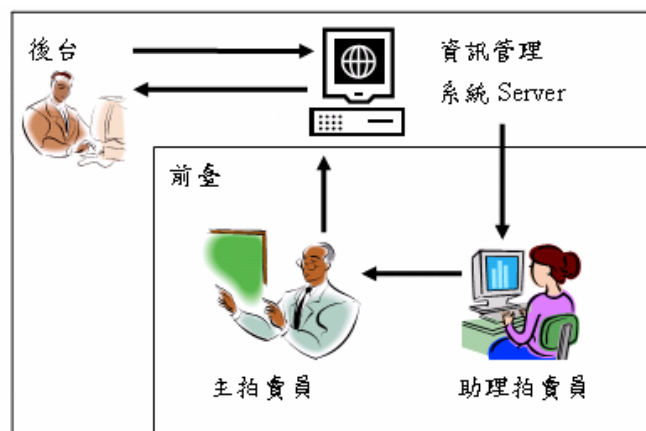


圖 3.5 花卉拍賣資訊傳送過程

3.1.2.4 花卉批發市場的分貨流程分析

分貨作業是指將拍賣完的花卉依拍賣結果分別分到各承銷人的堆貨區的過程[8]。在分貨作業中採用的是自動分貨機進行分貨，如圖 3.6，自動分貨機透過讀取貨品上射頻辨識標籤的承銷商代碼，將貨品依承銷人的不同各自送到各個承銷人的堆貨區，如此一來節省了大量的人力成本，而且加速了分貨作業的速度。



圖 3.6 自動分貨系統

3.1.2.5 花卉批發市場的領貨流程分析

領貨作業是指承銷人到各自的堆貨區將各自標得的貨品領走[8]。領貨管制員依各承銷商代號別，透過讀取射頻辨識標籤中的領貨內容與資訊管理部門所提供的資料加以比對，以避免誤領或未領的作業錯誤。經核對無誤後，將貨品堆置於提貨管制區內，等待承銷人前來提貨。此外，在提貨過程中，管制人員必須對提貨人員之身分進行確認，並以提貨管制電腦對待提貨品之出貨條碼進行掃描，逐一加以核對後放行。

整個過程在承銷人取走貨品之後開始進行整理的工作，而資訊處理室開始彙製交易行情表供市場諮詢參考，並將當天資料傳送至伺服器端進行資料整合。

3.1.3 零售據點的花卉貯存與運送

承銷商透過市場的交易，取得由產地供應的花卉。透過適當的貯運方式運送至零售據點，並送入零售據點的冷藏庫中，以適當的低溫來保持花卉之鮮度與高品質的貯存效果。由零售業者分擔後處理之適度保鮮，使消費者於使用時能賞心悅目、心曠神怡。

透過無線射頻辨識讀取器讀取標籤裡的資訊，並透過網路服務取得所需要的資訊。例如供應商、花卉批發市場、出貨日期等，可快速的進行盤點動作，並可在零售據點設置手持無線射頻辨識讀取器，透過讀取資訊可讓消費者多一些參考資訊，在選擇產品時多個憑藉。

3.2 花卉批發市場資訊儲存架構設計

將資料透過網路的匯集，而後經由資訊轉換系統存入三階正規化資料庫之中稱為網路型三階正規化資料庫，資訊架構如圖 3.7 所示[9]。本節說明花卉批發市場的每日交易資料如何透過檔案轉移服務傳送到伺服器端，並將原始資料透過資料轉移服務轉入三階正規化資料庫中。本節共分四小節，第 3.2.1 節介紹交易資料內容，第 3.2.2 節介紹檔案轉移服務，第 3.2.3 節介紹三階正規化設計。資料轉移服務於附錄五介紹。

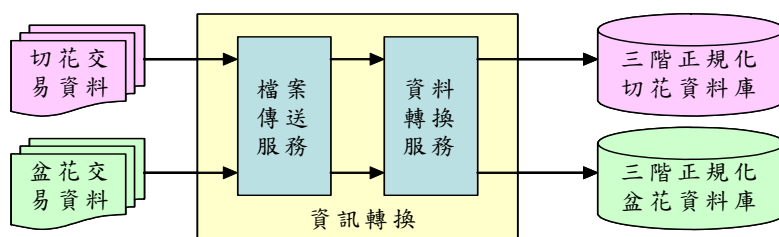


圖 3.7 資訊轉換系統架構[9]

3.2.1 交易資料介紹

花卉批發市場每日都會自動透過軟體，如圖 3.8，將每日的交易資料轉成文字檔，包括當日交易明細(A.txt)、供應商資料檔(Su.txt)、承銷商資料檔(Bu.txt)、花卉資料檔(Pr.txt)，再透過 FTS 軟體自動上傳功將資料傳送到伺服器端。

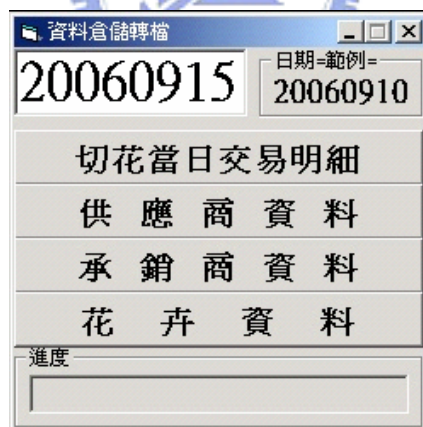


圖 3.8 轉檔軟體

對原始資料檔而言，供應人資料檔儲存了供應人代碼、姓名等基本資料；承銷人資料表則儲存了承銷人代碼、姓名、承銷地區等資訊；花卉資料檔儲存花卉品名代碼、花卉品名、種類等資料；而當日交易明細則儲存交易日期、供應人代碼、承銷人代碼、花卉品名代碼、序號、單價等資料，四個交易資料檔內容如圖 3.9 所示，資料檔格式如附錄一。

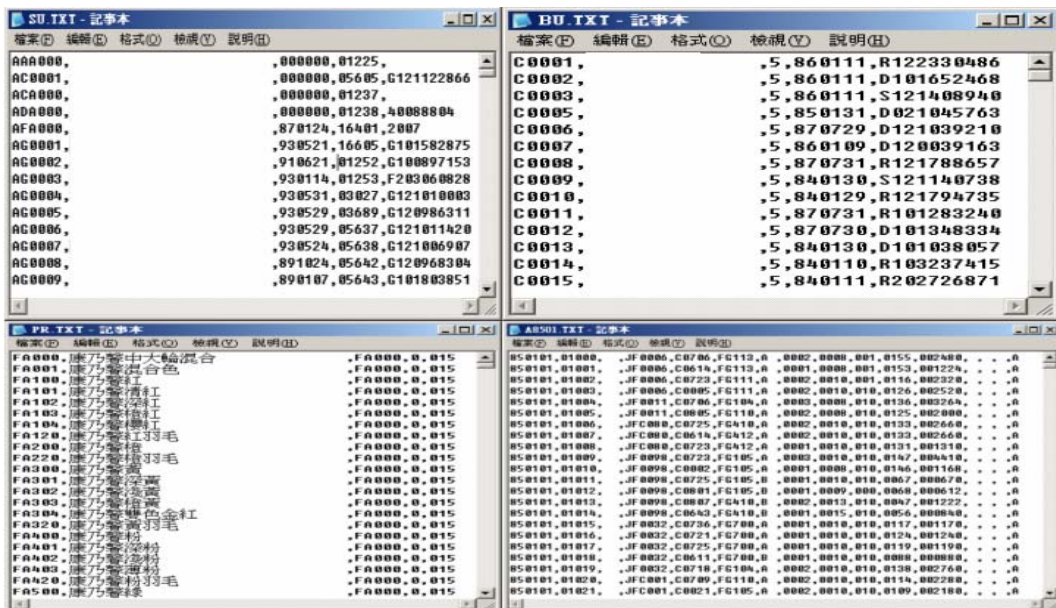


圖 3.9 交易資料原始檔案

3.2.2 檔案轉移服務

就 FTS 軟體而言，其基本畫面如圖 3.10 所示，操作項目主要有(1)連線(2)選擇(3)上傳(4)斷線(5)結束等五項，並提供全自動的服務[11]。

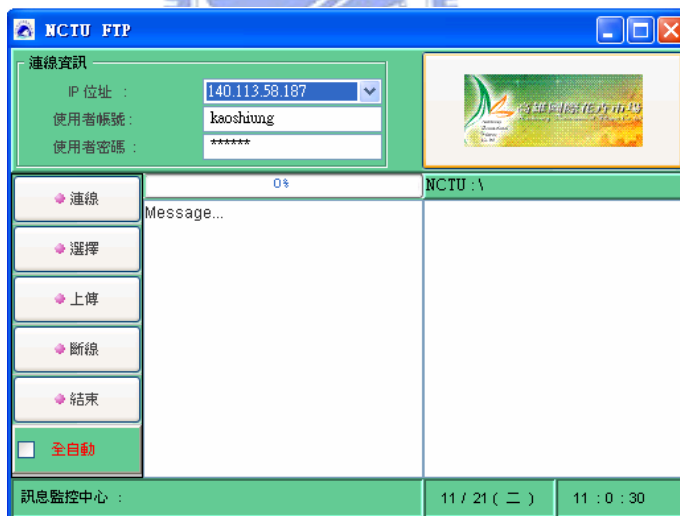


圖 3.10 檔案轉移服務視窗

由於這套軟體是專門為自動化上傳而設計，所以沒有提供下載的服務，在一般上傳的方式並沒有與其他檔案轉移軟體有太大的差異。透過 IP 位址以及使用者帳號、密碼，使用者可以透過『連線』選項對伺服器進行連線，如圖 3.11。



圖 3.11 與伺服器進行連線成功後視窗

透過『選擇』選項選取所需要上傳的檔案，藉由『上傳』選項將資料傳送到伺服器端，完成後透過『斷線』中斷與伺服器端的連線，最後透過『結束』選項關掉介面完成上傳動作。

在自動化方面，透過點取右上角的圖示可對於自動化進行設定，如圖 3.12 所示。遠端路徑為伺服器的路徑，空白則代表連線的初始目錄，而本地路徑則表示欲上傳檔案所在的資料夾。而自動上傳的方式可分為即時上傳與定時上傳，在定時上傳方面，軟體會在每天所設定的時間藉由執行緒自動偵測本地路徑是否有資料，並進行上傳作業，而即時上傳則無時無刻對本地路徑進行偵測。

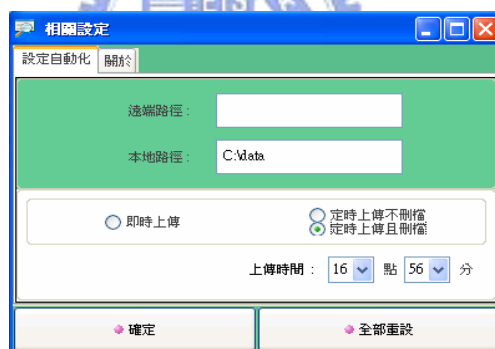


圖 3.12 自動化設定視窗

當上傳檔案成功後，訊息視窗會顯示上傳成功的畫面，如圖 3.13 所示，完成操作作業。

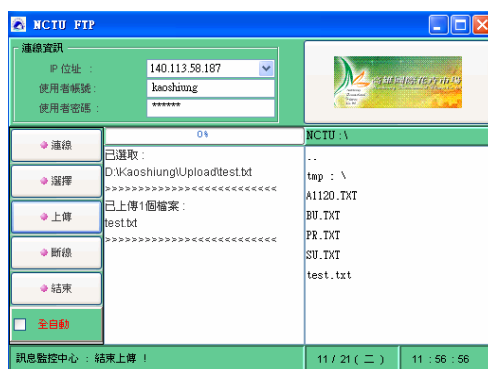


圖 3.13 檔案上傳成功畫面

3.2.3 三階正規化設計

三階正規化的工作不僅可以以最有效力的方式儲存資料、減少記憶體消耗，此外更可以藉由三階正規化資料庫所建立的資料關連性，幫助系統管理者找出原始資料中錯誤及重覆性的資料。

將四個交易資料檔內容透過三階正規化的設計原理，由上而下的分解法，將資料做進一步的資料分割。經過三階正規化後可分成十張資料表，包含供應縣市資料表、供應行政區資料表、供應人資料表、拍賣線資料表、拍賣資料表、承銷人資料表、花卉資料表、承銷區域資料表、花卉等級略表及花卉等級表，並對其建立關聯，即為三階正規化資料庫的架構圖，如圖 3.14。而原始切花交易資料中的供應人資料檔、承銷人資料檔、花卉品名資料檔以及交易資料檔四個資料檔案分別對應至資料庫中的供應人資料表、承銷人資料表、花卉資料表以及拍賣資料表四個事實資料表，其他六個維度資料表則是由這四張表再往下細分。

而十個資料表的各個屬性的詳細說明如附錄二，包含資料表的主鍵、外鍵、資料欄名稱、資料型別、長度、可否為空值及相關說明。

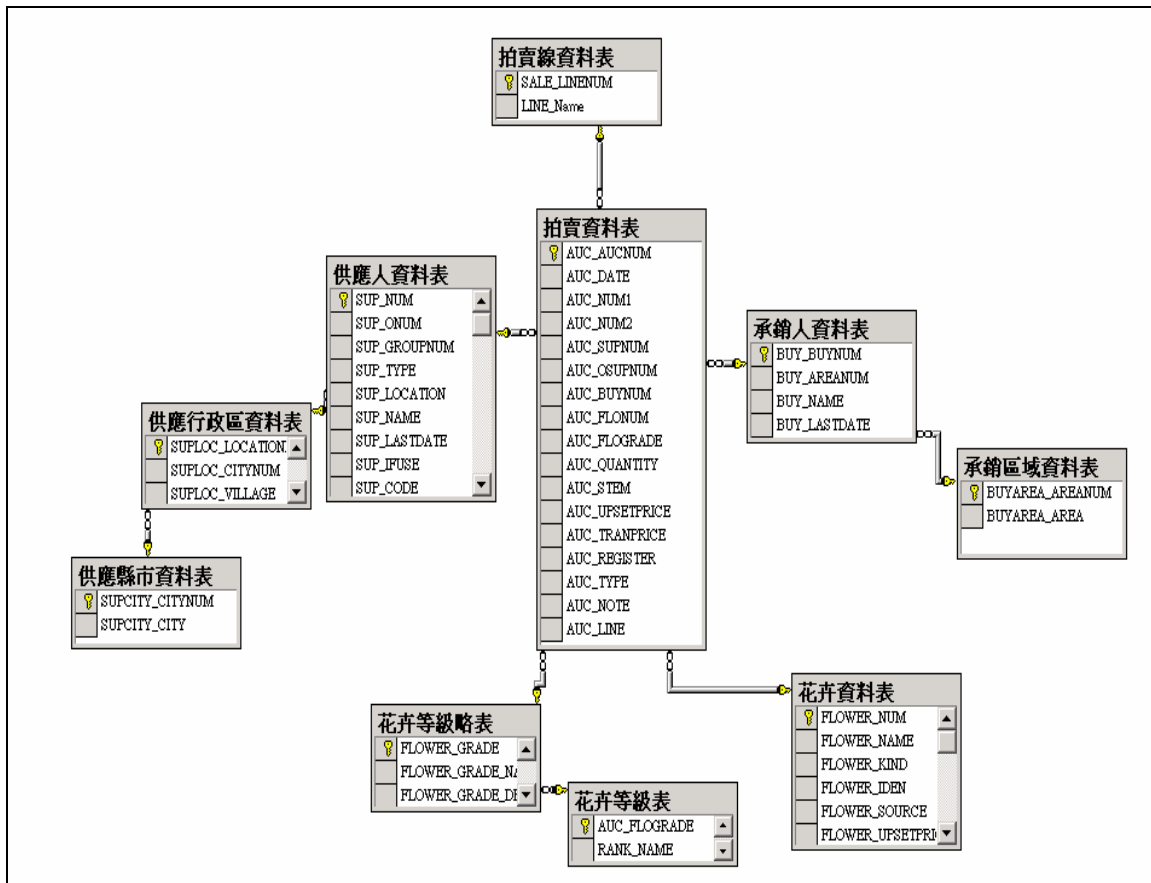


圖 3.14 花卉批發市場三階正規化資料庫架構

3.3 供應商與承銷商的資訊儲存架構設計

在花卉供應鏈上，主要從供應商到批發市場再到承銷商。為了讓整個資訊架構更加完整，必須對供應鏈上各端點建立資訊儲存架構，批發市場的資訊儲存架構已經在第 3.2 節中介紹過，因此本節介紹供應商與承銷商資訊儲存架構的設計。

供應商中包含產地花農、合作社以及農會等，著眼的因素為每一個生產批號下花卉栽種所使用的肥料種類、保存與運輸的處理過程等。圖 3.15 為供應商資料儲存架構的關聯圖，表 3.1 為供應商日誌基本資料表，表 3.2 為供應商施肥紀錄表，表 3.3 為肥料種類紀錄表。

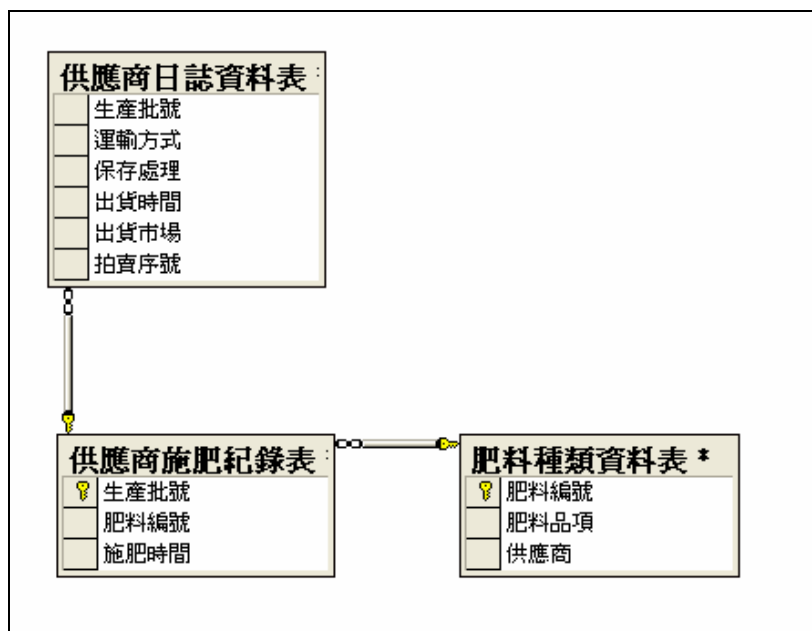


圖 3.15 供應商資料儲存架構關聯圖

表 3.1 供應商日誌基本資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	生產批號	Char		12	生產的識別碼
		運輸方式	Char	◎	30	運輸過程
		保存處理	Char	◎	30	採用什麼方式保存
		出貨時間	Datetime		4	
		出貨市場	Char		10	
		拍賣序號	Char		16	

表 3.2 供應商施肥紀錄表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		生產批號	Char		12	生產的識別碼
	◎	肥料編號	Char		6	所使用的肥料編號
		施肥時間	Datetime		4	

表 3.3 肥料種類資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		肥料編號	Char		6	所使用的肥料編號
		肥料品項	Char		30	所使用的肥料品項
		供應商	Char		30	肥料供應商

承銷商包含花卉零售商或零批商、販運商、大消費戶及花卉出口商等。承銷商著眼的因素為包裝方式的處理、貨物保存的方式以及出貨的對象及方式，如空運外銷到海外。圖 3.16 為承銷商資料儲存架構的關聯圖，表 3.4 為承銷商日誌基本資料表，表 3.5 為承銷商日誌資料表。

