

研究生：溫師翰

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

隨著越來越多的國際標準諸如 HACCP、ISO 22000 等的問世，食品追溯系統成為食品供應鏈中監控食品安全的有用工具。本研究提出使用網路服務技術以建構射頻識別相容的海鱷追溯系統。在這項新方法的架構下，以網路為基礎的追溯系統包含了服務提供者，服務仲介者以及服務需求者。由操作觀點而言，每個服務提供者利用網路服務描述語言文件發送本身的服務規格到通用描述、探索與整合仲介端註冊。每當服務需求者請求服務追溯一項食品的安全紀錄時，這項請求會將簡單物件存取協定為基礎的條碼或射頻識別標籤信息透過網際網路送到服務仲介端。接著服務仲介端會回應相對應的網路服務描述語言文件，將此請求轉傳到該項服務提供者以提供更多食品安全資訊的細節內容。進一步而言，有兩項主要的軟體成果已經達成。其一是稱為工作日誌的專用軟體系統，此系統實作於供應端，用以顯示如何扮演服務提供者的角色，例如蒐集食品安全相關資料，傳送網路服務描述語言文件等。另一項是附加條碼以及射頻識別介面的使用者友善介面的入口，此入口被用來展示服務需求端與服務仲介端之間的資訊互動。

關鍵詞：

網路服務

網路為基礎的追溯系統

工作日誌

入口

射頻識別介面

# Web Services Approach to a RFID-compatible Traceability System for Cobia

Student : Shih-Han Wen

Advisor : Dr.Gau-Rong Liang

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

## Abstract

Since the advent of more and more international standards such as HACCP, ISO 22000, etc, food traceability system becomes a useful management tool for monitoring food safety in food supply chains. In this thesis, web services technology has been proposed for constructing a RFID-compatible traceability system for cobia. Within the new approach, the web-based traceability system essentially consists of service providers, services brokers, and service requesters. From an operational viewpoint, each service provider submits its own service specification through WSDL documents to the UDDI registry of any service broker. Whenever a service requester asks for a service for tracing the safety record of a given food, the request is submitted by a SOAP-based barcode or RFID tag message through Internet to a service broker. Then the service broker replies a corresponding WSDL document which can be forwarded to a dedicated service provider for offering more detailed food safety information. Furthermore, two major software efforts have been achieved. One is a specific software system named working log which has been implemented at the supplier side for showing how to play the role of service provider such as collecting safety data, submitting WSDL documents, etc. The other is a user friendly portal with barcode and RFID interface which has been designed for showing how to interact between a service requester and a services broker.

Keywords :

Web Services

Web-based Traceability System

Working Log

Portal

RFID Interface

## 誌謝

本研究論文的順利完成，首要感謝我的指導教授梁高榮老師在研究過程以及論文寫作時給予寶貴的建議與指導，使得本研究的深度與廣度能夠完備周全。我自大三以專題生的身分進入實驗室以來，梁老師不但教授課業的知識，更藉由諸多研究專案的實作過程，訓練我獨立思考與實作的能力，使我有更完善的技能與勇氣來面對未來，在此致上由衷的感謝。同時感謝唐麗英老師、張永佳老師以及王春和老師在本論文口試與撰寫上給予諸多指導意見，使本論文更趨完備。

在實驗室的生涯中，要感謝實驗室諸位學長姐重豪、仁偉、漢清、哲煜、雍仁、軒豪、英泰、楓凱、士凱、淙亮、詩涵、宗沂、彥修、珮捷、伊婷、致穎、瞬正、耿豪、哲正、公麒、啟宗、昭志等，不論在研究計畫的指導與合作，實驗室生活的提點與同歡都給我莫大的鼓勵與信心。實驗室同學家瑜、佳佑在課業上的互相幫助砥礪，研究計畫的互相討論與思考，都讓我更加成長；同窗好友啟峰、豪君、政翰、正琪、長科、修來、昱皓、珮君、柏詠等 002、005 以及 006 實驗室的成員們，因為有你們，使我的研究所生涯更加多姿多采；加上學弟妹新凱、炯棠、俊端、昶年、彥志、潔妤、音帆分享生活上的點點滴滴，為研究所增添更多美好的回憶；特別感謝好友忠穎、俊宏對於我的研究專案提供自身的經驗與想法，在我徬徨迷惑時給予我適時的打氣與建議，在此深致謝意。另外對於實驗室日常事務的維運，在此對於前後任的研究助理麗芳與庭楨深表謝意，讓實驗室所有夥伴無後顧之憂，能夠專注於自身的研究工作。