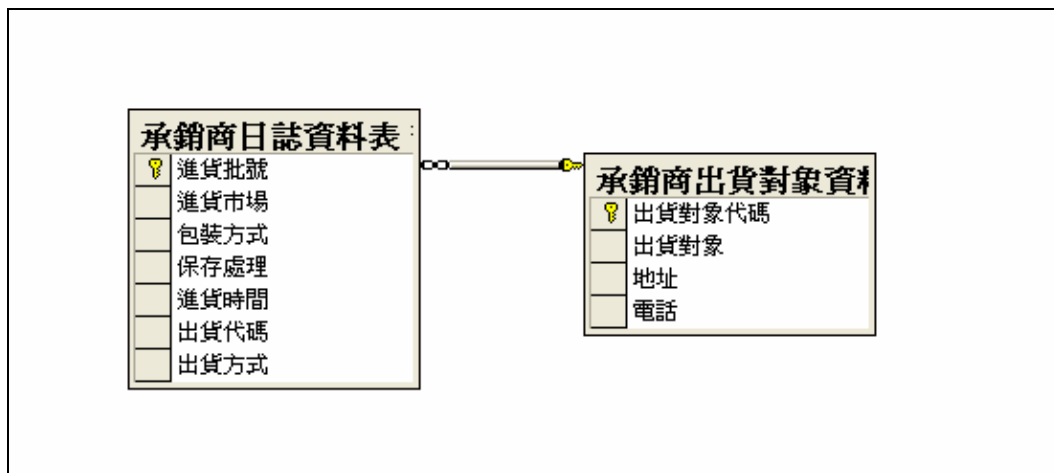


圖 3.16 承銷商資料儲存架構關聯圖

表 3.4 承銷商日誌基本資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		進貨批號	Char		16	拍賣市場的序號
◎		進貨市場	Char		30	拍賣市場
		包裝方式	Char	◎	30	採用什麼方式包裝
		保存處理	Char	◎	30	保存的處理方式
		進貨時間	Datetime		4	
	◎	出貨對象代碼	Char		30	外銷、自用等
		出貨方式	Char	◎	30	

表 3.5 承銷商日誌基本資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		出貨對象代碼	Char		6	
		出貨對象	Char		30	出貨對象姓名
		地址	Char	◎	30	出貨對象地址
		電話	Char	◎	30	出貨對象電話

3.4 花卉識別 Index 的設計

對追溯系統而言，要追溯的商品必須具有單一的識別碼以便在眾多的商品之中分辨出各個商品相關的資料。為了確認號碼的單一性，因此需要依靠一套固定的編碼系統來產生。

目前花卉供應鏈從上游供應商到批發市場再到承銷商，Tradeoff 的過程相當複雜。一個供應商不單單是送花到一種批發市場，而一個承銷商也不只會再同一個批發市場中進行拍賣動作，如圖 3.17 所示。

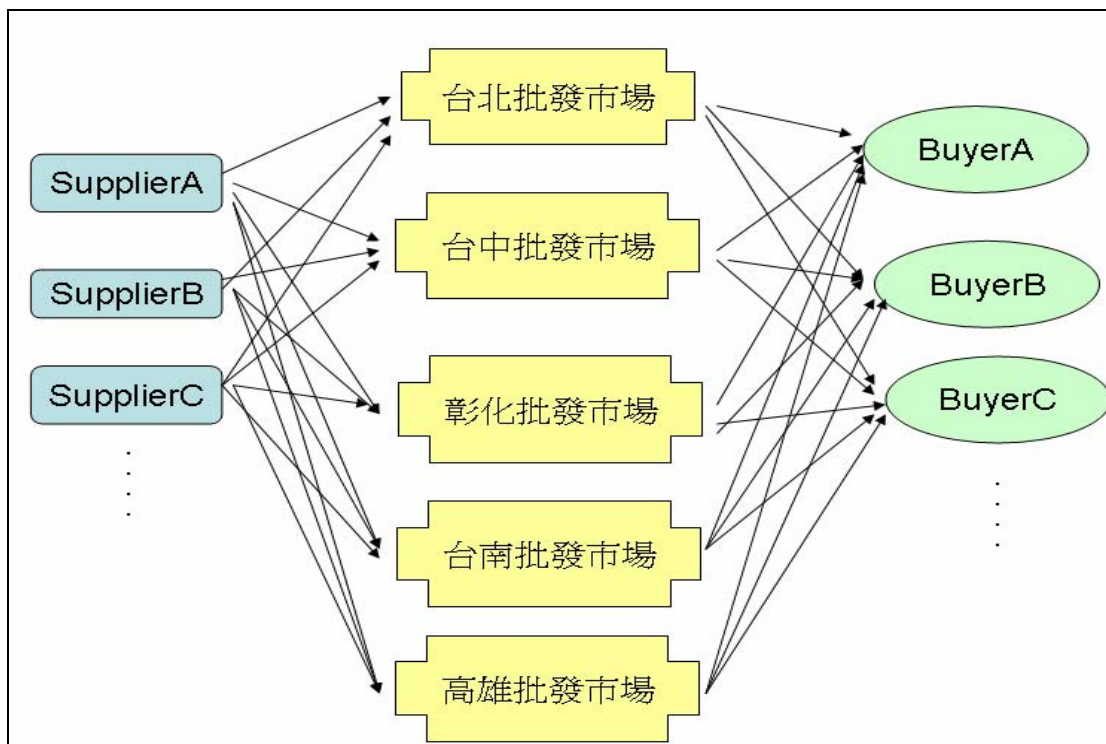


圖 3.17 供應鏈的 Tradeoff

目前批發市場中已經有一套建立拍賣序號的規則，如圖 3.18。前六碼為拍賣當天的日期，接著的五碼為供應此批花卉的供應商序號。最後的五碼，前三碼代表序號，後兩碼則為子序號。序號是依每個供應人所供應的花卉件數向下編列的流水號，子序號則為供應中的件數。

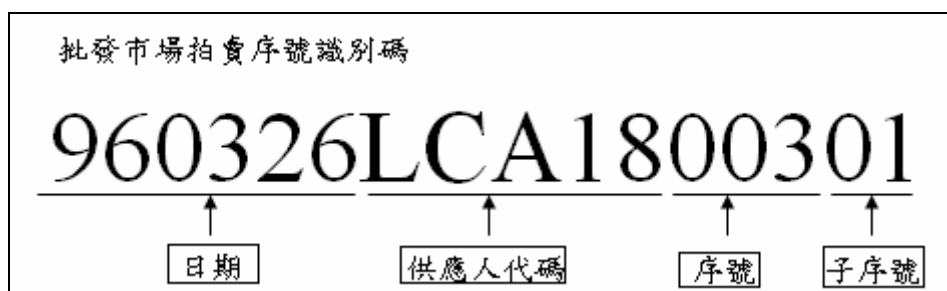


圖 3.18 批發市場拍賣序號識別碼

但是花卉供應鏈中也包含了供應商與承銷商，目前供應商與承銷商並沒有一套固定的編碼系統。對供應商而言，目前提供貨物到批發市場中，所填寫的進貨單中只包含供應人代碼、花卉品名及件數。除此之外供應商日誌中也需要為每筆花卉商品定義一個唯一的識別碼，才能夠識別出每筆花卉商品。而編碼的設計還需要注意到漸進性、及適用性，因為花卉產業為傳統產業，從業人員較為保守，因此在做編碼考慮時應考慮到使用者的接受程序，此外也需要考慮到合乎到產業的特性。圖 3.19 為本研究對於供應商生產批號所設計的編碼系統。前三碼為年份，中間的六碼序號為供應商剛栽種下的序號，而子序號則為收割時所設的流水號。目前承銷商的進貨批號也沒有存在一套編碼系統，最簡單的方式，直接以年份加流水號當作承銷商的進貨批號，所有承銷商日誌資料都以拍賣序號比對的進貨批號為準，如圖 3.20。

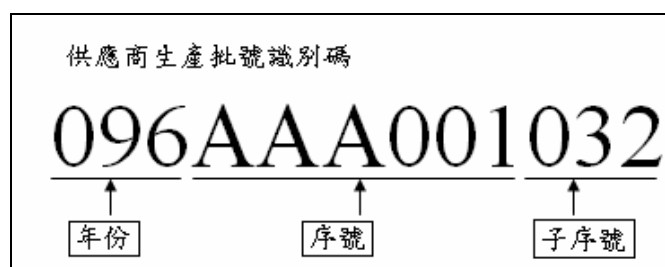


圖 3.19 建議的供應商生產批號識別碼



圖 3.20 建議的承銷商進貨編號識別碼

由於供應鏈中 Tradeoff 的複雜性以及條碼 Index 對於追溯系統的重要性，此外同一筆生產批號，其拍賣序號不一定相同，因此由拍賣序號追溯生產批號時可能會發生錯誤。因此本研究在 Index 設計上對標籤存取供應鏈上各端點的批號，分別包括供應商的生產批號、批發市場的拍賣序號及承銷商的進貨編號，每經過一個端點依序將編號加入條碼之中。此外標籤內容中也包含網路服務中批發市場所提供網路服務的服務名稱。整個 Index 如圖 3.21，前 12 碼固定為供應商的生產批號，接著四碼為批發市場所提供網路服務的的服務名稱，接著 16 碼固定為批發市場的拍賣序號，後 9 碼固定為承銷商的進貨編號。



圖 3.21 標籤內所存取的條碼資訊

第四章 網路服務仲介端與伺服器端的設計

追溯系統的資訊儲存架構設計的部份已在第三章建構完成，而本研究利用網路服務技術取得資料並將資料呈現出來，在本章將說明追溯系統網路服務伺服器端的建構，於第五章再說明客戶端的建構。本系統採用 Apache Axis 的套件建立網路服務，採用的是 JAVA 程式語言，使用的原因一方面 Axis 已經發展成熟，另一方面透過 Axis 一步一步的建構更能夠完整的了解網路服務的內容。

本章主要包含四個部份，第 4.1 節介紹此追溯系統網路服務的大架構；第 4.2 節部份介紹服務仲介端的建立，使用的是微軟 Windows Server 2003 內建的 UDDI 伺服器；第 4.3 節部份介紹 Axis 套件的系統架構與運作模式；第 4.4 節的部份藉由 Apache Axis 套件建立系統的伺服器端，其中包括環境佈署、建置流程規劃、伺服器端程式設計等三方面。內容包括接收商業實體的登錄註冊、網路服務的發佈以及服務需求端查詢適當的服務位址。

4.1 網路服務導向追溯系統的架構

目前花卉供應鏈無法進行資訊的追蹤與追溯，為達此功能因此有了此追溯系統的誕生。第三章已經說明花卉供應鏈結合無線射頻辨識技術的狀況，並說明完整的資訊儲存過程。但是存取於資料庫中的資料並沒有一個媒介供使用者取得，更無從談起追蹤與追溯，因此透過網路服務與 JAVA 程式語言建立一個圖形介面的追溯系統供使用者使用。追溯系統架構如圖 4.1 所示，而虛線裡面為網路服務的部分。

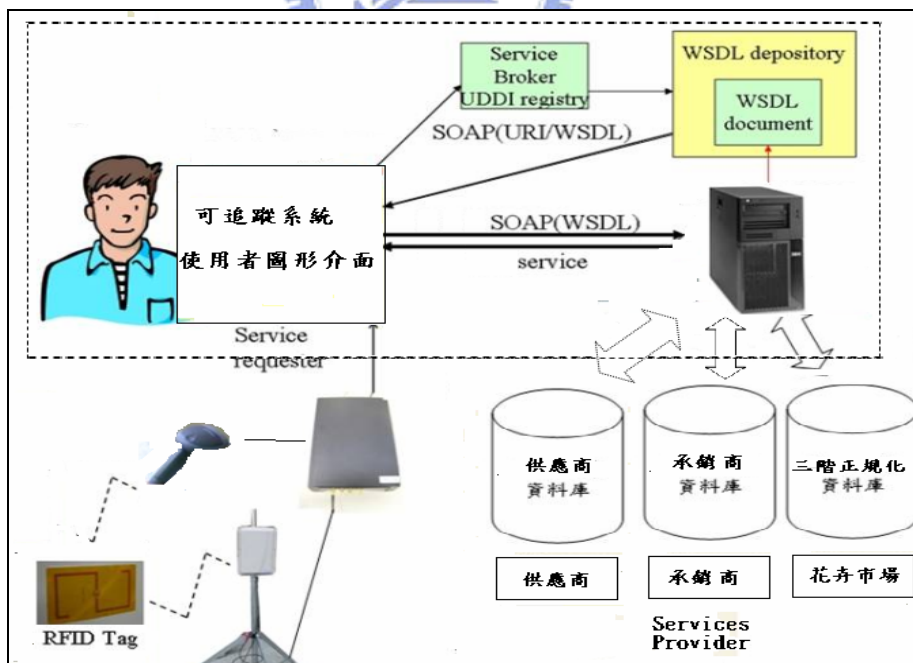


圖 4.1 追溯系統架構

就網路服務而言，在第 2.3 節中已經詳細介紹過，而網路服務技術主要分成服務需求端、服務伺服器端、服務仲介端三者。服務提供者在此是供應鏈中花卉批發市場的業者、供應商與承銷商，每個業者建立屬於自己的網路服務，而服務內容是存取該業者資料庫中的日誌資訊。

透過『網路服務描述語言文件』將提供的服務規格書註冊在服務仲介端；而服務仲介者使用『通用描述、發現與整合』技術，是網路服務架構中的第三方仲介，告知服務需求者某項服務的提供者在網路上的位址；而服務需求者，也就是供應鏈上的各端點(包括供應商、批發市場、承銷商、消費者等)。

而追溯系統的資訊流程，各批發商、花卉市場、承銷商每日將各自的日誌資料轉入各個資料庫，這部份在第三章已經描述過。各業者建立各自的網路服務，服務內容為與其資料庫溝通，並將網路服務佈署到伺服器端，且向仲介端註冊該網路服務，到此為止為追溯系統後端的部份。前端的部份，使用者透過讀取器讀取了標籤的資料(包括批發市場服務名稱、條碼序號)。透過服務名稱向仲介端查詢適合的網路服務，並透過仲介端回傳的批發市場網路服務伺服器端位址，將條碼序號傳送給伺服器端，經過伺服器端對資料庫進行查詢後得到該拍賣商品的批發市場資料與供應商及承銷商的服務名稱。再透過供應商與承銷商的服務名稱向仲介端取得供應商與承銷商的服務位址，透過其網路服務即可取得供應商與承銷商的產銷日誌資料。

整個網路服務流程依序為(1)服務伺服器端向仲介端註冊服務、(2)服務需求端向服務仲介端查詢服務位址、(3)服務需求端向服務伺服器端進行服務三項。關於服務伺服器端向仲介端註冊服務在第 4.2 節進行說明，需求端透過射頻辨識讀取到標籤的條碼在第 5.1 節進行說明，而服務需求端向服務仲介端查詢服務位址則在第 5.2 節進行說明。服務需求端向服務伺服器端取得服務又可分為需求端傳送請求給伺服器端，伺服器端得到請求並對資料庫進行查詢；再將得到的結果傳送回需求端，再透過使用者圖形介面顯示出來，需求端的部分由第 5.3 節進行說明，伺服器端的部份則在第 4.4 節中探討。



4.2 網路服務仲介端的建置

欲公布某項網路服務必須靠 UDDI 伺服器的幫助。服務提供者向 UDDI 伺服器註冊其服務，而服務需求者再藉由搜尋來找到自己所需的服務。對 UDDI 伺服器而言，對內必須負責管理與營運註冊中心，對外必須接受登錄、發佈及查詢等作業，而每個註冊中心還會彼此交換註冊資訊以達到資料的整合。

而目前 UDDI 版本很多，比較著名的包括微軟正式商用註冊中心、IBM UDDI 註冊中心等。本研究使用的是個人架設的 UDDI 伺服器，採用的是微軟 Windows Server 2003 內建的 UDDI 服務，架設位置在交通大學的實驗室伺服器上。透過個人架設更可以控管後端，除此之外對於 UDDI 的架構更能深刻的認識，亦不需要因為伺服端的要求而有所限制。首頁如圖 4.2 所示，共包含搜尋、發行、協調等三項，跟微軟正式商用註冊中心相較之下，除了多了協調其他介面相當相似。

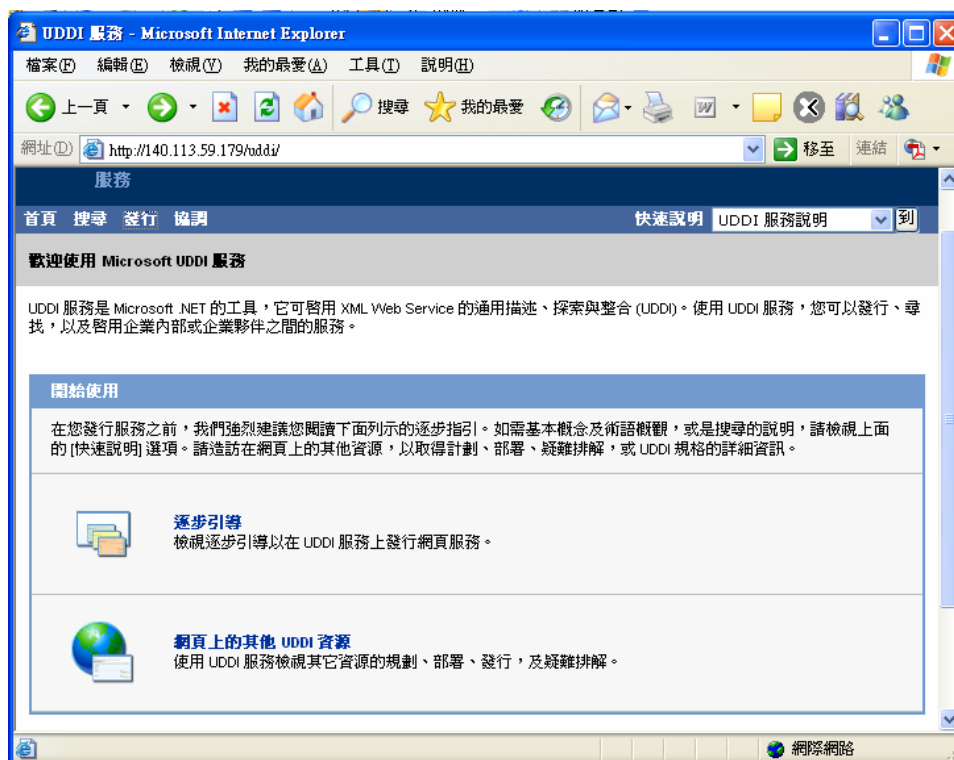


圖 4.2 Microsoft UDDI 建置 Server 首頁

所有佈署於伺服端的網路服務，都得將服務註冊到 UDDI 伺服器上，才能發佈其網路服務，而服務在表 1.2 中已經描述過，在此不再撰述。而本節以『KSflower』服務個體為例，說明註冊的過程，其他則雷同『KSflower』。

發行的部份，首先先建立一個商業實體，名為『KSflower』，給予一個服務，名為『kifc』。而服務的繫結為『http://140.113.59.178:8080/axis/services/KSflower』，該繫結資訊連結到名為『KStModel』的 tModel。發行過程頁面過多加上首頁中有詳細的介紹，在此跳過。發行完後的頁面大致如圖 4.3，而此網路服務發行於 UDDI 伺服器的資料如表 4.1。

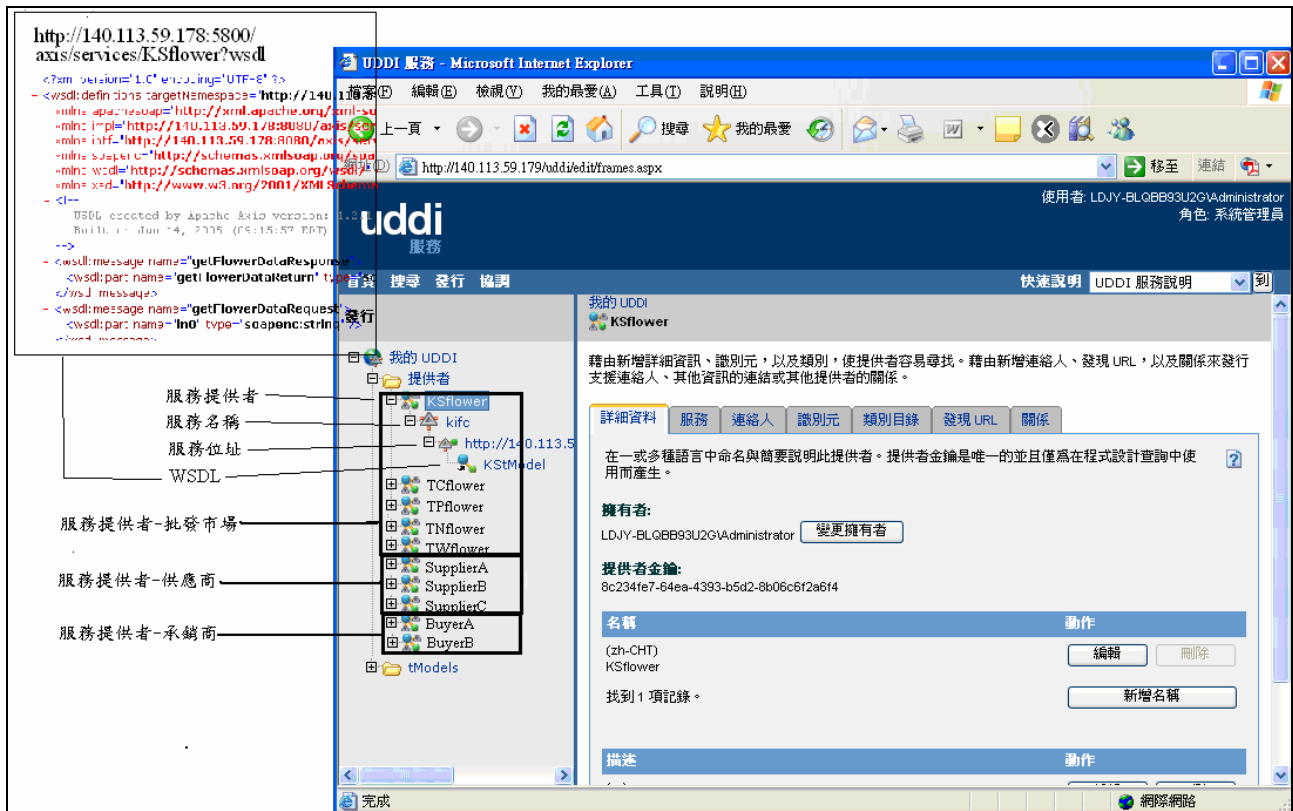


圖 4.3 UDDI 發行網路服務完成頁面

表 4.1 網路服務註冊資料

資料結構模型	元素	內容
businessEntity	Name	KSflower
	description	高雄批發市場
	categoryBag	分類配置 Microsoft-com:visualstudio-net 機碼名稱 data
businessService	Name	Kifc
	description	Ksflower to database
bindingTemplate	accessPoint	http://140.113.59.178:5800/axis/services/KSflower
	description	KSflower to db's access point
tModel	name	KstModel
	Description	KSflower tModel
	overviewDoc	http://140.113.59.178:5800/axis/services/KSflower?wsdl
	categoryBag	分類配置 Microsoft-com:visualstudio-net 機碼名稱 data

協調的部份包括分類配置、統計與資料匯入。透過分類配置，管理員可以透過分類管理檢視與管理 UDDI 服務資料；統計則可以檢視目前站上所有的資料，包括有多少個商業實體、服務或是 tModel 等的註冊；透過匯入能夠將過去備份的資料重新匯進系統，防止系統壞損產生資料不見的狀況。透過網頁進行搜尋的部份與其他的 UDDI 系統類似，可依自己所要搜尋的類別進行查詢，共分為依類別瀏覽、依服務、依提供者、依 tModel 四種。

4.3 Axis 的系統架構與運作模式

Axis 是 Apache 組織提供的一個 SOAP 引擎，它提供了完整的 SOAP 解決方案，透過 Axis 能夠輕易完成伺服器端、客戶端等的網路服務開發。Axis 的重要特性包括速度、彈性、穩定度、元件導向的佈署方式、傳輸框架以及對 WSDL 的支援性。對速度而言，Axis 採用 SAX 技術來解析，速度較快。對彈性而言，Axis 允許使用者自由的擴展其功能，因此使用者可以自由加入想要增加的特性。對穩定度而言，Axis 定義了一套穩定的發佈介面，這套發佈介面並不會隨著 Axis 的更新而更新，除非相關的標準產生了改變。對元件導向的佈署方式而言，Axis 內部的處理器使所有導向的元件以處理器的方式很容易的組合與重覆使用。對傳輸框架而言，在需求端與回應端都可以透過多種傳輸協定溝通，而且傳輸部分與 Axis 的核心引擎是完全分離的。對 WSDL 的支援性而言，Axis 不僅支援 SOAP 標準，更完整的支援 WSDL，並提供 WSDL 工具將 WSDL 轉換成 JAVA 程式碼，或是由 JAVA 程式碼轉換成 WSDL 檔，使得開發程式十分方便，除此之外更可由佈署內容自動產生 WSDL 文件。

Axis 的內部運作是一連串的訊息傳遞，並循著一定的程式循序漸進，而程序只會依佈署規劃(Deployment Configuration)及伺服器端或客戶端而有所改變。在每個端點的處理器會解讀訊息並進行處理，並將處理過後的訊息封裝成訊息上下文(MessageContext)再傳到下個端點。Axis 引擎中包含傳輸層(Transport)、基本層(Global)與服務層(Service)三層。

伺服器端的訊息傳遞過程如圖 4.4 所示，小的圓滾桶代表處理器(Handler)，而大的圓滾桶則為一個鏈式結構(Chain)。當聆聽者(Transport Listener)收到訊息時，將呼叫請求的相關資訊封裝進訊息上下文，依傳輸層、基本層、服務層順序傳遞與進行處理，最後傳送回目標服務(Target Service)。目標服務會傳送一個訊息給聆聽者表示傳輸完成，完成伺服器端訊息傳遞的一個週期。

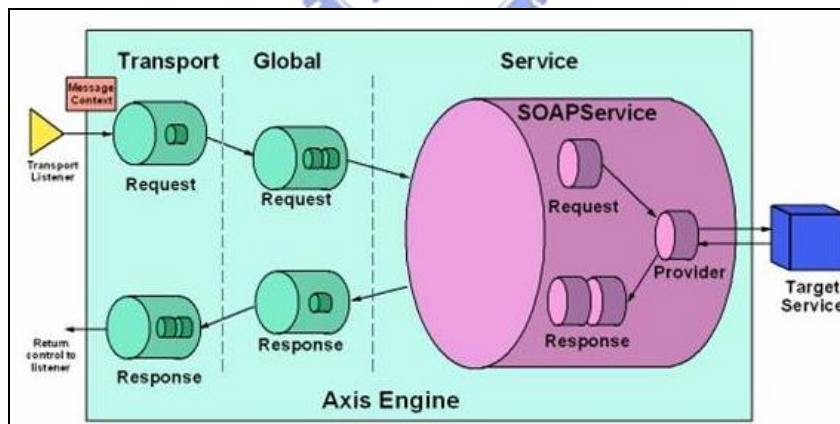


圖 4.4 Axis 用於伺服器端的運作架構[16]

服務需求端的訊息傳遞過程與伺服器端類似，如圖 4.5。所不同的是服務需求端是由客戶端應用程式(Client Application)直接發出訊息，並依序通過服務層、基本層、傳輸層，傳達給目標服務，而目標服務再將處理過的訊息以反向傳送回來，完成需求端訊息傳遞的一個週期。

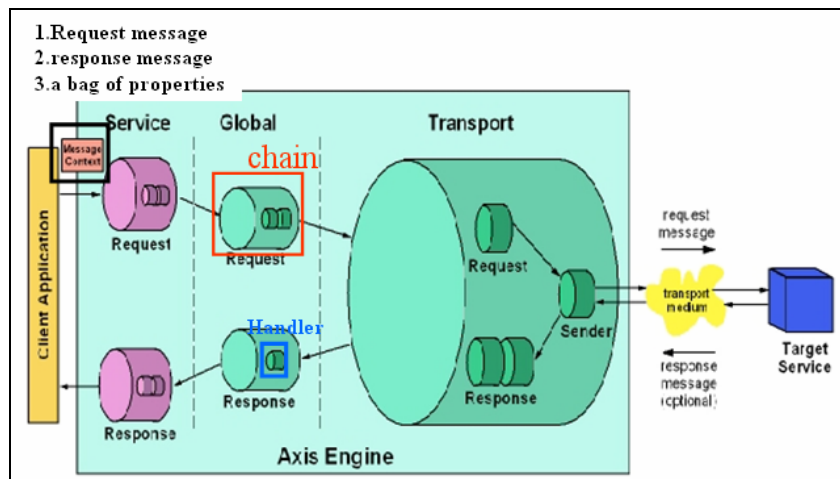


圖 4.5 Axis 用於服務需求端的運作架構[16]

Axis 對於 WSDL 的支援相當的完善，主要的支援可以分成三部分：`?wsdl`、WSDL2JAVA、JAVA2WSDL。對 `?wsdl` 而言，當服務佈署於 Axis 之上，使用者可以透過服務的網址並在其後加上 `?wsdl`，即可取得該服務的 WSDL 文件。取得 WSDL 文件相當方便，藉此觀察服務的內容較為簡易。對 WSDL2JAVA 工具而言，使用者可以透過這套工具將 WSDL 檔依繫結規則轉換成 JAVA 程式。相反地，對 JAVA2WSDL 工具，則是將 JAVA 程式轉換成 WSDL 文件。透過 WSDL2JAVA 及 JAVA2WSDL 兩個工具開發自己的客戶端或伺服器端服務變得十分簡易，也使得網路服務的實現難度大為降低。而 WSDL 與 JAVA 的繫結規則依伺服器端與客戶端而有所不同，伺服器端的繫結規則如表 4.2 所示，客戶端則如表 4.3 所示。

表 4.2 WSDL 與 Java 的伺服器端繫結規則[16]

WSDL clause	Java generated
each binding	a skeleton class
	an implementation template class
each service	a deploy.wsdd file
	a undeploy.wsdd file

表 4.3 WSDL 與 Java 的客戶端繫結規則[16]

WSDL clause	Java generated
each entry in the type section	a java class
each portType	a java interface
each binding	a stub class
	a skeleton class
	an implementation template class
each service	a service interface
	a service implementation
	a deploy.wsdd file
	a undeploy.wsdd file

4.4 網路服務伺服端的建置

本節中介紹網路服務伺服端的建置過程與服務提供者(供應商、批發市場、承銷商)所提供的服務內容介紹。第 4.4.1 介紹伺服端的建置流程，第 4.4.2 節說明各服務的內涵。

4.4.1 伺服端的建置流程

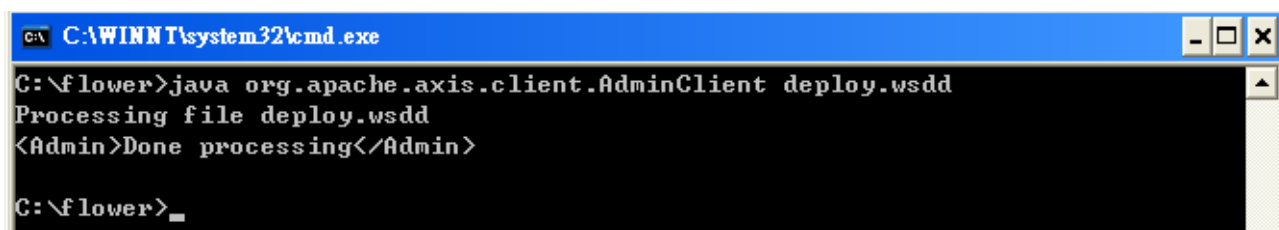
伺服端的建置，首先是伺服器的安裝部份。由於 Axis 是一套附屬於 Servlet Engine 上進行網路服務的專案，因此在安裝 Axis 前必須先安裝 Servlet Engine。Axis 支援大部份的 Servlet Engine，在此使用官方網站上推薦的 Jakarta Tomcat。安裝完後並將 Axis 中 webapps 裡的資料貼到 Tomcat 裡，並將 Axis 的 jar 檔貼到 JAVA 預設的 classpath，安裝即完成。安裝程序也會由不同的需求而有所改變，在首頁的 Validation 頁面會顯示目前安裝的狀況及缺少的套件。Axis 的伺服器端佈署方式可分為兩種：即時佈署(Instant Deployment)及客製佈署(Customer Deployment)。即時佈署又稱為 JWS 佈署，是 Axis 中最簡單的佈署方式。只要將 JAVA 程式附檔名改成 .jws，並移到 axis 資料夾中便佈署完成。但由於較為簡單限制也較多，必須有伺服端的原始碼才能使用。更因為佈署過程過於簡單，一旦佈署失敗，除錯的過程也會比較複雜。除此之外程式不能封裝，也就是不能有 package 的限制；客製佈署又稱 WSDD(Web Service Deployment Descriptor)佈署，透過 deploy.wsdd 將 class 檔佈署到伺服器上，也有 undeploy 的機制，屬於 Axis 中較正式的佈署方法。客制化佈署的優點在於建立方法較為嚴謹，可以使用的服務內容也比較多，因此擴充性會比即時佈署的方法高。WSDL2JAVA 產生的程式檔也都會附加 deploy.wsdd，不需要程式的原始碼及循序漸進的佈署過程為其優點。

而伺服器端的建構過程也分為兩種，一種為先建立伺服端的程式，另一種為先建立 WSDL 檔。直接建立伺服端程式的方法較為直接，透過程式內容直接描述網路服務所要進行的內容；透過 WSDL 檔佈署網路服務較常用於透過別人的網路服務改寫而成為新的網路服務時使用。

在程式與資料庫的連線部分採用的是 JDBC API，透過 JDBC 可以讓程式開發人員建構 SQL 敘述，並將其嵌入 Java API 的呼叫中，藉此可使資料庫與 Java 的應用程式之間的資訊互相傳遞。本研究使用 jTDS 專案中的 JDBC API，版本為 2005 年 6 月 16 日釋出的 1.1 版。

本研究在伺服端建置部分採用客製化佈署，並採用先建立伺服端程式的方式。程式內容為當伺服端接收到需求端傳送過來的訊息，依訊息所給予的條件到資料庫進行查詢並將資料回傳給需求端，具體的程式碼見附錄三。本節伺服端建置部分以 KSflower 說明，其他雷同 KSflower 的佈署方式。

程式編譯完成後，利用 AdminClient 工具透過 deploy.wsdd 將服務佈署在 Tomcat Server 上，如圖 4.6。並將 class 檔貼到 Tomcat 伺服器的 Axis 資料夾中，再透過連結頁面 <http://140.113.59.178:8080/axis/services/KSflower>，驗證如圖 4.7 的畫面，表示服務佈署成功。



```
C:\WINNT\system32\cmd.exe
C:\flower>java org.apache.axis.client.AdminClient deploy.wsdd
Processing file deploy.wsdd
<Admin>Done processing</Admin>
C:\flower>
```

圖 4.6 利用 AdminClient 執行 deploy.wsdd

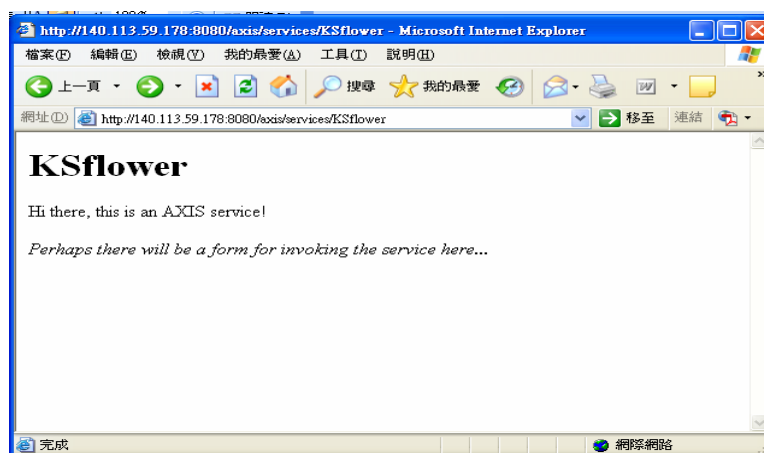


圖 4.7 服務佈署成功驗證畫面

4.4.2 網路服務的內含說明

本研究中具有很多個網路服務，供應鏈中從上游到下游的每個供應商、承銷商、批發市場都有自己的網路服務。以批發市場來說，各家市場擁有自己的網路服務，因此共有五項網路服務，包括 TPflower、TCflower、TNflower、TWflower、KSflower。而每個服務也有四個函式，包括 getAucData、getSupplierData、getBuyerData、getFlowerData。對供應商來說，每個供應商都有自己的網路服務，例如供應商 A 有網路服務 SupplierA，而服務內容是對供應商 A 所提供的供應日誌資料庫進行存取，而其服務具有兩個函式分別為 getDateData 和 getManureData。承銷商則雷同供應商，具有自己的網路服務，服務內容為對自己的承銷日誌資料庫做存取，而承銷商的服務也具有一個函式為 getDateData。表 4.4 整合所有服務的服務個體、名稱以及內容和所連節的資料庫。

表 4.4 所有網路服務內容整合

服務提供者	服務個體	服務方法	服務內容	連結資料庫
供應商	SupplierA、SupplierB、SupplierC...	getDateData	取得供應商日誌	各別供應商資料庫
		getManureData	取得施肥日誌	
批發市場	TPflower、TCflower、TNflower、TWflower、KSflower	getAucData	取得市場資訊	各別批發市場的資料庫
		getSupplierData	取得供應商資料	
		getBuyerData	取得承銷商資料	
		getFlowerData	取得花卉資料	
承銷商	BuyerA、BuyerB...	getDateData	取得承銷商日誌	各承銷商資料庫

在第 4.4.1 節中提到過，本研究網路服務的建立方式為建立 Java 程式碼，再將程式檔轉為網路服務。批發市場的網路服務功能主要透過服務對批發市場資料庫進行存取，圖 4.8 為批發市場網路服務內容程式與批發市場資料庫欄位的對應。批發市場的網路服務共包含四個服務，分別為供應商資料、批發市場資料、承銷商資料及花卉資料，完整的程式碼如附錄三。

供應商的網路服務功能主要透過服務對供應商日誌資料庫進行存取，圖 4.9 為供應商網路服務內容程式與供應商日誌資料庫欄位的對應。供應商共包含兩個服務，分別為供應商日誌及供應商施肥紀錄。

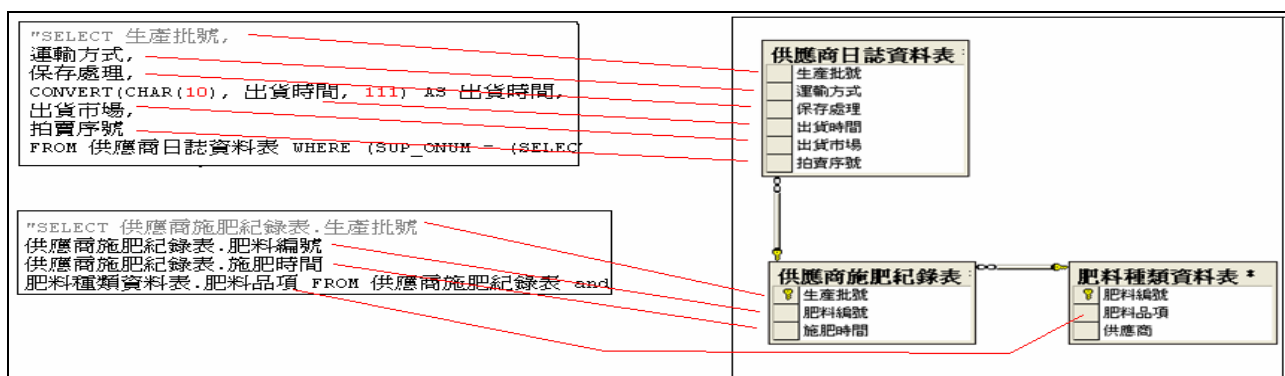


圖 4.9 供應商網路服務內容程式與供應商日誌資料庫對應

承銷的網路服務主要透過服務對承銷商日誌資料庫進行存取，圖 4.10 為承銷商網路服務內容程式與承銷商日誌資料庫欄位的對應。承銷商的網路服務包含一個服務，為承銷商日誌。