在實驗室期間，曾參與農委會之研究計畫：「花卉資料庫正規化、資料倉儲設計及 XML 規格連線工程」計畫(計畫編號：92B022)、「台灣區花卉資訊電子化建置與應用服務計畫(II)」(計畫編號：93W527)、「花卉批發市場交易作業標準化之研究」(計畫編號：93W534)、「台灣區花卉資訊電子化建置與應用服務計畫(III)」(計畫編號：94I819)、「農產品批發市場作業流程之研究」(計畫編號：94I817)；以及漁業署研究計畫：「RFID 技術在漁業上的加值應用研究」(計畫編號：94I812)、「漁產品產銷履歷資訊建置與整合」(計畫編號：94I825)。在這些研究計畫的執行過程中，對於我的研究提供了許多寶貴的資源與幫助，並增益我對於專案實施的技巧與經驗，特此感謝。

最後要感謝我的父母親人，無論是物質上或是精神上，都給予我最大的支持與幫助，陪伴我分享生活中的甘苦。感謝所有陪伴我度過碩士生涯的師長與朋友們一路的支持與鼓勵。

# 目錄

中文摘要.....	I
ABSTRACT.....	II
誌謝.....	III
圖目錄.....	VI
表目錄.....	IX
<b>第一章 緒論.....</b>	<b>1</b>
1.1 研究動機.....	1
1.2 問題界定.....	3
1.3 研究目的.....	5
1.4 研究方法.....	7
1.5 論文架構.....	9
<b>第二章 文獻回顧.....</b>	<b>10</b>
2.1 產銷履歷的流程.....	10
2.1.1 食品安全相關制度說明.....	10
2.1.2 養殖水產品供應鏈流程.....	11
2.1.3 產銷履歷與資訊技術的關係.....	15
2.2 射頻識別技術.....	17
2.2.1 EPC Network 架構.....	17
2.2.2 射頻識別標籤、標牌與編碼.....	18
2.2.3 射頻識別讀寫器.....	19
2.2.4 中介軟體架構與資訊服務系統.....	20
2.2.5 射頻識別技術的困境.....	21
2.3 網路服務技術.....	23
2.3.1 服務導向架構.....	23
2.3.2 簡單物件存取協定.....	24
2.3.3 網路服務描述語言.....	26
2.3.4 通用描述、發現與整合.....	27
2.3.5 業務流程描述語言.....	28
2.4 射頻識別與網路服務技術的分析結論.....	30
<b>第三章 工作日誌的設計.....</b>	<b>31</b>
3.1 海鱸供應鏈作業流程分析.....	31
3.1.1 海鱸供應鏈作業 IDEF0 規格.....	31

3.1.2 海纜供應鏈作業流程裴氏圖.....	36
3.2 工作日誌系統三階正規化分析與設計 .....	38
3.2.1 設計階層式架構模型.....	38
3.2.2 工作日誌系統從 IDEF0 到三階正規化 IDEF1X 規格.....	39
3.2.3 工作日誌系統三階正規化資料庫規格.....	43
3.3 工作日誌系統 .....	50
3.3.1 工作日誌系統設計裴氏圖.....	50
3.3.2 工作日誌系統操作說明.....	51
3.4 其他工作日誌設計 .....	56
<b>第四章 履歷追蹤查詢系統架構－ 伺服器端設計 .....</b>	<b>57</b>
4.1 使用 AXIS 實作網路服務專案 .....	57
4.1.1 海纜生產供應鏈追溯與追蹤系統架構設計.....	57
4.1.2 Axis 的運作模式.....	59
4.1.3 網路服務內容架構.....	61
4.2 JWS(JAVA WEB SERVICE)建置簡單網路服務伺服器端程式.....	63
4.2.1 伺服器端服務類別設計.....	63
4.2.2 在 Axis Server 部署服務伺服器端程式.....	66
4.3 WSDL-ORIENTED 建置網路服務伺服器端程式 .....	68
4.3.1 WSDL 文件設計 .....	68
4.3.2 使用 WSDL2Java 產生服務類別程式碼.....	71
4.3.3 在 Axis Server 部署服務伺服器端程式.....	74
<b>第五章 履歷追蹤查詢系統架構- 服務需求端設計 .....</b>	<b>75</b>
5.1 網路服務需求端設計 .....	75
5.2 仲介端登錄、發佈與查詢服務 WEB-BASED 的操作 .....	77
5.3 以 NON-WEB API 設計服務需求端與仲介端的溝通 .....	82
5.3.1 仲介端可展加註語言(XML)文件規格.....	82
5.3.2 使用 UDDI4J 解析仲介端可展加註語言文件.....	84
5.4 射頻識別讀寫功能的設計 .....	87
5.4.1 射頻識別硬體架構.....	87
5.4.2 射頻識別軟體架構.....	89
5.4.3 射頻識別系統的操作.....	90
5.5 服務需求端整合入口的操作 .....	93
<b>第六章 結論與未來方向 .....</b>	<b>96</b>
6.1 結論 .....	96
6.2 未來發展方向與建議 .....	98
<b>參考文獻.....</b>	<b>99</b>