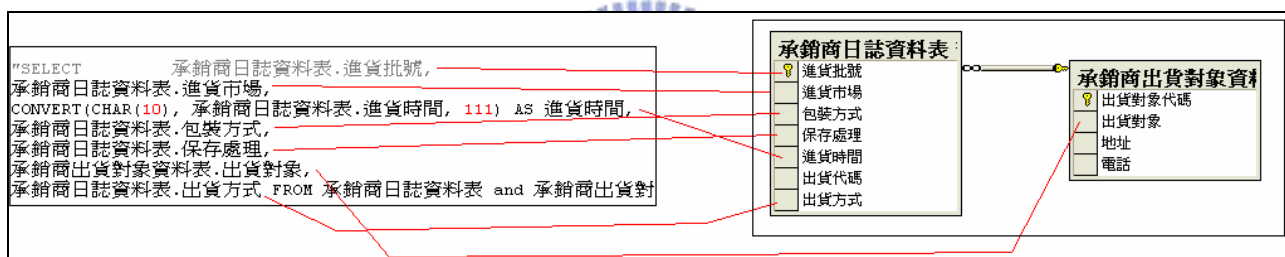


圖 4.10 承銷商網路服務內容程式與承銷商日誌資料庫對應

而供應商、批發市場、承銷商所提供的網路服務都具有各自的 WSDL 檔，如圖 4.11、圖 4.12 及圖 4.13。WSDL 是網路服務的描述語言，透過 WSDL 的假構可以獲取更多的使用細節。WSDL 文件由一系列標準化的元素所組成，這些元素可以描述出服務在伺服器端與客戶端的交換方式。由於本研究所有的網路服務都是針對資料庫做存取，因此將各個 WSDL 檔的節點資訊整合成表 4.5，並同時將節點資訊分別標示於圖 4.11、圖 4.12、圖 4.13 中。

表 4.5 WSDL 文件設計要點與對應元素

規格需求	包含訊息的節點名稱	圖 4.11、4.12、4.13 對應節點序號
說明使用的編碼、規格等技術以及這些技術參考的對應網址	wsdl definition	1
服務伺服器端輸入與輸出的變數名稱	wsdl message 中分別有 Response 與 Request	2
執行服務的呼叫函式名稱	operation name	3
存取服務的位址	wsdlsoap address location	4

WSDL 檔的架構大致可分為 definitions、message、portType、binding、service 五個部分。本例以高雄批發市場的 WSDL 做說明，如圖 4.12。Definition 為 WSDL 文件的根元素，所有文件的內容都包含在其中；message 說明哪些資訊需要傳輸，getFlowerDataRequest 送出一個名為 in() 字串，getFlowerDataResponse 則回傳一個名為 getFlowerDataReturn 的字串；PortType 定義一連串網路服務傳送與接收的作業，提供呼叫的函數為 getFlowerData，輸入與回應的訊息分別為 getFlowerDataRequest 及 getFlowerDataResponse；Binding 說明通訊協定間鏈結方式，採用 http，輸入與輸出的 namespace 分別為 http://DefaultNamespace 及 http://140.113.59.178:8080/axis/services/KSflower；service 說明此網路服務進行 KSflowerService 這項服務，且繫結到 KSflowerSoapBinding 這項 Binding，而服務位址為 http://140.113.59.178:8080/axis/services/KSflower。

```

<?xml version="1.0" encoding="UTF-8" ?>
<wsdl:definitions targetNamespace="http://localhost:8080/axis/services/Supplier" xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/Supplier" xmlns:intf="http://localhost:8080/axis/services/Supplier"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!--
    WSDL created by Apache Axis version: 1.2.1
    Built on Jun 14, 2005 (09:15:57 EDT)
  -->
  <wsdl:message name="getDateDataResponse">
    <wsdl:part name="getDateDataReturn" type="xsd:string" />
  </wsdl:message>
  <wsdl:message name="getDateDataRequest">
    <wsdl:part name="in0" type="xsd:string" />
  </wsdl:message>
  <wsdl:portType name="Supplier">
    <wsdl:operation name="getDateData" parameterOrder="in0">
      <wsdl:input message="impl:getDateDataRequest" name="getDateDataRequest" />
      <wsdl:output message="impl:getDateDataResponse" name="getDateDataResponse" />
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="SupplierSoapBinding" type="impl:Supplier">
    <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
    <wsdl:operation name="getDateData">
      <wsdlsoap:operation soapAction="" />
      <wsdl:input name="getDateDataRequest">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace" use="encoded" />
      </wsdl:input>
      <wsdl:output name="getDateDataResponse">
        <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/services/Supplier"
          use="encoded" />
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="SupplierService">
    <wsdl:port binding="impl:SupplierSoapBinding" name="Supplier">
      <wsdlsoap:address location="http://localhost:8080/axis/services/Supplier" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
  
```

圖 4.11 供應商提供的網路服務 WSDL 檔

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="http://localhost:8080/axis/services/KSflower" xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/KSflower" xmlns:intf="http://localhost:8080/axis/services/KSflower"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <!--
  WSDL created by Apache Axis version: 1.2.1
  Built on Jun 14, 2005 (09:15:57 EDT)
-->
+ <wsdl:message name="getSupplierDataRequest">
- <wsdl:message name="getAucDataRequest">
  <wsdl:part name="in0" type="xsd:string" /> (2)
</wsdl:message>
+ <wsdl:message name="getFlowerDataRequest">
+ <wsdl:message name="getBuyerDataResponse">
+ <wsdl:message name="getBuyerDataRequest">
+ <wsdl:message name="getSupplierDataResponse">
+ <wsdl:message name="getFlowerDataResponse">
- <wsdl:message name="getAucDataResponse">
  <wsdl:part name="getAucDataReturn" type="xsd:string" /> (2)
</wsdl:message>
- <wsdl:portType name="KSflower">
- <wsdl:operation name="getAucData" parameterOrder="in0">
  <wsdl:input message="impl:getAucDataRequest" name="getAucDataRequest" />
  <wsdl:output message="impl:getAucDataResponse" name="getAucDataResponse" />
</wsdl:operation>
+ <wsdl:operation name="getSupplierData" parameterOrder="in0">
+ <wsdl:operation name="getBuyerData" parameterOrder="in0">
+ <wsdl:operation name="getFlowerData" parameterOrder="in0">
</wsdl:portType>
- <wsdl:binding name="KSflowerSoapBinding" type="impl:KSflower">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="getAucData">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="getAucDataRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace" use="encoded" />
</wsdl:input>
- <wsdl:output name="getAucDataResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/services/KSflower"
  use="encoded" />
</wsdl:output>
</wsdl:operation>
+ <wsdl:operation name="getSupplierData">
+ <wsdl:operation name="getBuyerData">
+ <wsdl:operation name="getFlowerData">
</wsdl:binding>
+ <wsdl:service name="KSflowerService"> (4)
</wsdl:definitions>

```

(1)

(2)

(2)

(3)

(4)

圖 4.12 高雄批發市場提供的網路服務 WSDL 檔

```

<?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions targetNamespace="http://localhost:8080/axis/services/Buyer" xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://localhost:8080/axis/services/Buyer" xmlns:intf="http://localhost:8080/axis/services/Buyer"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsoap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
- <!--
  WSDL created by Apache Axis version: 1.2.1
  Built on Jun 14, 2005 (09:15:57 EDT)
-->
- <wsdl:message name="getDateDataResponse">
  <wsdl:part name="getDateDataReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:message name="getDateDataRequest">
  <wsdl:part name="in0" type="xsd:string" />
</wsdl:message>
- <wsdl:portType name="Buyer">
- <wsdl:operation name="getDateData" parameterOrder="in0">
  <wsdl:input message="impl:getDateDataRequest" name="getDateDataRequest" />
  <wsdl:output message="impl:getDateDataResponse" name="getDateDataResponse" />
</wsdl:operation>
</wsdl:portType>
- <wsdl:binding name="BuyerSoapBinding" type="impl:Buyer">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
- <wsdl:operation name="getDateData">
  <wsdlsoap:operation soapAction="" />
- <wsdl:input name="getDateDataRequest">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace" use="encoded" />
</wsdl:input>
- <wsdl:output name="getDateDataResponse">
  <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://localhost:8080/axis/services/Buyer"
  use="encoded" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
- <wsdl:service name="BuyerService">
- <wsdl:port binding="impl:BuyerSoapBinding" name="Buyer">
  <wsdlsoap:address location="http://localhost:8080/axis/services/Buyer" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

(1)

(2)

(3)

(4)

圖 4.13 承銷商提供的網路服務 WSDL 檔

第五章 網路服務客戶端的設計

在第三章及第四章已經分別介紹過了供應鏈資訊轉入資料庫中及追溯系統伺服端的部份，到此為止追溯系統後端已建構完成。本章描述追溯系統前端架構，並說明前端與後端如何溝通，也就是客戶端的部份。

服務客戶端主要流程可分成三個部份：客戶端利用 Socket 透過射頻讀取器取得標籤中的內容，包括服務名稱與條碼序號，這部份在 5.1 節進行說明，另外也包括射頻辨識系統軟硬體的設備與操作。透過取得的服務名稱，服務需求端對服務仲介端查詢適合的網路服務，並得到該網路服務位址，這部份在 5.2 節進行說明。需求端透過服務位址及條碼序號與伺服端溝通，進行該網路服務作業，得到相關於該條碼的資料，這部份在第 5.3 節說明。而第 5.4 節說明整個需求端介面的操作。

5.1 射頻辨識讀寫功能的設計

對追溯系統而言最重要的是條碼的內容，有了條碼的資訊才能進行後續的追溯行為。而花卉追溯系統的載具使用的是射頻辨識標籤，因此本節介紹本研究中射頻辨識系統的部份，包括整個系統架構、硬體以及讀寫功能等軟體的設計。本節共分做兩小節，5.4.1 為射頻辨識硬體架構，5.4.2 則介紹射頻辨識軟體架構。

5.1.1 射頻辨識硬體架構

圖 5.1 是一個標準的射頻辨識系統的硬體架構圖，透過連結讀取器的天線所放出的電磁波與貼在容器或商品上的標籤進行溝通，而客戶端的應用程式透過通訊協定與讀取器溝通得到條碼的資訊。

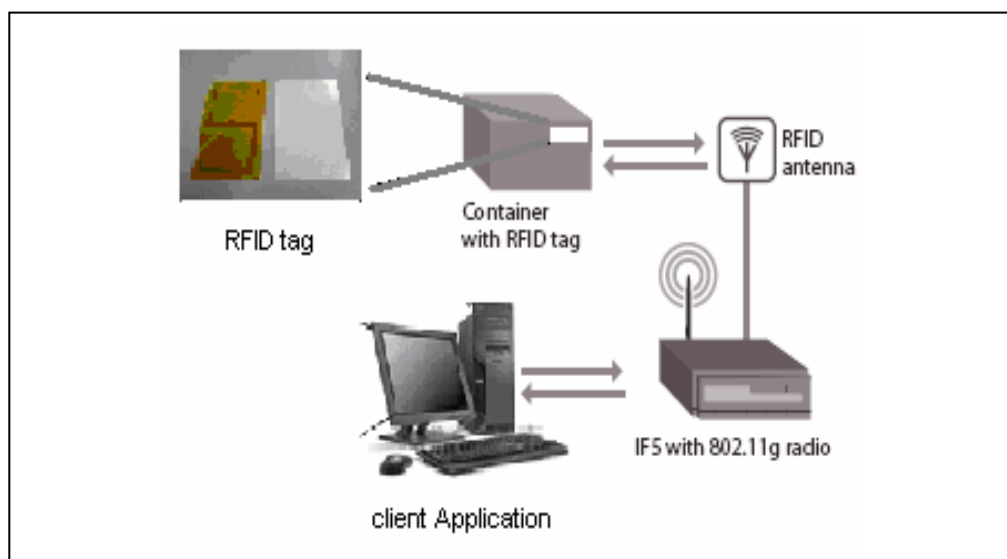


圖 5.1 RFID 系統架構圖

射頻辨識系統主要的硬體設備包括讀取器、天線、標籤等。在讀取器部分，本研究採用美商 Intermec 公司所發行的 IF5 射頻辨識讀取器，它是一套以 Linux 2.6 為核心的嵌入式系統主機 [18]。而就讀取器的運作可分為訊號輸入與訊號輸出兩部分。在訊號輸入的部分，透過銅軸電纜與天線相連接，當射頻辨識標籤進入天線可接收的範圍，會將資料傳回給天線，再透過銅軸電纜傳回給讀取器；訊號輸出的部份，使用乙太網路輸出，使得應用程式可以透過網際網路上各式各樣的通訊協定取得標籤內的資訊。IF5 讀取器如圖 5.2。而讀取器的規格及支援性如表 5.1。

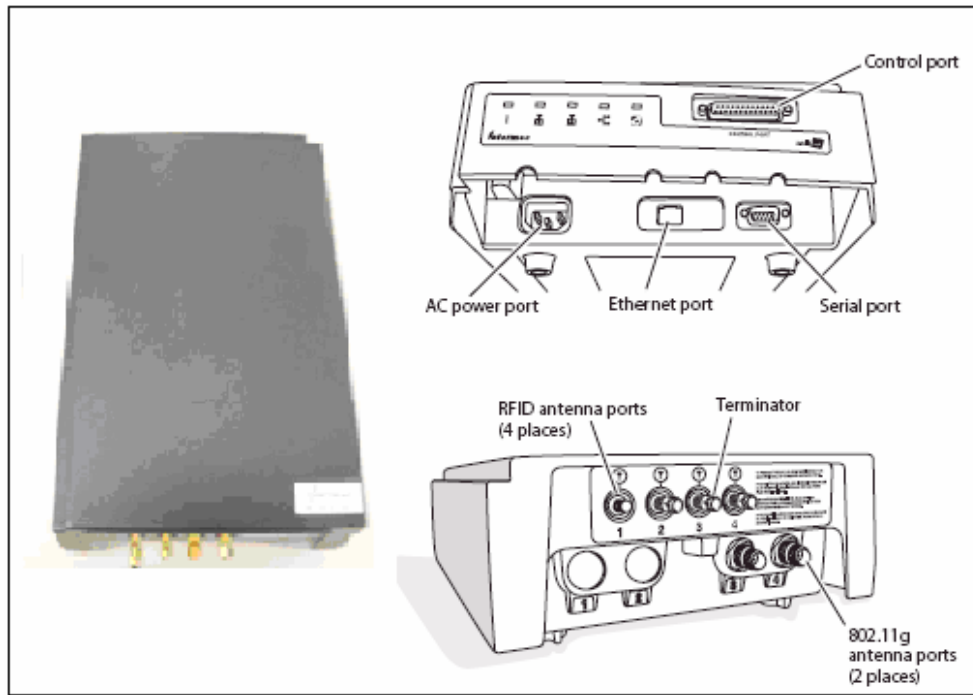


圖 5.2 IF5 射頻辨識讀取器[18]

表 5.1 IF5 讀取器的規格及支援性[18]

類別	規格
Linux base	Linux 2.6.11
Configuration	Done through a user interface either through the serial port or through a Web browser interface.
Antennas	The reader has 4 antennas that can be controlled.
GPIO	The reader has 4 inputs port and 4 output ports.
JVM	A J2ME v1.0-compliant CDC/Foundation JVM.
JavaScript	JavaScript Support.
JDBC	JDBC Support
FTP	FTP Server for downloading files to the IF5.
User Space	2MB, plus 64MB of Compact Flash storage capability.

射頻辨識標籤是由內含數Byte~數千Byte 大小的記憶體及簡單邏輯電路的IC 晶片以及小型天線所組成。天線的部份是感應射頻電波，以磁生電的方式啟動晶片，使得標籤具有讀寫功能；而晶片的部份主要是記載資料的記憶體區塊，而記憶體區塊又可分為唯讀區塊的 UID 識別碼及可讀寫的資料區塊。UID 碼為出廠時的識別碼，用於辨識標籤的唯一性，UID 碼不可修改；可讀寫的資料區塊用於儲存商品唯一識別碼的地方，也是讀取器真正要取得資料的地方。標籤規格如圖 5.3 所示。

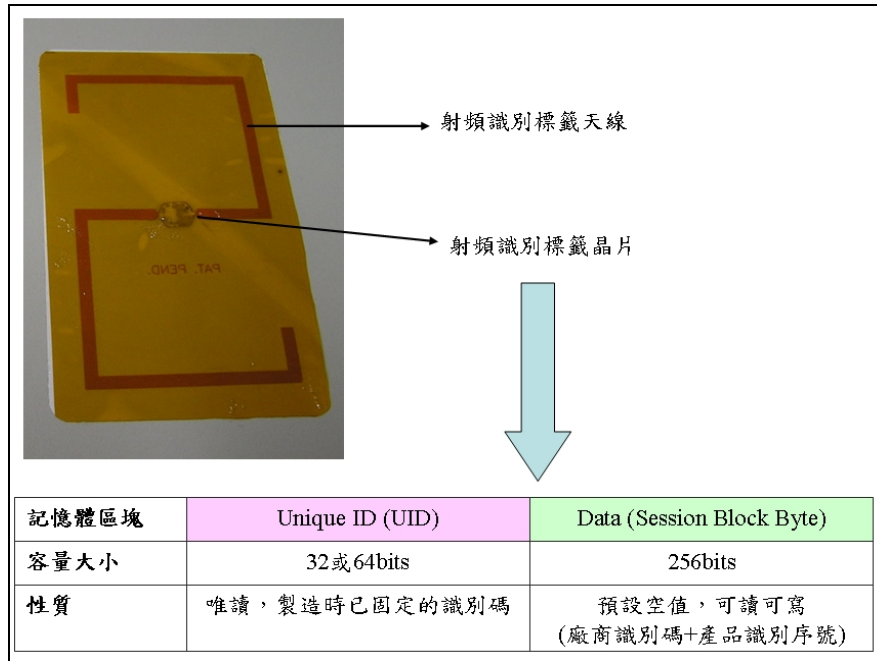


圖 5.3 射頻識別標籤規格[18]

在 3.4 節中已經提到本研究中 Index 的設計，接著透過 RFID 讀取器對標籤進行寫入的動作，圖 5.4 為標籤條碼的示意圖。以此片標籤為例，『HEF04B4946433936』是出廠時的 UID 碼，Data 中的第 2 碼到第 13 碼『096AAA001032』為供應商所提供的生產批號，接著四碼『KIFC』為高雄批發市場提供網路服務的服務名稱，第 18 到 33 碼『960326LCA1800301』為批發市場的拍賣序號，34 到 42 碼『096000231』為承銷商的進貨編號。



圖 5.4 射頻識別標籤內容

5.1.2 射頻辨識軟體架構

射頻辨識系統的軟體部份所扮演的角色，主要是讓使用者能夠透過一種方式對讀取器進行操作，包括系統的設定等，主要的功能是對射頻辨識標籤進行讀取、寫入的動作。

Intermec 公司針對這個 IF5 讀取器開發一套 BRI(Basic Reader Interface)指令集，讓使用者可以透過簡單的方式對內部及外部的應用程式進行操作，包括對射頻辨識標籤進行讀取、寫入等動作。BRI 的指令與說明約略如表 5.2。

BRI 的各指令也有其參數，以本程式為例，從第二個字串開始寫入 20 個字元，如 Write String(2,20)=""KIFC960326LCA1800301""，同理，讀取這段字串則為 Read String(2,20)，此外還有很多參數，包括 Hex()、Int()等，詳情請參考說明手冊。

表 5.2 BRI 的指令與說明[18]

BRI 指令	說明
READ	讀取標籤代碼
WRITE	寫入標籤代碼
REPAET	重覆執行上一個指令
SET	建立一個簡短代號以代替一串長敘述
ATTRIBUTE	對讀取平臺設立參數規格
VERSION	傳回目前的軟體版本
FACDFLT	將設定回復到原廠設定

IF5 讀取器支援透過各種的通訊協定與其溝通，例如 telnet、http、ftp 等通訊協定。透過通訊協定，一般應用程式可以簡單的對 IF5 建立一個 Socket 連線，並傳送 BRI 指令進行操作。Socket 為兩個程式之間的通信方法，透過與目的端位址建立連線後，將寫入 Socket 的資料傳送到目的端，而目的端也將執行結果透過 Socket 傳送回來，完成一次 Socket 的傳輸。

以追溯系統來說，射頻辨識讀寫器可視回 Socket 目的端，而追溯系統的客戶端即為 Socket 的本地端。對目的端而言，ServerSocket 這個物件在預設埠號 2189 建立一個 ServerSocket 物件，並透過 accept()這個方法傾聽來自本地端的連線。對本地端而言，透過 2189 這個埠號，將 Socket 封包傳送到遠端，經過遠端 accept()方法的確認後將成功訊息回傳給本地端，完成 Socket 的連線。

接著本地端要對讀取器進行操作，例如讀取標籤上的代碼。首先本地端將 BRI 中的 READ 指令封裝到 Socket 裡傳送給目的端，讀取器收到 READ 指令後透過天線讀取到標籤上的代碼並將其封裝到 Socket 裡面再回傳給本地端，本地端收到代碼後透過 close()的方法結束 Socket 連線。追溯系統客戶端透過 Socket 與射頻辨識讀取器溝通的架構如圖 5.5 所示。

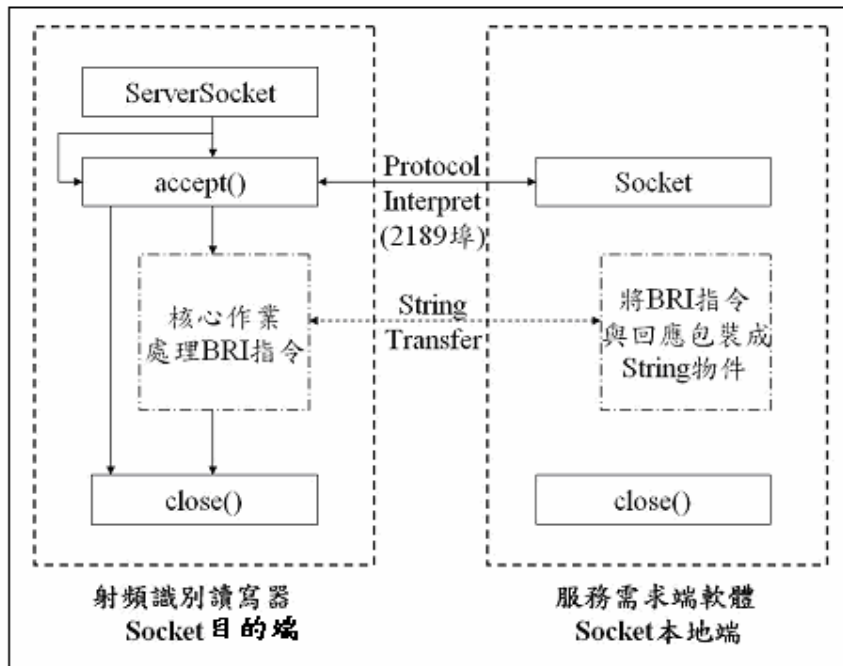


圖 5.5 追溯系統客戶端與射頻辨識讀取器溝通架構

而追溯系統的 Socket 程式碼如圖 5.6 所示，透過這個 Socket 程式，即可透過讀取器所讀到的標籤而取得其標籤內容，包括 Service Name 與條碼序號。

<pre>Socket socket = new Socket("140.113.59.175", 2189);</pre>	<p>建立連線</p>
<pre>DataInputStream datain = new DataInputStream(socket.getInputStream()); DataOutputStream dataout = new DataOutputStream(socket.getOutputStream());</pre>	<p>設定 IO</p>
<pre>String readStr = "read String(2,26)"; for (int i = 0; i < readStr.length(); i++) dataout.write(readStr.charAt(i)); dataout.write('\r'); dataout.write('\n'); dataout.flush(); try { Thread.sleep(500); } catch (Exception ex) {} byte[] inBytee = new byte[1024]; datain.read(inBytee); String Str1 = new String(inBytee, 0, inBytee.length);</pre>	<p>將BRI指令傳出並回傳結果</p>

圖 5.6 追溯系統客戶端 Socket 程式碼

5.2 追溯系統客戶端與仲介端溝通設計

在 2.3.6 節以及 4.4 節中已經分別介紹過 UDDI 的架構及使用者如何透過 Web-based 的介面使用 UDDI 註冊中心的註冊、搜尋等部份，在本節要介紹追溯系統客戶端如何與 UDDI 註冊中心溝通。本研究採用的是 UDDI4J 這項 JAVA 函式庫所提供的 API，UDDI4J 是 Java 方案中一種用來和註冊中心溝通的套件。UDDI4J 和透過 Web-based 的介面操作一樣都可以對 UDDI 註冊中心進行註冊、搜索等動作，所不同的是 UDDI4J 將每個操作封裝成 API，程式開發人員透過此 API 就能對 UDDI 註冊中心進行操作。針對某些特定目的開發的系統，例如在自行開發的應用程式上，需要 UDDI 註冊中心的資料能夠自動與其他程式互動的時候，Web-based 的介面並不能充分滿足這些需求。因此，透過 Non-Web API 的運用，可以達成將 UDDI 註冊、發佈與搜尋等功能整合到自行開發的應用程式。

圖 5.7 是一個客戶端與仲介端溝通的概念圖，而 Inquiry 及 Response 為客戶端與仲介端溝通的部份，也就是 UDDI4J 所扮演的角色。在 Inquiry 部份，UDDI4J 透過 Service Name 建構一個請求訊息(Inquiry Message)，並將其送到服務仲介端，也就是 UDDI 註冊中心；在 Response 部份，UDDI 註冊中心搜索符合該 Service Name 的資料後，將其 Access Point 封裝成一個回應訊息(Response Message)，回傳給需求端，完成需求端與仲介端的一次溝通。

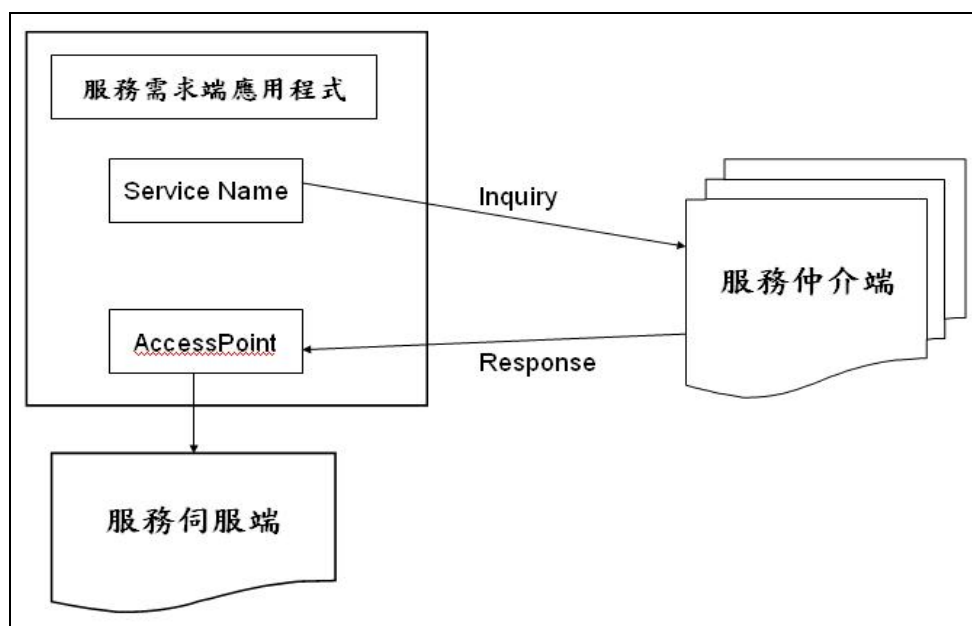


圖 5.7 服務需求端與服務仲介端互動關係

UDDI4J 主要可以分為表 5.3 等各種類別。org.uddi4j.client.UDDIProxy 建立與註冊中心的連線，並透過 org.uddi4j.request 及 org.uddi4j.response 類別下的函式對註冊中心進行操作。而 UDDI4J 主要以 org.uddi4j.response.DispositionReport 來表明操作的成功或失敗，以及 org.uddi4j.UDDIException 來表明錯誤情況。

表 5.3 UDDI4J 主要 API 結構與功能對照表

類別	功能
org.uddi4j.client.UDDIProxy	服務仲介端的主機 Proxy
org.uddi4j.datatype	發送與接收資訊的資料類別
org.uddi4j.request	發送請求訊息到服務仲介端
org.uddi4j.response	從服務仲介端傳回回應訊息
org.uddi4j.transport	傳送附屬資訊

圖 5.8 為本研究透過 UDDI4J 對註冊中心進行操作的程式碼說明。一開始透過 proxy 類別下的 setInquiryURL 方法設定所要溝通的 UDDI 註冊中心。由於與 UDDI 註冊中心的溝通主要是搜尋符合該 ServiceName 的 AccessPoint，因此在確認連線完成後，將 ServiceName 傳送到 UDDI 註冊中心，並開始搜索符合的資料。搜索完成後，將 AccessPoint 回傳給追溯系統的客戶端。至此完成追溯系統客戶端與仲介端的溝通，接著透過 AccessPoint 可以開始進行與追溯系統伺服器端的動作。完整程式碼於附錄四。

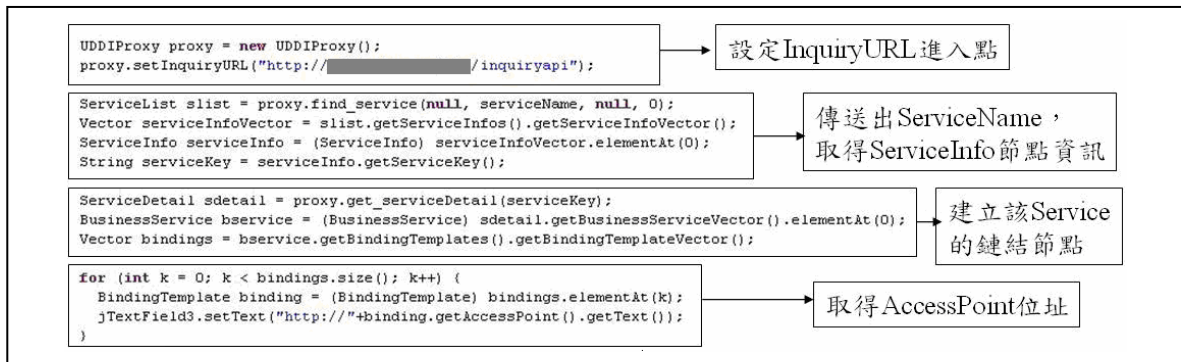


圖 5.8 使用 UDDI4J 套件取得服務需求端所需資訊的程式說明

5.3 追溯系統客戶端與伺服器端溝通設計

目前追溯系統客戶端的部分，在 5.1 節及 5.2 節，已經分別介紹過透過 Socket 取得射頻辨識標籤上的條碼以及透過 Service Name 向 UDDI 註冊中心取得伺服器的 Access Point，接下來本節要說明客戶端如何透過 Access Point 及條碼序號與伺服器端溝通。

網路服務客戶端與伺服器端溝通的設計，並沒有一定的標準，只要是符合 SOAP 格式的文件、可以解析出 WSDL 的內容，即可完成網路服務客戶端建置。

在本研究中，客戶端與伺服器端的溝通主要使用 Apache 提供的一組套件 org.apache.axis.client，透過 org.apache.axis.client 類別中的方法來啟動服務及接收服務伺服器端回應結果並處理。

org.apache.axis.client 中的 org.apache.axis.client.Service 類別，是 SOAP Web Services 的起點。透過 org.apache.axis.client.Call 類別所產生一個名為 call 的物件，為建立遠端呼叫的一種方法。Org.apache.axis.client.Call 類別的主要目的是呼叫遠端的網路服務。利用其中的 setTargetEndpointAddress(String url) 方法，定義網路服務伺服器端的位置；setOperationName (String opName) 方法，定義伺服器端的函數名稱；setReturnType (QName type) 方法，決定傳回的資料型態；call.invoke (Object[] params) 當中的 params 代表要傳入的參數，在此為射頻辨識標籤的條碼序號，並透過這個參數代入伺服器端，經過處理將回傳的資料型態設為 String，而回傳的變數為 ret。此外函數裡頭的變數，Java 將它變成物件的陣列，也就是將 Web services 的資料型態當成物件來看待。圖 5.9 為客戶端與伺服器端溝通的 org.apache.axis.client 部分程式碼，這段程式碼為客戶端向批發市場進行取得批發市場日誌的服務，圖 5.10 為客戶端的 getAucData 方法名對應到的批發市場日誌服務 WSDL 的 getAucData 作業訊息。