## 圖目錄

圖 1.1 網路服務技術與產銷履歷整合服務架構圖 .....	5
圖 1.2 網路服務連接工作日誌而形成的分散式產銷履歷系統 .....	6
圖 1.3 研究方法架構 .....	8
圖 1.4 論文架構圖 .....	9
圖 2.1 養殖漁產品產銷供應鏈架構圖 .....	11
圖 2.2 飼料廠到養殖場的資訊追溯模式 .....	12
圖 2.3 養殖場到物流運輸的資訊追溯模式 .....	13
圖 2.4 物流運輸到加工廠的資訊追溯模式 .....	13
圖 2.5 加工廠到銷售端點(消費者)的資訊追蹤模式 .....	14
圖 2.6 射頻識別技術的產銷履歷模式 .....	15
圖 2.7 射頻識別技術與條碼技術的成本比較 .....	16
圖 2.8 條碼與網路服務技術並行的產銷履歷模式 .....	16
圖 2.9 EPC NETWORK架構圖 .....	17
圖 2.10 SOAP ENVELOPE架構 .....	25
圖 2.11 網路服務描述架構與網路服務描述語言對應圖 .....	26
圖 2.12 通用描述、發現與整合在網路服務架構的角色 .....	27
圖 2.13 重要資料節點的對應關係 .....	28
圖 2.14 BPEL4WS整合網路服務的架構圖 .....	29
圖 3.1 海鱸供應鏈生產作業流程IDEF0 表達法 .....	31
圖 3.2 海鱸供應鏈生產作業四大流程IDEF0 圖 .....	32
圖 3.3 海鱸供應鏈魚苗及原物料驗收作業流程IDEF0 圖 .....	33
圖 3.4 海鱸供應鏈儲藏設施與作業流程IDEF0 圖 .....	34
圖 3.5 海鱸供應鏈養殖作業流程IDEF0 圖 .....	35
圖 3.6 海鱸供應鏈包裝出貨作業流程IDEF0 圖 .....	36
圖 3.7 階層轉換法 .....	36
圖 3.8 海鱸供應鏈作業流程裴氏圖 .....	37
圖 3.9 階層式架構模型圖 .....	38
圖 3.10 設計階層與工作日誌資料庫設計關係對照圖 .....	39
圖 3.11 三綱要模型圖 .....	40
圖 3.12 概念性綱要IDEF1X規格 .....	41
圖 3.13 「藥品供應商」與「用藥紀錄」多對多關聯表的拆解 .....	41
圖 3.14 內部綱要IDEF1X規格 .....	42
圖 3.15 外部綱要五大模組與應用程式模組對應圖 .....	43
圖 3.16 工作日誌系統裴氏圖 .....	50
圖 3.17 魚苗供應商/魚苗資料維護介面 .....	51