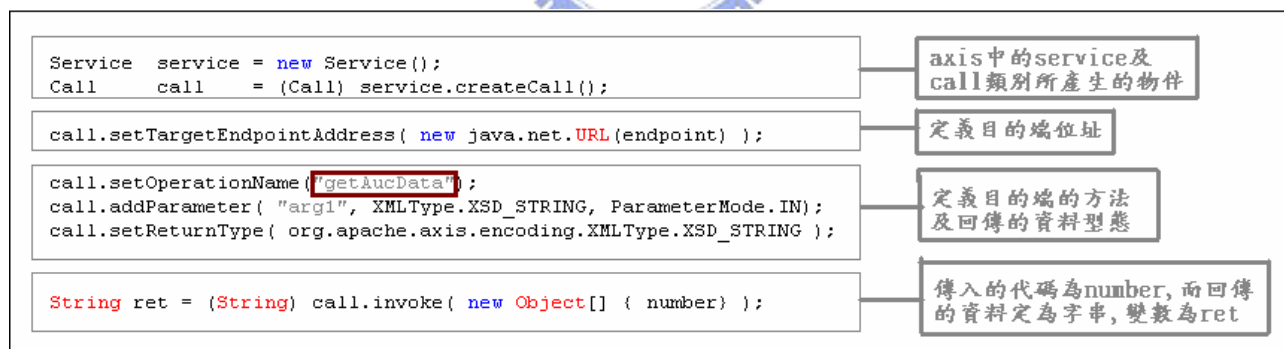


圖 5.9 客戶端與伺服器端溝通的程式碼


```

+ <wsdl:message name="getSupplierDataRequest">
- <wsdl:message name="getAucDataRequest">
  <wsdl:part name="in0" type="xsd:string" />
</wsdl:message>
+ <wsdl:message name="getFlowerDataRequest">
+ <wsdl:message name="getBuyerDataResponse">
+ <wsdl:message name="getBuyerDataRequest">
+ <wsdl:message name="getSupplierDataResponse">
+ <wsdl:message name="getFlowerDataResponse">
- <wsdl:message name="getAucDataResponse">
  <wsdl:part name="getAucDataReturn" type="xsd:string" />
</wsdl:message>
- <wsdl:portType name="KSflower">
  - <wsdl:operation name="getAucData" parameterOrder="in0">
    <wsdl:input message="impl:getAucDataRequest" name="getAucDataRequest" />
    <wsdl:output message="impl:getAucDataResponse" name="getAucDataResponse" />
  </wsdl:operation>
  + <wsdl:operation name="getSupplierData" parameterOrder="in0">
  + <wsdl:operation name="getBuyerData" parameterOrder="in0">
  + <wsdl:operation name="getFlowerData" parameterOrder="in0">
  </wsdl:portType>
- <wsdl:binding name="KSflowerSoapBinding" type="impl:KSflower">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
  - <wsdl:operation name="getAucData">
    <wsdlsoap:operation soapAction="" />
    + <wsdl:input name="getAucDataRequest">
    <wsdl:output name="getAucDataResponse">
  </wsdl:operation>
  + <wsdl:operation name="getSupplierData">
  + <wsdl:operation name="getBuyerData">
  + <wsdl:operation name="getFlowerData">
  </wsdl:binding>
- <wsdl:service name="KSflowerService">
  - <wsdl:port binding="impl:KSflowerSoapBinding" name="KSflower">
    <wsdlsoap:address location="http://localhost:8080/axis/services/KSflower" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

getAucData的message

getAucData的message

getAucData的
operation name

getAucData
的binding

圖 5.10 批發市場 WSDL 的 getAucData 服務

本文中 使用 Call 類別的方式有別於一般的預存程序(Stored Procedure)，所謂的預存程序是指在資料庫中將一串要執行的敘述句封裝成一個程序；當執行該程序時，如同執行該敘述句。而在本研究之中，並沒有使用預存程序，純粹以 Call 類別取而代之，如圖 5.9。Call 類別的特色是將欲處理的字串封裝起來傳輸，雖然如果資料庫內容需要修改時，程式需要跟著修改，對於彈性而言不如預存程序的優良。但是 Call 類別的簡易性及封裝性，使得網路服務的建置相對的簡易，並以封裝的型式傳輸，被竊取的時候安全性大增。

回傳的資料透過字串的處理方式從頁面以表格的方式呈現出來。使用者可以透過表格中資訊找尋自己所需要的，頁面的操作在 5.4 節中介紹。相關於客戶端界面的設計，包括字串的處理方式、表格的呈現方式等在本節不再撰述，請參考附錄四。

5.4 追溯系統客戶端整合入口的操作

在本節主要描述追溯系統客戶端界面的操作，包括操作步驟、資訊流程以及介面說明，另外透過查詢結果的欄位說明和一個例子讓整個說明更加清楚、完整。

圖 5.11 為追溯系統客戶端的程式介面，設計精簡且容易操作。上方的四個 TextField，分別秀出供應鏈上各端點的代碼以及所取得的伺服器位址，並透過旁邊兩個 button 操作取得資料。下方有直接的操作說明，而取得的資料結果透過上方的頁籤切換頁面以表格的方式顯示。

花卉追溯系統

批號資料 供應商資料 高雄花市資料 承銷商資料 花卉資料 供應商日誌 施肥日誌 承銷商日誌

供應商生產批號 096AAAD01032

批發市場拍賣序號 K1FC960326LCA1800301

承銷商進貨編號 096000231 取得AccessPoint

服務位址 8:8080/axis/services/KSflower 取得資料

操作說明:

1. 經由 RFID 讀入批號
2. 按下取得 Access Point, 得到Web Service位址
3. 按下取得資料, 並依需求切換至不同的畫面

圖 5.11 追溯系統客戶端程式介面

圖 5.12 是追溯系統客戶端的操作步驟圖，透過這個圖形，可以清楚的了解到整個操作分成三個步驟。首先透過 Socket 與射頻辨識讀取器溝通，自動取得標籤中的資料，顯示於代碼欄位之中；接著，透過按下『取得 Access Point』按鈕，向 UDDI 註冊中心查詢符合 Service Name 的網路服務伺服器端資料，其位址顯示於 Services 欄位之中，如箭號所指；最後，按下『取得資料』按鈕，透過 Access Point 將條碼序號送往追溯系統網路服務伺服器端，再依網路服務所提供的內容向該資料庫查詢，依類別顯示結果於供應商資料、批發市場資料、承銷商資料、花卉資料等四個頁面的表格之中。接著以所取得的供應人代碼與承銷人代碼為 UDDI 註冊的服務名稱，向 UDDI 仲介端查詢該供應商與承銷商這筆資料的日誌及供應商施肥紀錄，將結果分別顯示於供應商日誌與承銷商日誌及供應商施肥紀錄等三個頁面的欄位中。

以代碼『960326LCA1800301』為例，所得到的結果顯示於各頁面之中，查詢結果的各個頁面圖如圖 5.13 所示。5.11(a)為供應商資料；(b) 為批發市場資料；(c)為承銷商資料；(d)為花卉資料；(e)為供應商日誌；(f)為供應商施肥日誌；(g)為承銷商日誌。表 5.4 為詳細的查詢結果資料表。透過查詢的結果，使用者可以針對該花卉商品得到所需的資訊。例如供應商部分，由南投草屯的李玉燕提供，採取冷藏運輸，保鮮預冷處理；批發市場部份，在電子化交易、提貨管制監控的高雄批發市場進行拍賣，花卉等級 A3、件數 1 件、把數 6 把、成交價 24 元；承銷商為高雄縣的蘇榮雄，採取分級包裝、保鮮處理的方式；花卉則是國產的玫瑰比浪卡，代碼為 FR602。

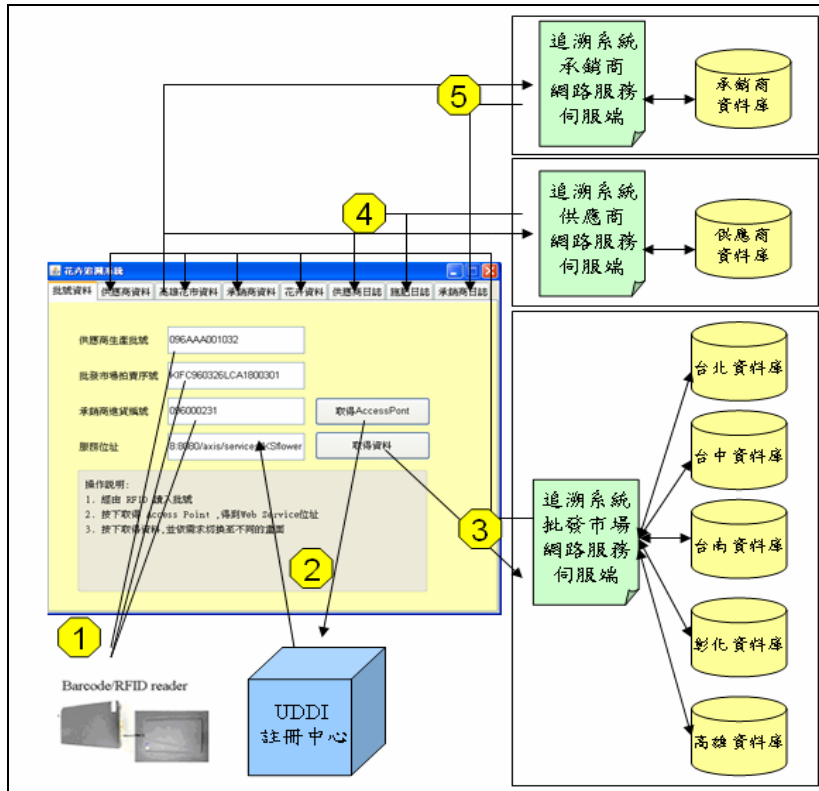


圖 5.12 追溯系統客戶端操作步驟圖

<p>系統</p> <p>供應商資料 高雄批發市場資料 承銷商資料 花卉資料 供應商日誌 承銷商日誌</p> <table border="1"> <thead> <tr> <th>供應人代碼</th> <th>供應人姓名</th> <th>最後供貨日期</th> <th>供應地區</th> </tr> </thead> <tbody> <tr> <td>LCA18</td> <td>李玉燕</td> <td>2007/04/02</td> <td>南投草屯</td> </tr> </tbody> </table>	供應人代碼	供應人姓名	最後供貨日期	供應地區	LCA18	李玉燕	2007/04/02	南投草屯	<p>系統</p> <p>供應商資料 批發市場資料 承銷商資料 花卉資料</p> <table border="1"> <thead> <tr> <th>日期</th> <th>花卉等級</th> <th>件數</th> <th>把數</th> <th>成交價</th> <th>備註</th> </tr> </thead> <tbody> <tr> <td>2007/03/26</td> <td>A3</td> <td>1</td> <td>6</td> <td>24</td> <td>電子化拍賣,提貨管制監控</td> </tr> </tbody> </table>	日期	花卉等級	件數	把數	成交價	備註	2007/03/26	A3	1	6	24	電子化拍賣,提貨管制監控																					
供應人代碼	供應人姓名	最後供貨日期	供應地區																																							
LCA18	李玉燕	2007/04/02	南投草屯																																							
日期	花卉等級	件數	把數	成交價	備註																																					
2007/03/26	A3	1	6	24	電子化拍賣,提貨管制監控																																					
<p>(a)查詢結果供應商資料頁面</p>	<p>(b)查詢結果批發市場資料頁面</p>																																									
<p>系統</p> <p>供應商資料 高雄批發市場資料 承銷商資料 花卉資料 供應商日誌 承銷商日誌</p> <table border="1"> <thead> <tr> <th>承銷人代碼</th> <th>承銷人姓名</th> <th>承銷人所在區域</th> <th>最後交易日期</th> </tr> </thead> <tbody> <tr> <td>30919</td> <td>蘇榮雄</td> <td>高雄縣</td> <td>2007/04/01</td> </tr> </tbody> </table>	承銷人代碼	承銷人姓名	承銷人所在區域	最後交易日期	30919	蘇榮雄	高雄縣	2007/04/01	<p>系統</p> <p>供應商資料 批發市場資料 承銷商資料 花卉資料</p> <table border="1"> <thead> <tr> <th>花卉代碼</th> <th>花卉名</th> <th>識別碼_F_國產_I_進口</th> </tr> </thead> <tbody> <tr> <td>FR602</td> <td>玫瑰比浪卡</td> <td>F</td> </tr> </tbody> </table>	花卉代碼	花卉名	識別碼_F_國產_I_進口	FR602	玫瑰比浪卡	F																											
承銷人代碼	承銷人姓名	承銷人所在區域	最後交易日期																																							
30919	蘇榮雄	高雄縣	2007/04/01																																							
花卉代碼	花卉名	識別碼_F_國產_I_進口																																								
FR602	玫瑰比浪卡	F																																								
<p>(c)查詢結果承銷商資料頁面</p>	<p>(d)查詢結果花卉資料頁面</p>																																									
<p>系統</p> <p>供應商資料 高雄批發市場資料 承銷商資料 花卉資料 供應商日誌 承銷商日誌</p> <table border="1"> <thead> <tr> <th>流水號</th> <th>肥料種類</th> <th>運輸方式</th> <th>保存處理</th> <th>出貨時間</th> <th>出貨市場</th> <th>拍賣序號</th> </tr> </thead> <tbody> <tr> <td>96032602...</td> <td>有機肥料...</td> <td>冷藏運輸</td> <td>保鮮預冷...</td> <td>2007/03/26</td> <td>高雄批發...</td> <td>960326LC...</td> </tr> </tbody> </table>	流水號	肥料種類	運輸方式	保存處理	出貨時間	出貨市場	拍賣序號	96032602...	有機肥料...	冷藏運輸	保鮮預冷...	2007/03/26	高雄批發...	960326LC...	<p>系統</p> <p>供應商資料 高雄花市資料 承銷商資料 花卉資料 供應商日誌 施肥日誌</p> <table border="1"> <thead> <tr> <th>生產批號</th> <th>施肥時間</th> <th>肥料編號</th> </tr> </thead> <tbody> <tr><td>096AAA001032</td><td>2006/07/24</td><td>肥料1</td></tr> <tr><td>096AAA001032</td><td>2006/07/28</td><td>肥料2</td></tr> <tr><td>096AAA001032</td><td>2006/08/05</td><td>肥料3</td></tr> <tr><td>096AAA001032</td><td>2006/08/15</td><td>肥料4</td></tr> <tr><td>096AAA001032</td><td>2006/08/30</td><td>肥料5</td></tr> <tr><td>096AAA001032</td><td>2006/09/12</td><td>肥料6</td></tr> <tr><td>096AAA001032</td><td>2007/02/10</td><td>肥料7</td></tr> <tr><td>096AAA001032</td><td>2007/03/15</td><td>肥料8</td></tr> </tbody> </table>	生產批號	施肥時間	肥料編號	096AAA001032	2006/07/24	肥料1	096AAA001032	2006/07/28	肥料2	096AAA001032	2006/08/05	肥料3	096AAA001032	2006/08/15	肥料4	096AAA001032	2006/08/30	肥料5	096AAA001032	2006/09/12	肥料6	096AAA001032	2007/02/10	肥料7	096AAA001032	2007/03/15	肥料8
流水號	肥料種類	運輸方式	保存處理	出貨時間	出貨市場	拍賣序號																																				
96032602...	有機肥料...	冷藏運輸	保鮮預冷...	2007/03/26	高雄批發...	960326LC...																																				
生產批號	施肥時間	肥料編號																																								
096AAA001032	2006/07/24	肥料1																																								
096AAA001032	2006/07/28	肥料2																																								
096AAA001032	2006/08/05	肥料3																																								
096AAA001032	2006/08/15	肥料4																																								
096AAA001032	2006/08/30	肥料5																																								
096AAA001032	2006/09/12	肥料6																																								
096AAA001032	2007/02/10	肥料7																																								
096AAA001032	2007/03/15	肥料8																																								
<p>(e)查詢結果供應商日誌資料頁面</p>	<p>(f)查詢結果供應商施肥日誌資料頁面</p>																																									
<p>系統</p> <p>供應商資料 高雄花市資料 承銷商資料 花卉資料 供應商日誌 施肥日誌</p> <table border="1"> <thead> <tr> <th>進貨批號</th> <th>進貨市場</th> <th>進貨時間</th> <th>包裝方式</th> <th>保存處理</th> <th>出貨對象</th> <th>出貨方式</th> </tr> </thead> <tbody> <tr> <td>096000231</td> <td>高雄花市</td> <td>2007/03/26</td> <td>分級包裝</td> <td>保鮮處理</td> <td>自用</td> <td>無</td> </tr> </tbody> </table>	進貨批號	進貨市場	進貨時間	包裝方式	保存處理	出貨對象	出貨方式	096000231	高雄花市	2007/03/26	分級包裝	保鮮處理	自用	無																												
進貨批號	進貨市場	進貨時間	包裝方式	保存處理	出貨對象	出貨方式																																				
096000231	高雄花市	2007/03/26	分級包裝	保鮮處理	自用	無																																				
<p>(g)查詢結果承銷商日誌資料頁面</p>																																										

圖 5.13 追溯系統客戶端查詢結果頁面

表 5.4 追溯系統客戶端查詢結果表

頁面	欄位	資料
供應商資料 圖 5.13(a)	供應人代碼	LCA18
	供應人姓名	李玉燕
	最後供貨日期	2007/04/02
	供應地區	南投草屯
批發市場資料 圖 5.13(b)	日期	2007/03/26
	花卉等級	A3
	件數	1
	把數	6
	成交價	24
	備註	電子化拍賣,提貨管制系統
承銷商資料 圖 5.13(c)	承銷人代碼	30919
	承銷人姓名	蘇榮雄
	承銷人所在區域	高雄縣
	最後交易日期	2007/04/01
花卉資料 圖 5.13(d)	花卉代碼	FR602
	花卉名	玫瑰比浪卡
	識別碼_F_國產_I_進口	F
供應商日誌 圖 5.13(e)	流水號	96032602035
	肥料種類	有機肥料、化學肥料
	運輸方式	冷藏運輸
	保存處理	保鮮預冷處理
	出貨時間	2007/03/26
	出貨市場	高雄批發市場
	拍賣序號	960326LCA1800301
施肥日誌 圖 5.13(f)	生產批號	096AAA001032
	施肥時間	2007/03/15
	肥料編號	肥料 8
承銷商日誌 圖 5.13(g)	進貨批號	096000231
	進貨市場	高雄花市
	進貨時間	2007/03/26
	包裝方式	分級包裝
	保存處理	包鮮處理
	出貨對象	自用
	出貨方式	無

第六章 結論與未來方向

本章針對本論文所提出的追溯系統做一個總結。第 6.1 節為本研究的結論與研究貢獻，第 6.2 節探討可以加以改進與未來發展的方向。

6.1 結論與研究貢獻

本研究提出網路服務與射頻辨識導向的追溯系統，最後針對此追溯系統幾個部份做討論，包括追溯系統的架構、所解決的問題和可行性及安全性等。

隨著科技的進步，對於產品的品管及安全性的要求也越來越重視，為了因應這個環境，因而有了追溯系統等技術的誕生。本研究是針對花卉產業，利用目前火紅的射頻辨識技術和網路服務技術搭配資料庫建立花卉追溯系統。第 1.2 節中有提到花卉供應鏈中的問題，包括搭載的載具為傳統的一維條碼，因而在作業中需要許多的人力、作業包含許多的不確定性以及一維條碼需受限於在各方面外在環境上的限制。本研究透過導入射頻辨識技術取代一維條碼，利用射頻辨識技術高度的環境容忍力、較遠的讀取距離、可同時讀取多個標籤及儲存容量大、可在一個標籤中儲存多個序號等優點改善過去的缺失。此外由於荷蘭的成功導入與優良的績效，可知射頻辨識技術的導入在花卉產業中是可行的，對於台灣的花卉批發市場也有莫大的助益。而且射頻辨識技術受限的條件，如水、金屬等問題並不存在於花卉產業之中，此外相較於傳統條碼，射頻辨識標籤較貴的成本對於高單價的花卉產業也不具有那麼大的威脅性。

另外供應鏈沒有透明化，無法追蹤及追溯到各層級中的狀況，以及過去使用者沒有一個介面可以得知花卉供應鏈中的資訊，因而顧客跟業者無法有良好的互動等問題，透過追溯系統強大的追溯與追蹤功能都可獲得改善。不僅供應鏈中重要的資訊有個管道可以獲得，此外也強化了風險的管理，業者更能透過追溯系統的介面發佈消息與消費者溝通。而網路服務強大的功能在前幾章已經有詳細的介紹，透過網路服務的架構能夠改善過去集中式架構更甚至是單純的主從式架構大量的成本。而且網路服務技術發展已經純熟，相較之下網路服務的服務性使得建構系統較為容易，網路服務強大的仲介功能也讓系統的架構趨於完善，此外網路服務強大的移植性對於完整的追溯系統架構是不可或缺的。

本研究具有另外一項重要的貢獻。網路服務與射頻辨識導向的追溯系統是一個資訊整合交換的系統架構，此架構具有可移植性，不管未來資訊如何演進，業者都可以參考這個架構及建置過程建立適合該產業的追溯系統。此外本系統對於操作的方便也有些許考量，如跨平台能力與操作的容易性。考量到使用者的作業平台不一致，因此使用具有跨平台能力的 Java 語言及分散式架構建立系統，操作的簡單性考量讓使用者能夠透過一、兩個按鈕即可得到所需的資訊，不需要經過繁雜的操作，讓使用者可以容易的上手。

安全性一直以來都是射頻辨識技術中令人注意的部分，舉例來說如果在動物身上植入 RFID 晶片，可以隨時得知牠所在位置及牠的資訊，不用擔心走失的問題。但相對的，一樣的問題如果發生在人身上呢？人的隱私沒有一個保障，因此安全性的問題造成多數學者提出疑問，因此針對安全性有許多解決之道因應而生，例如鎖密碼等方案。不僅如此，網路服務也曾被提出過安全性的問題，任何牽扯到資訊都存在有安全性的問題。在資訊安全的部份，本系統搭載的射頻辨識標籤中存有的只有花卉的代碼，所有的資訊都存在於資料庫中，光讀取

到條碼，沒有經過網路服務等程序一樣無法得到資料。而透過網路服務傳輸資料的方式在第 5.3 節中也提過，是採取 Call 類別傳輸字串，當字串到達客戶端時才開始切割欄位，因此在傳輸之中竊取任何資料都無法解讀。對花卉產業而言每一筆資料的重要性或許相對不高，但是對於金融產業，或是其他牽涉到重要資訊的產業，安全性的要求就是完全不可忽略了。



6.2 未來發展方向與建議

以本研究為基礎，未來的發展方向可以分為兩部份，第一個部份是實際導入問題的解決，第二個部份為客戶端介面的修改。

本研究主要針對花卉產業建立資訊整合交換的追溯系統架構，但是這個架構仍屬於一個雛型，實際系統導入花卉產業仍存在許多細節問題，例如對於射頻辨識技術而言，每個讀取器、每個標籤如何裝置？在拍賣作業之中標籤如何進行溝通？標籤如何回收？流程如何設置？對於網路服務技術而言，仲介端使用何者註冊中心？程序是否需要別的功能？這些問題都是實際系統在導入之後才會注意到的問題，當然問題絕對不只有這樣。而未來有一天花卉業者及政府正視到這方向的存在，基於考量之下是否有導入的必要性，才會考量到更多的問題，這些都是系統未來發展的方向。

客戶端介面修改的部分，目前這個系統的客戶端使用的是有別於網頁的程式介面，雖然是免安裝而且隨身碟即可帶著走的小容量，但是卻缺乏像網頁般的方便性，無法單純透過網路就可以直接進到追溯系統。未來可能改良程式介面為 Java 的 Applet 並內嵌於網頁之中，透過 Socket 與射頻辨識讀取器及仲介端、伺服器端溝通。此外，手機內建 RFID 讀取器已經是各手機大廠致力開發的重點之一，未來透過手機中的讀取器，使用者便可自行取捨所需的資訊。透過 J2ME，將追溯系統介面寫入手機之中，顧客即可透過自己的手機便可獲得花卉的資訊了。射頻辨識技術的急速發展，標籤的普及將使得讀取器隨著人們的需求有了多樣化，而連帶使得技術下的產物如追溯系統的介面也有了多重的變化性。



參考文獻

1. 日經 BP 社 RFID 技術編輯部，RFID 技術與應用，旗標出版股份有限公司，2004。
2. 吳信輝，「Web Services 技術介紹(九)」，資訊技術雜誌，第 2106 期，30-36 頁，2005。
3. 施威銘研究室，SQL Server2000 設計實務，旗標出版股份有限公司，2000。
4. 施威銘研究室，SQL Server2000 管理實務，旗標出版股份有限公司，2000。
5. 財團法人資訊工業策進會，經濟部工業局製造業自動化及電子化人才培訓計畫講義，2005。
6. 戚玉樑，網路服務技術導論，全華科技圖書股份有限公司，2004 年。
7. 梁高榮，「XML 與 RDF 技術介紹」，機械工業雜誌，七月，230-248 頁，2001。
8. 梁高榮，農產品交易工程學，國立交通大學出版社，1999。
9. 郭軒豪、賴俊翰、梁高榮、江東陽，「網路型三階正規化切花/盆花資料庫的建立」，機械工業，十二月，218-235 頁，2002。
10. 張堂穆、鍾國成，「從農業知識經濟探討台灣花卉運銷策略應用與發展」，機械工業雜誌，九月，271-284 頁，2001。
11. 張淙亮、梁高榮，「利用爪哇技術建立檔案轉移服務」，機械工業，五月，215-223 頁，2004。
12. 劉洋，魏飛，JBOSS 入門與應用-EJB 與 Web Services 開發手冊，博碩文化股份有限公司，2005。
13. 顧君暉，「RFID 與 EPC Network」，機械工業雜誌，十二月，47-55 頁，2005。
14. Garfinkel, S. L., "Adopting Fair Information Practices to Low Cost RFID Systems," Ubiquitous Computing Workshop, 2002.
15. http://en.wikipedia.org/wiki/Service-oriented_architecture, SOA Website.
16. <http://ws.apache.org/axis/>, Axis Website.
17. <http://www.epcglobalinc.org/>, EPCGlobal Website.
18. <http://www.intermec.tw/>, Intermec Technology.
19. <http://www.irfid.cn/html/76/n-1476.html>, Aalsmeer-RFID Website.
20. <http://www.rfidproductnews.com/issues/2007.03/success.php>, Flora Holland-RFID WebSite.
21. <http://www.rtc.itri.org.tw/RFID>, RFID Website.
22. <http://www.uddi.org/>, UDDI Website.
23. <http://www.w3.org/2002/ws/>, Web Service Website.
24. <http://www.w3.org/TR/soap/>, W3C SOAP Specification Website.
25. <http://www.w3.org/TR/wsdl>, WSDL Website.
26. <http://www.w3.org/TR/xslt>, XSLT Website.
27. Kevin, J.M., "Web Services : What's Real and What's Not?" IT Pro, April 2005.
28. Monson-Haefel, R., J2EE Web Services, Addison Wesley, 2004.
29. Morrison, C., "Traceability in Food Processing: An Introduction," Editor Lees, M., FoodAuthenticity and Traceability, pp. 459- 472, Woodhead Publishing Ltd, 2003.
30. Mousavi, A., Sarhadi, M., Lenk, A. and Fawcett, S., "Tracking and Traceability in The Meat Processing Industry: A Solution," British Food Journal, Vol. 104, Iss. 1, pp. 7- 19, 2002.
31. van Corp, K. J., "Tracking and Tracing: A Structure for Development and Contemporary Practices," Logistics Information Management, Vol 15, No. 14, pp.24-33, 2002.

附錄一 資料檔格式說明

附錄一為花卉批發市場上傳至伺服器端的原始資料格式。檔案的傳送包含了交易資料檔、供應人資料檔、承銷人資料檔及花卉品名資料檔，其格式說明如下：

附錄表 1 交易資料檔格式說明：(檔名範例 A.TXT)

資料檔	欄位	格式	例子	說明
交易資料檔	日期	PIC X(08)	20070102	
	序號	PIC 9(08)	AG017004	
	子序號	PIC 99	01	
	供應人	PIC X(05)	AG017	
	承銷人	PIC X(05)	30720	XXXXXX 則代表殘貨
	品名	PIC X(05)	FY008	
	等級	PIC X(02)	A+	
	件數	PIC 9(04)	1	
	把/件	PIC 9(04)	50	
	底價	PIC 9(03)	10	
	成交價	PIC 9(04)	12	
	套版類別	PIC X	1	
	拍賣/訂貨	PIC X	2	
	事故備註	PIC X		
拍賣線	PIC X	1		

附錄表 2 供應人資料檔格式說明：(檔名範例 SU.TXT)

資料檔	欄位	格式	例子	說明
供應人資料檔	供應代號	PIC X(06)	AB002	
	名稱	PIC X(24)	黃小容	
	最後供貨日期	PIC X(08)	20061229	
	語音代碼	PIC X(05)	15333	

附錄表 3 花卉品名資料檔格式說明：(檔名範例 PR.TXT)

資料檔	欄位	格式	例子	說明
花卉資料檔	品名代號	PIC X(05)	FA101	
	品名	PIC X(30)	康乃馨紅	
	品類	PIC X(05)	FA000	
	國產進口識別碼	PIC X(01)	F	F:國產 I:進口
	底價	PIC 9(03)	15	

附錄表 4 承銷人資料檔格式說明: (檔名範例 BU.TXT)

資料檔	欄位	格式	例子	說明
承銷人資料檔	承銷人代號	PIC X(05)	10231	
	名稱	PIC X(14)	張大華	
	地區別	PIC X	13	11=台北市 12=基隆市 13=台北縣 14=新竹縣市 15=桃園縣市 21=苗栗縣市 22=台中縣市 23=彰化縣市 24=南投縣市 25=雲林縣市 31=嘉義縣市 32=台南縣市 33=高雄市 34=高雄縣 35=屏東縣市 41=宜蘭縣市 42=花蓮縣市 43=台東縣市 51=澎湖縣市 52=金門縣市 53=連江縣市 54=南海諸島 99=其他
	最後交易日期	PIC X(08)	20061230	

註：

PIC X 表 1 文字 BYTE

PIC 9 表 1 數字 BYTE

PIC 99 表 2 數字 BYTE

附錄二 資料庫中資料表的欄位屬性說明

附錄二為網路型三階正規化資料庫資料表的欄位屬性說明，共包含十個資料表，此十個資料表其欄位屬性說明如下：

附錄表 5 拍賣資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		AUC_AUCNUM	CHAR		16	日期+序號+子序號
		AUC_DATE	DATETIME		8	日期
		AUC_NUM1	CHAR		6	序號
		AUC_NUM2	CHAR		2	子序號
	◎	AUC_SUPNUM	INT		4	自建供應人唯一識別碼
		AUC_OSUPNUM	CHAR		5	供應人原始代碼
	◎	AUC_BUYNUM	CHAR		5	承銷人原始代碼
	◎	AUC_FLONUM	CHAR		5	花卉品名
		AUC_FLOGRADE	CHAR		2	花卉等級
		AUC_QUANTITY	INT		4	件數
		AUC_STEM	INT		4	把數
		AUC_UPSETPRICE	INT		4	底價
		AUC_TRANPRICE	INT		4	成交價
		AUC_REGISTER	CHAR		1	3:38cm 4:48cm 5:58cm
	◎	AUC_LINE	CHAR		1	拍賣線
		AUC_NOTE	CHAR	◎	10	事件備註
		AUC_TYPE	CHAR		1	拍賣類別

附錄表 6 供應人資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	SUP_NUM	INT		4	自行建立供應人唯一識別碼
		SUP_ONUM	CHAR		5	供應人原始代碼
	◎	SUP_GROUPNUM	CHAR		5	供應團體代碼
		SUP_TYPE	CHAR		1	供應類別代碼
		SUP_LOCATIONNUM	CHAR		3	行政區代碼 供應人代碼前2位
		SUP_NAME	VARCHAR		50	供應人名稱
		SUP_LASTDATE	DATETIME	◎	8	最後供貨日期
		SUP_CODE	CHAR		5	供應人語音密碼
		SUP_IFUSE	CHAR		1	供應人代碼使用狀況(Y 或 N)

附錄表 7 供應行政區資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	SUPLOC_LOCATIONNUM	CHAR		2	供應行政區代碼
	◎	SUPLOC_CITYNUM	CHAR		1	供應縣市代碼
		SUPLOC_VILLAGE	VARCHAR		8	供應鄉鎮名稱

附錄表 8 供應縣市資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	SUPCITY_CITYNUM	CHAR		1	供應縣市代碼
		SUPCITY_CITY	VARCHAR		8	供應縣市名稱

附錄表 9 承銷人資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		BUY_BUYNUM	CHAR		5	承銷代碼
	◎	BUY_AREANUM	CHAR		2	地區別代碼
		BUY_NAME	VARCHAR		50	承銷地區名稱
		BUY_LASTDATE	DATETIME	◎	8	最後交易日期

附錄表 10 承銷區域資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	BUYAREA_AREANUM	CHAR		2	承銷區域代碼
		BUYAREA_AREA	VARCHAR		15	承銷區域名稱

附錄表 11 花卉資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		FLOWER_NUM	CHAR		5	花卉代碼
		FLOWER_NAME	VARCHAR		50	花卉品名
	◎	FLOWER_KIND	CHAR		5	花卉種類
		FLOWER_IDEN	CHAR		1	0:切 1:盆
		FLOWER_SOURCE	CHAR		1	花卉來源(F)國產(I)進口
		FLOWER_UPSETPRICE	INT	◎	4	底價

附錄表 12 花卉等級表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	AUC_FLOGRADE	CHAR		1	花卉等級代碼
		GRADE_NAME	VARCHAR	◎	8	等級名稱

附錄表 13 花卉等級略表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎		FLOWER_GRADE	CHAR		2	花卉等級代碼
		FLOWER_GRADE_NAME	VARCHAR		2	花卉等級全名
	◎	FLOWER_GRADE_DENAME	CHAR		10	花卉等級名稱

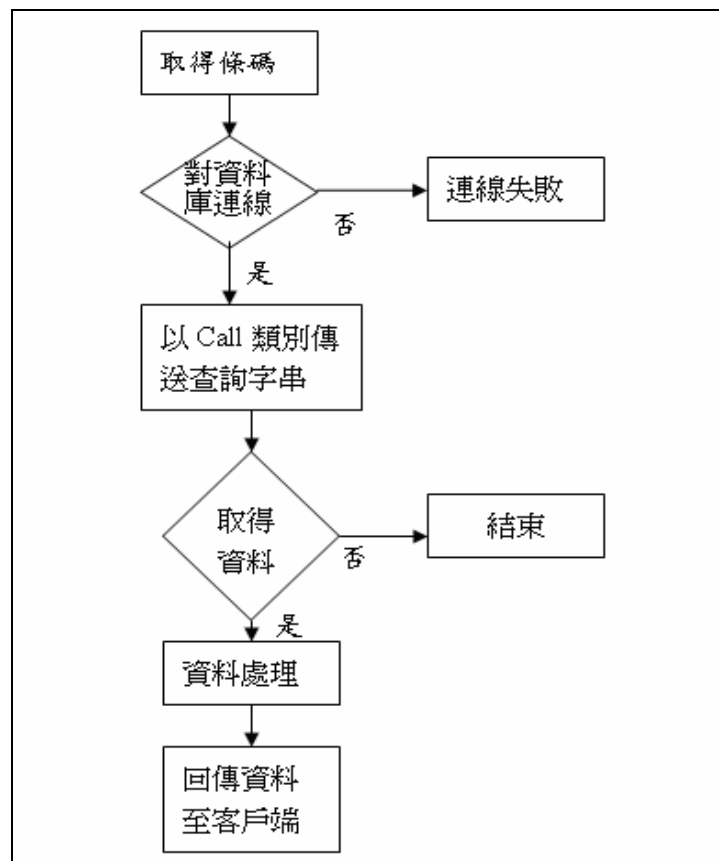
附錄表 14 拍賣線資料表

主鍵	外鍵	欄位名	資料型別	允許空值	長度	說明
◎	◎	SALE_LINENUM	CHAR		1	拍賣線代碼
		LINE_NAME	VARCHAR		4	拍賣線名稱



附錄三 網路服務伺服器端程式說明

網路服務伺服器端的程式大概可以分成三個部分，分別為網路服務內容程式、佈署程式、卸載程式。在第 1.4 節已經提過供應商、批發市場、承銷商各自提供其網路服務，而服務內容是針對各自的資料庫做存取，因此在網路服務內容的程式建構隨著類別與方法名的不同各自有不同的傳送 SQL 字串對各自的資料庫做查詢。以附錄表 15 而言，是批發市場服務對拍賣資料表進行存取的動作，其他的網路服務也雷同，而網路服務內容的程序如附錄圖 1。附錄表 16 為佈署服務程式(Deploy.wsdd)的程式碼，附錄表 17 為卸載服務程式(Undeploy.wsdd)的程式碼。



附錄圖 1 網路服務程式碼流程

附錄表 15 網路服務內容程式碼

```
// KSflower.java
import java.sql.*;
import java.util.*;
import javax.swing.*;
public class KSflower{
    private Connection connection;
    private Statement statement;
    public String getFlowerData(String clientdata)
    {
```

```

String sqldata1 = clientdata;
String sqlstatement = "SELECT * from 拍賣資料表 WHERE AUC_AUCNUM = " + sqldata1 + """;
try {
    Class.forName("net.sourceforge.jtds.jdbc.Driver");
    // 對資料庫建立連線
    Connection connection
=DriverManager.getConnection("jdbc:jtds:sqlserver://140.113.#####:1433/Kaoshiung", "#####", "#####");
    // 建立對資料庫查詢的敘述
    statement
=
connection.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_READ_ONLY);
    // 執行查詢動作
    ResultSet resultSet = statement.executeQuery(sqlstatement);
    // 得到結果
    StringBuffer results = new StringBuffer();
    ResultSetMetaData metaData = resultSet.getMetaData();
    int numberOfColumns = metaData.getColumnCount();
    //前兩碼紀錄 column 數
    if (numberOfColumns<10) results.append("0" + new Integer(numberOfColumns).toString());
    else results.append(new Integer(numberOfColumns).toString()) ;
    //前兩碼紀錄 row 數
    resultSet.last();
    int numberOfRows = resultSet.getRow();
    resultSet.beforeFirst();

    if (numberOfRows<10) results.append("0" + numberOfRows) ;
    else results.append(new Integer(numberOfRows).toString()) ;
    for ( int i = 1; i <= numberOfColumns; i++)
        results.append( metaData.getColumnName( i ) + "\t" );
    results.append( "\\\" );
    while ( resultSet.next() ) {
        for ( int i = 1; i <= numberOfColumns; i++)
            results.append( resultSet.getObject( i ) + "\t" );
        results.append( "\n" );
    }
}
// 回傳資料
return results.toString();
}
catch ( SQLException sqlException ) {
    return "Database Error";
}
}

```

```

catch ( ClassNotFoundException classNotFound ) {
    return "Driver Not Found";
}
finally {
    try {
        statement.close();
    }
    catch ( SQLException sqlException ) {
        return "Database Error";
    }
}
}
}
}


```

附錄表 16 佈署服務程式的程式碼(Deploy.wsdd)

```

// deploy.wsdd
<deployment xmlns="http://xml.apache.org/axis/wsdd/"
    xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">
    <service name="KSflower" provider="java:RPC">
        <parameter name="className" value="KSflower"/>
        <parameter name="allowedMethods" value="*" />
    </service>
</deployment>

```



附錄表 17 佈署服務程式的程式碼(Undeploy.wsdd)

```

// undeploy.wsdd
<undeployment
    xmlns="http://xml.apache.org/axis/wsdd/"
    <!-- Services from TanhouService WSDL service -->
    <service name="getFlowerData"/>
</undeployment>

```

附錄四 網路服務客戶端程式說明

客戶端的程式可以分為三大類，分別為客戶端透過 Socket 跟 RFID 讀取器溝通的 Socket Thread 程式碼，如附錄表 18。客戶端與仲介端溝通的 UDDI4J 程式碼，如附錄表 19，其程式流程如附錄圖 2。客戶端與伺服器端溝通的程式碼，如附錄表 20，其程式流程圖如附錄圖 3。

附錄表 18 客戶端透過 Socket 與 RFID 讀取器溝通的程式碼

```
// Socket thread
Socket s = new Socket("140.113.59.174", 2189);
DataInputStream in = new DataInputStream(s.getInputStream());
DataOutputStream out = new DataOutputStream(s.getOutputStream());
//讀取標籤字串第二位開始 20 碼
String cmdstring= "read String(2,20)";
for (int i = 0; i < cmdstring .length(); i++) out.write(cmdstring.charAt(i));
out.write('\r');
out.write('\n');
out.flush();
try {
    Thread.sleep(500);
} catch(Exception ex){};
byte[] inBytee = new byte[1024];
in.read(inBytee);
```



附錄表 19 客戶端與仲介端溝通的 UDDI4J 程式碼

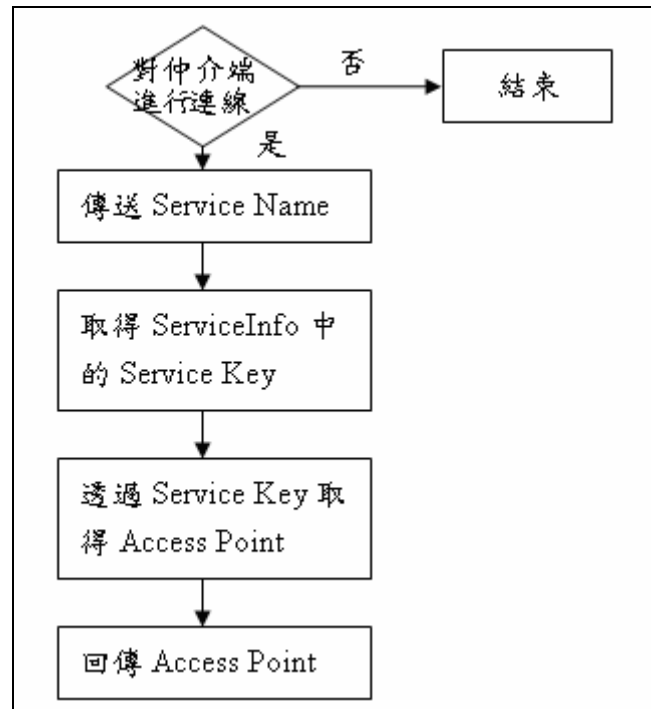
```
// UDDI4J
try {
// 連線
    String serviceName = jTextField2.getText();
    UDDIProxy proxy = new UDDIProxy();
    proxy.setInquiryURL("http://#####/inquiryapi");
//取得 ServiceInfo 及 ServiceInfo 中的 ServiceKey
    ServiceList ss = proxy.find_service(null, serviceName, null, 0);
    Vector serviceInfoVector = ss.getServiceInfos().getServiceInfoVector();
    ServiceInfo serviceInfo = (ServiceInfo) serviceInfoVector.elementAt(0);
    String serviceKey = serviceInfo.getServiceKey();
//透過 ServiceKey 取得 ServiceDetail 及 AccessPoint
    ServiceDetail sdetail = proxy.get_serviceDetail(serviceKey);
    BusinessService bservice = (BusinessService) sdetail.getBusinessServiceVector().elementAt(0);
    Vector bindings = bservice.getBindingTemplates().getBindingTemplateVector();
    for (int k = 0; k < bindings.size(); k++) {
```



```

        BindingTemplate binding = (BindingTemplate) bindings.elementAt(k);
        jTextField3.setText("http://" + binding.getAccessPoint().getText());
    }
} catch (Exception e) {
    e.printStackTrace();
}

```



附錄圖 2 UDDI4J 程式流程

附錄表 20 客戶端與伺服器端溝通的程式碼

```

// client 與 server 溝通
// String endpoint = "http://#####:8080/axis/services/MyService";
//設定目標網路服務的 AccessPoint
String endpoint = jTextField3.getText();
String number = jTextField1.getText().substring(4);
System.out.println( endpoint);
//產生 Service 物件與 Call 物件
Service service = new Service();
Call call = (Call) service.createCall();
//設定位址與作業名、參數與回傳的資料型態
call.setTargetEndpointAddress( new java.net.URL(endpoint) );
call.setOperationName("getAucData");
call.addParameter( "arg1", XMLType.XSD_STRING, ParameterMode.IN);
call.setReturnType( org.apache.axis.encoding.XMLType.XSD_STRING );
//執行 invoke，並將查詢得到的字串指定為 rett

```

```

String ret = (String) call.invoke( new Object[] { number } );
String column=ret.substring(0,2); //傳回字串前兩個值為 column 數
String row=ret.substring(2,4); //傳回字串第三及第四個值為 row 數
String rett= ret.substring(4); //剩下字串為抓到的整個資料

// 回傳值由 Table 印出
if(Integer.parseInt(row)>0){
    //分割 columnName 及 data
    StringTokenizer st = new StringTokenizer(rett,"\\");
    int i=0;
    String[] rettt=new String[2];
    while (st.hasMoreTokens()) {
        rettt[i]=st.nextToken();
        i++;
    }
    //分割 columnName
    StringTokenizer st_columnNames = new StringTokenizer(rettt[0]);
    int j=0;
    String[] columnNames=new String[Integer.parseInt(column)];
    while (st_columnNames.hasMoreTokens()) {
        columnNames[j]=st_columnNames.nextToken();

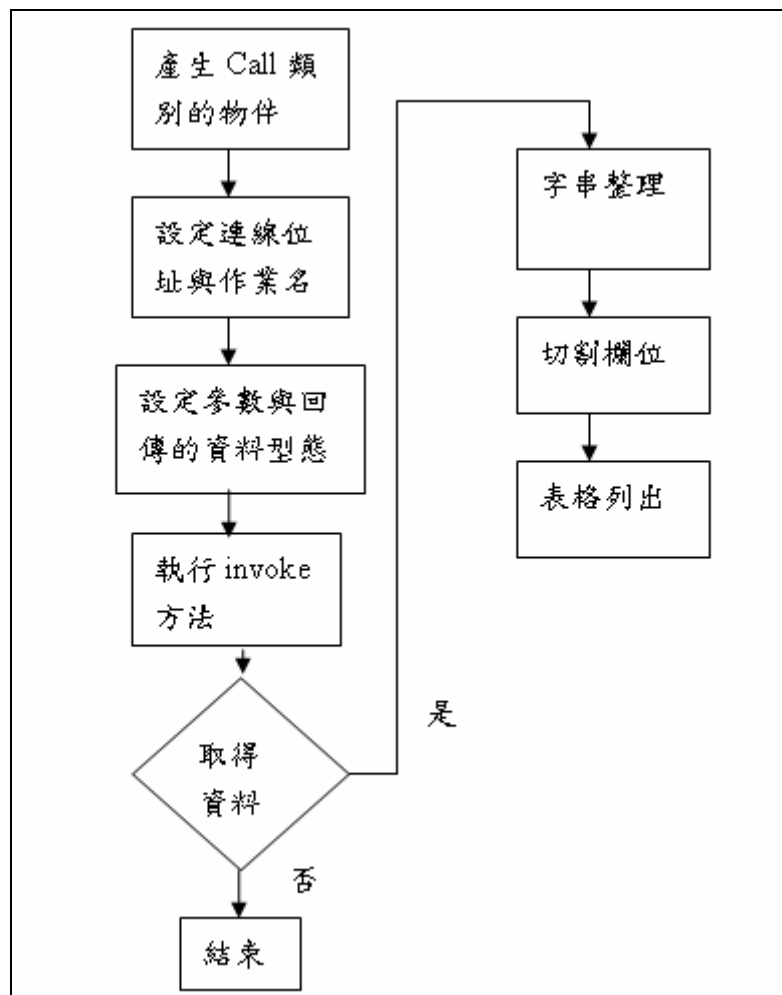
        //      System.out.println(columnNames[j]);
        j++;
    }
    //分割 data 成 2 維陣列
    StringTokenizer st_data_row = new StringTokenizer(rettt[1],"\n");
    StringTokenizer st_data;
    int k=0;
    String[] data_row=new String[Integer.parseInt(row)];
    Object[][] data=new String[Integer.parseInt(row)][Integer.parseInt(column)];
    while (st_data_row.hasMoreTokens()) {
        data_row[k]=st_data_row.nextToken();
        st_data= new StringTokenizer(data_row[k]);
        int m=0;
        while(st_data.hasMoreTokens()){
            data[k][m]=st_data.nextToken();
            m++;
        }
        k++;
    }
}

```

```

}
//由 table & Scroll 印出
table = new JTable(data, columnNames);
jScrollPane1 = new JScrollPane(table);
jScrollPane1.setBounds(new Rectangle(10,150,500,200));//table 所在位置
jPanel2.add(jScrollPane1);
}

```



附錄圖 3 客戶端與伺服器端溝通的程式碼流程

附錄五 資料轉換服務

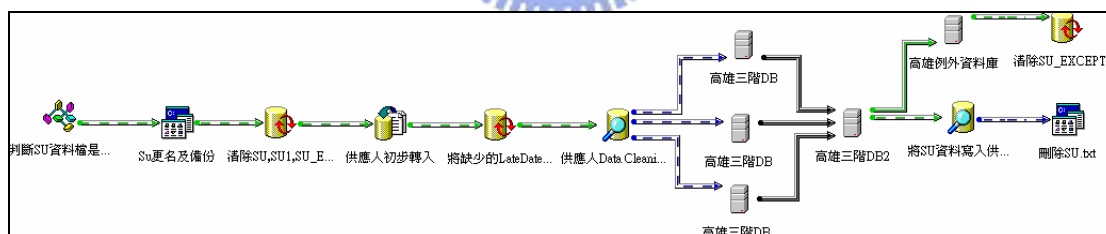
資料轉換服務建立時的基本四個步驟為(1)設定資料來源，(2)設定資料目的地，(3)設定轉換的方式，(4)利用程式修改資料格式及內容、設定執行方式，在本節中描述如何使用上述的四個基本步驟來建立資料轉換服務模組。

由於每一筆交易資料都得在供應人、承銷人、花卉品名等欄位成功的對應之下才能進行匯入，所以在資料轉換服務模組建立的考量下分為兩個部分。第一部分為資料轉換服務 1，先將供應人資料檔、承銷人資料檔、花卉品名資料檔匯入資料庫，分別對於供應人資料表、承銷人資料表、花卉品名資料表進行新增、修改、停用等動作。在資料轉換服務 1 完成後，再執行資料轉換服務 2，將交易資料檔匯入交易資料表中。

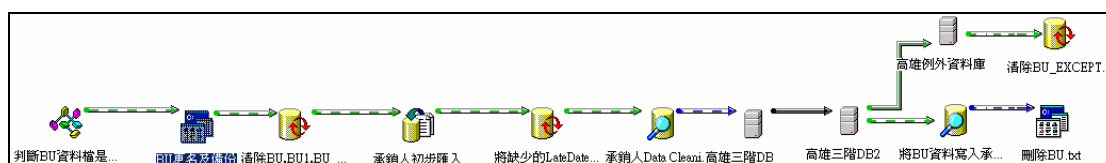
本附錄分為三小節，附錄 5.1 節的『資料轉換服務 1』說明如何將供應人資料檔、承銷人資料檔、花卉品名資料檔等原始資料匯入，進行新增、修改、停用等動作；附錄 5.2 節的『資料轉換服務 2』則說明匯入交易資料檔的過程，包括欄位對應失敗之後資料移至例外資料庫的動作；附錄 5.3 節將資料轉換服務 1 和資料轉換服務 2 結合在一起，並進行備份的動作，除此之外更介紹自動化排程，使得整個資料更新過程達到完全自動化。

附錄 5.1 資料轉換服務 1

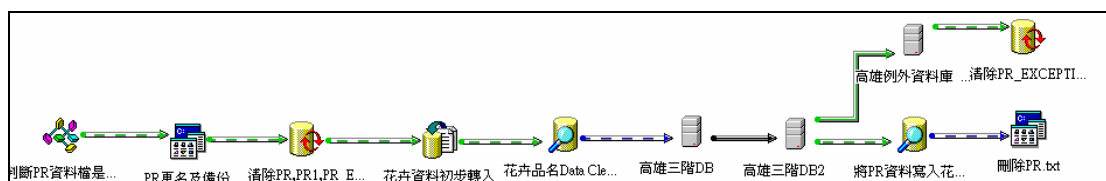
在資料轉換服務 1 中主要可分為幾個工作：(1)判斷資料是否存在，(2)資料更名及備份，(3)資料淨化，(4)異動操作判斷，(5)資料庫資料異動，(6)刪除原始檔。藉此將供應人資料檔、承銷人資料檔、花卉品名資料檔匯入資料庫中，供應人、承銷人及花卉資料的資料轉換服務流程如附錄圖 4。



a. 供應人資料轉換服務流程



b. 承銷人資料轉換服務流程



c. 花卉資料的資料轉換服務流程

附錄圖 4 供應人、承銷人及花卉資料的資料轉換服務流程

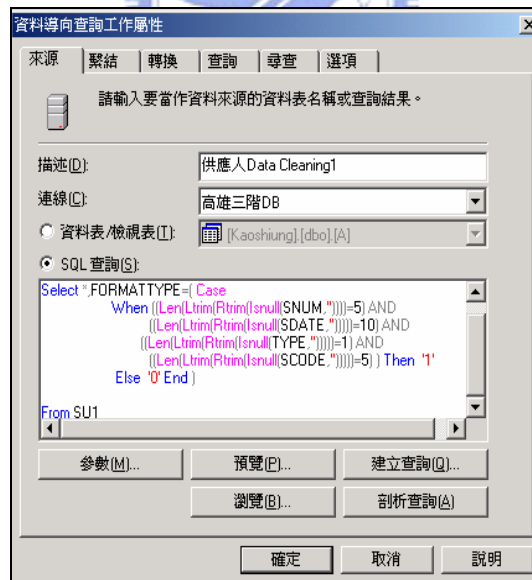
由於供應人資料較複雜且存在緩慢改變維度(Slowly Change Dimension)的問題，所以在供應人資料表中採用一個額外的識別碼，也就是 SUP_NUM 這個欄位。此外對於供應人資料的轉換也更加嚴謹，在進行匯入工作時分成三部份，分別為新增、修改及停用。除此之外三條流程內容大同小異，故在此以供應人資料轉換服務流程為例介紹整個轉換服務的架構與細節，其餘兩條將不再贅述。

一開始轉換服務先執行『判斷供應人資料檔(SU.TXT)是否存在』工作，若檔案存在則進行後續流程的資料更新工作；若否，則供應人資料轉換服務將不會執行後續轉換流程。

在指定的路徑 D:\Kaoshiung\Upload 下，程式會判斷第二十一個字元是否存在一個 S 的字元，也就是如果在該路徑存在 SU.txt 資料檔，D:\Kaoshiung\Upload\Su.txt，程式便會傳回成功的命令，反之則傳回失敗的命令。

確定供應人資料檔案存在後，接著呼叫一外部批次執行檔，運用 DOS 的命令將供應人資料檔更名及備份到其他資料夾，接著將暫存資料表清空，開始進行匯入的工作。經過初步的匯入至暫存資料表之後，先將部份欄位內容做更正，包括原供應人的 SUP_LASTDATE 欄位由 000000 將其設為 1900-01-01，及供應人的 TYPE 欄位由空值設為 9，也就是其他。

在資料經過謬誤修正之後，開始正式進行供應人資料資料淨化(Data Cleaning)的工作，如附錄圖 5。首先會針對供應人資料進行格式比對，包括欄位格式是否符合、資料表供應行政區欄位中的資料是否存在於供應人行政區資料表的 SUPLOC_LOCATION 欄位之中。如果不符合格式則將資料匯入 SU_EXCEPTION 資料表中，停止資料轉換的進行，待系統管理者修正資料後再重新進行資料轉換工作，以確保系統資料的正確性。



附錄圖 5 資料淨化

在進行資料淨化工作後，接著將淨化後的資料與三規正規化資料庫中資料進行比對以決定相關的異動操作。在此步驟中，異動操作會依資料比對的結果來決定資料異動的方式，包括新增、修改、停用三項，而新增、修改、停用判斷的程式碼見附錄表 21。

完成異動操作之後，結束供應人資料表、承銷人資料表、花卉品名資料表的更新，將例外資料表的內容匯出至例外資料庫的例外資料表，並把供應人等三個資料檔刪除，完成資料

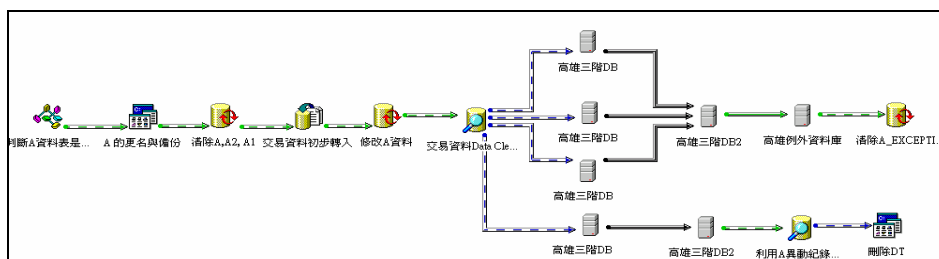
轉換服務 1 的整個作業流程。

附錄表 21 新增、修改、停用判斷之程式碼與說明

項目	內容	說明
新增	<pre>SELECT S.SNUM,S.SNAME,S.SDATE,S.SCODE,S.TYPE FROM SU1 As S WHERE (S.SNUM NOT IN (SELECT SUP_ONUM FROM 供應人資料表)) OR ((S.SNUM IN (SELECT SUP_ONUM FROM 供應人資料表 WHERE SUP_IFUSE='Y')) AND (S.SNAME NOT IN (SELECT SUP_NAME FROM 供應人資 料表 WHERE SUP_IFUSE='Y'))) OR (S.SNUM IN (SELECT SUP_ONUM FROM 供應人資料表 WHERE SUP_IFUSE='N') AND (S.SNAME NOT IN (SELECT SUP_NAME FROM 供應人資 料表 WHERE SUP_IFUSE='Y'))))</pre>	<ol style="list-style-type: none"> 1. 供應人代碼不存在於供應人資料表中。 2. 供應人代碼存在於供應人資料表中，但供應人姓名不存在於供應人資料表中。 3. 代碼存在於供應人資料表中，但已停用，且供應人姓名不存在於供應人資料表中。
修改	<pre>SELECT S.SNUM,S.SNAME,S.SDATE,S.SCODE,TYPE FROM SU1 As S WHERE ((S.SNUM IN (SELECT SUP_ONUM FROM 供應人資料 表 WHERE SUP_IFUSE='Y')) AND (S.SNAME IN (SELECT SUP_NAME FROM 供應人資料表 WHERE SUP_IFUSE='Y'))))</pre>	供應人代碼和姓名都存在於供應人資料表中。
停用	<pre>SELECT G.SUP_ONUM,G.SUP_NAME,G.SUP_LASTDATE,G.SUP_CODE,G. SUP_TYPE FROM SU1 As S, 供應人資料表 As G WHERE((S.SNUM=G.SUP_ONUM)AND (S.SNAME<>G.SUP_NAME)AND(G.SUP_IFUSE='Y'))</pre>	供應人代碼存在於供應人資料表中，但該代碼對應的姓名與資料表中的姓名不相同。

附錄 5.2 資料轉換服務 2

完成了資料轉換服務 1，也就是供應人資料、承銷人資料、花卉品名資料的更新，接著將交易資料檔匯入交易資料表，資料轉換服務 2 的架構如附錄圖 6。



附錄圖 6 花卉批發市場交易資料轉換服務流程

資料轉換服務 2 中，其主要工作流程近似於前述之資料轉換服務流程 1。其中與資料轉換服務 1 最大的不同處在於，此流程中主要將交易資料中問題資料的格式及內容做修正，並對於每日的交易資料提供一組具可唯一辨識性的編碼，用以做為資料庫中拍賣資料表中的主鍵。

在交易資料的修正方面，由於欄位較多，而且有對應上的問題，資料需經過處理的過程較多，詳情見附錄表 22。

附錄表 22 交易資料修正

項目	內容	解說
修補 ATYPE 為 NULL	UPDATE A2 SET ATYPE='1' where ATYPE is null	將 ATYPE 欄位為空值更正為 1。
修補殘貨交易紀錄的承銷人	UPDATE A2 SET ABU='00000', ATYPE='0' WHERE RTRIM(ISNULL(ABU,''))='XXXXX'	更正殘貨的資料的承銷人代碼欄位及類型欄位。
修補 NUM2 不滿兩位數者	UPDATE A2 SET ANUM2='0'+ Right(Rtrim(ANUM2),1) WHERE((Len(Rtrim(ANUM2))=1)AND Right(Rtrim(ANUM2),1)in ('1','2','3','4','5','6','7','8','9','0'))	補齊交易子序號的位數。
修補 ADATE 為由西元轉為民國	UPDATE A2 SET ADATE=ADATE-19110000	將交易日期從西元轉為民國。
修補拍賣線不規則值令其為 D 拍賣線	UPDATE A2 SET ALINE='4' WHERE ALINE NOT IN ('0','1','2','3')	將拍賣線謬誤的資料更正為 4，也就是其他。
修補 ANUM2 為 NULL	UPDATE A2 SET ANUM2='20' WHERE ANUM2 is NULL	交易子序號如為空值則設為 20。
修補 FLOWER_RANK1	UPDATE A2 SET ARANK='B' WHERE ((Len(Rtrim(ARANK))=1) AND Right(Rtrim(ARANK),1) in ('1','2','3','4','5','6','7','8','9','0'))	花卉等級只有一個字元，且存在於 0~9 之間，更正為 B。
修補 FLOWER_RANK2	UPDATE A2 SET ARANK='A'+ Right(Rtrim(ARANK),1) WHERE ((Len(Rtrim(ARANK))=2) AND Left(Rtrim(ARANK),1) = '1' AND Right(Rtrim(ARANK),1) in ('1','2','3','4','5','6','7','8','9','0','+','-'))	花卉等級有兩個字元，且第一個字元為 1，更正為 A，例如 12->A2。

在資料異動部份，程式會檢查交易資料中的供應商代碼、承銷人代碼、花卉品名代碼是否有分別對應至供應商資料表中的供應商代碼、承銷人資料表中的承銷人代碼、花卉品名資料表中的花卉品名代碼，三者判斷的程式碼，見附錄表 23。

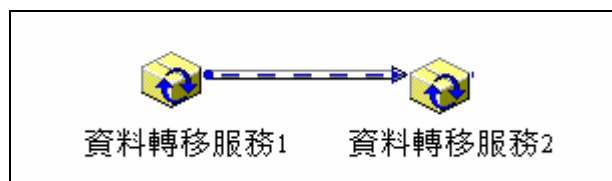
附錄表 23 交易資料與供應商、承銷人、花卉品名資料表的比對

項目	內容	說明
供應商代碼比對	SELECT * FROM A1 WHERE ASU NOT IN (SELECT SUP_ONUM FROM 供應商資料表 WHERE SUP_IFUSE='Y')	交易資料中的供應商代碼不存在於供應商資料表中。
承銷人代碼比對	SELECT * FROM A1 WHERE ABU NOT IN (SELECT BUY_BUYNUM FROM 承銷人資料表)	交易資料中的承銷人代碼不存在於承銷人資料表中。
花卉品名代碼比對	SELECT * FROM A1 WHERE APR NOT IN (SELECT FLOWER_NUM FROM 花卉資料表)	交易資料中的花卉品名代碼不存在於花卉品名資料表中。

將比對失敗的資料移至 A_EXCEPTION 資料表，剩餘資料轉入交易資料表，並將例外資料表的內容匯出至例外資料庫的例外資料表，及交易資料檔刪除，完成資料轉換服務 2 的整個作業流程。

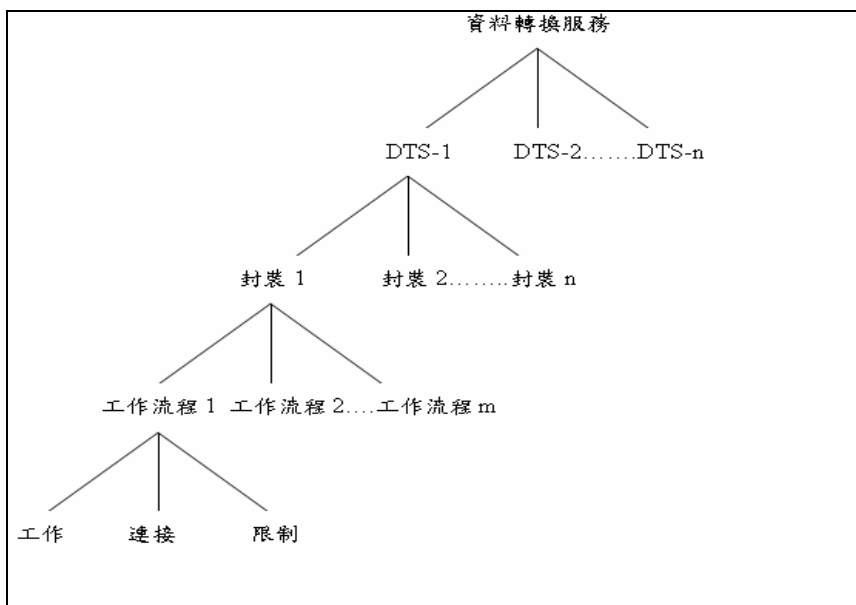
附錄 5.3 自動化排程

完成資料轉移服務 1、2 的設計之後，接下來則設定自動化排程，可以免除每日接到傳送的資料後，需再使用手動方式轉換資料，更使得整個系統的更新可以達到完全的自動化。



附錄圖 7 系統更新轉換程序

在附錄圖 7 中，系統更新轉換程序主要是將之前設計完成之資料轉換服務 1 及資料轉換服務 2 依設計的流程批次執行處理。附錄圖 8 為資料轉換服務作業樹狀結構圖。由圖中可看出，資料轉換服務作業是由數個封裝(Package)組成，在封裝中則包含了數個作業流程(Workflow)。而在每一工作流程中又包含了連線(Connection)、工作(Task)、限制(Constraint)等三種主要類型的作業。附錄表 24 為系統更新轉換程序中的資料轉換服務作業統計表。

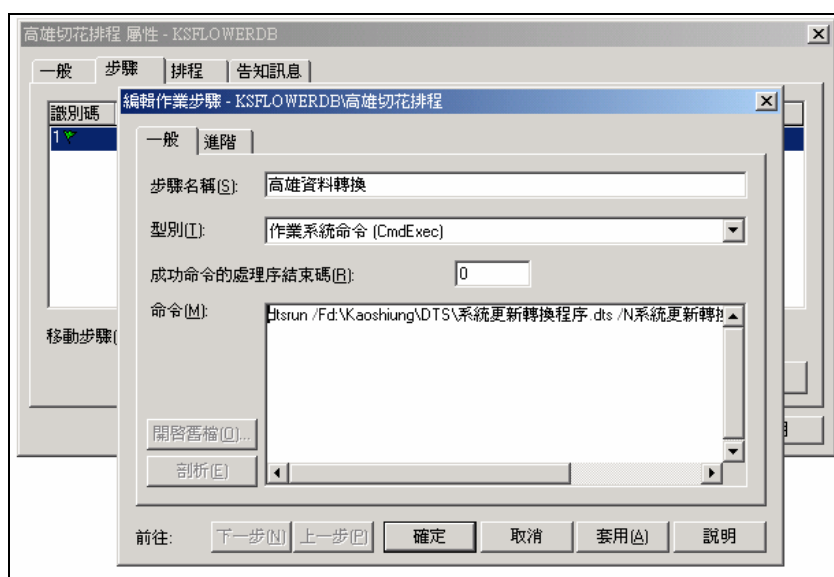


附錄圖 8 資料轉換服務作業樹狀結構圖

附錄表 24 資料轉換服務工作統計表

	封裝	工作流程	連線	工作	限制
DTS	封裝-1(資料轉換服務 1)	工作流程-1	2	14	12
		工作流程-2	2	12	10
		工作流程-3	2	11	9
	封裝-2(資料轉換服務 2)	工作流程-1	2	16	13

在完成系統更新轉換程序設計後，接著介紹自動化排程設計方式，在此採用的是 SQL Server2000[3][4]套裝軟體中的排程設定。在步驟內的命令指定執行『系統更新轉換程序.dts』，如附錄圖 9，在排程內的排程類型設為重覆執行。



附錄圖 9 自動化設定排程語法