圖 3.18 飼料供應商/飼料資料維護介面 .....	51
圖 3.19 藥品供應商/藥品資料維護介面 .....	52
圖 3.20 養殖戶基本資料介面 .....	52
圖 3.21 養殖場基本資料介面 .....	52
圖 3.22 放養紀錄介面 .....	53
圖 3.23 飼料餵養紀錄介面 .....	53
圖 3.24 用藥紀錄介面 .....	54
圖 3.25 收穫紀錄介面 .....	54
圖 3.26 水質檢測紀錄介面 .....	54
圖 3.27 漁產品檢驗紀錄介面 .....	55
圖 3.28 承銷商資料維護介面 .....	55
圖 3.29 出貨單維護介面 .....	55
圖 4.1 追溯與追蹤系統實作架構.....	58
圖 4.2 WEB SERVICES技術架構圖 .....	58
圖 4.3 AXIS組成元件架構圖 .....	59
圖 4.4 AXIS用於伺服端的運作架構.....	60
圖 4.5 AXIS用於服務需求端的運作架構.....	61
圖 4.6 EAN.UCC系統使用於養殖水產供應鏈.....	62
圖 4.7 JWS方法建置流程IDEF0.....	63
圖 4.8 JDBC架構圖 .....	64
圖 4.9 基本資料服務 .....	65
圖 4.10 檢驗表單內容服務 .....	65
圖 4.11 飼料與餵養紀錄服務 .....	66
圖 4.12 藥品紀錄服務 .....	66
圖 4.13 TCP MONITOR驗證服務運作狀況 .....	67
圖 4.14 WSDL-ORIENTED建置流程.....	68
圖 4.15 TANHOU.WSDL文件以ALTOVA XMLSPY展開.....	69
圖 4.16 TANHOU.WSDL文件綱要 .....	69
圖 4.17 WSDL的DEFINITIONS元素.....	70
圖 4.18 WSDL的MESSAGE元素.....	70
圖 4.19 WSDL的PORTTYPE元素 .....	70
圖 4.20 WSDL的BINDING元素 .....	71
圖 4.21 WSDL的SERVICE元素 .....	71
圖 4.22 ORG.APACHE.AXIS.WSDL.WSDL2JAVA工具執行命令.....	71
圖 4.23 WSDL2JAVA工具執行完成結果 .....	72
圖 4.24 使用WSDL2JAVA產生的程式碼類別圖 .....	73
圖 4.25 執行DEPLOY.WSDD以及部署成功結果.....	74
圖 5.1 使用AXIS API套件設計服務需求端流程 .....	75

圖 5.2 開發服務需求端使用CALL類別方法的程式碼 .....	76
圖 5.3 UDDI技術架構圖 .....	77
圖 5.4 商業實體的登錄註冊 .....	78
圖 5.5 發佈服務操作流程 .....	78
圖 5.6 新增商業實體 .....	79
圖 5.7 新增服務 .....	79
圖 5.8 新增存取點 .....	80
圖 5.9 新增技術模型實體 .....	80
圖 5.10 服務成功發佈 .....	81
圖 5.11 依照服務搜尋 .....	81
圖 5.12 可展加註語言文件的資料結構對應 .....	82
圖 5.13 服務需求端應用程式與服務仲介端互動關係 .....	85
圖 5.14 使用UDDI4J套件取得服務需求端所需資訊的程式說明 .....	86
圖 5.15 INTERMEC IF5 射頻識別裝置架構圖 .....	87
圖 5.16 射頻識別裝置的訊號讀取流程 .....	88
圖 5.17 射頻識別標籤規格 .....	88
圖 5.16 射頻識別讀寫功能溝通架構 .....	89
圖 5.17 IF5 開機啟動畫面 .....	90
圖 5.18 操作參數的設定畫面 .....	91
圖 5.19 單行指令列執行BRI .....	92
圖 5.20 段落式腳本執行BRI .....	92
圖 5.21 產銷履歷入口與網路服務整合執行流程 .....	93
圖 5.22 產銷履歷入口操作說明 .....	94
圖 5.23 產銷履歷服務畫面 .....	95
圖 6.1 可展加註語言環境下的安全機制 .....	98



## 表目錄

表 2.1 第 1 級標籤(96 位元)的資料格式 .....	18
表 2.2 含水分木棧板的有效讀取距離測試結果(單位：公尺).....	22
表 2.3 SOAPHEADER 的屬性值 .....	25
表 2.4 射頻識別標籤與條碼式標籤比較表 .....	30
表 3.1 養殖場資料檔資料表 .....	44
表 3.2 養殖場明細檔資料表 .....	44
表 3.3 箱網基本檔資料表 .....	44
表 3.4 水質檢測紀錄 .....	44
表 3.5 放養資料資料表 .....	45
表 3.6 飼養紀錄資料表 .....	45
表 3.7 飼料明細檔資料表 .....	45
表 3.8 飼料供應商資料表 .....	46
表 3.9 用藥紀錄資料表 .....	46
表 3.10 藥品明細檔資料表 .....	46
表 3.11 藥品供應商資料表 .....	46
表 3.12 魚苗進貨資料資料表 .....	47
表 3.13 魚苗供應商資料表 .....	47
表 3.14 收穫紀錄資料表 .....	47
表 3.15 檢驗表單資料表 .....	47
表 3.16 檢驗明細資料表 .....	48
表 3.17 出貨記錄資料表 .....	48
表 3.18 承銷商資料資料表 .....	48
表 3.19 運輸物流日誌基本資料表 .....	56
表 3.20 加工廠日誌基本資料表 .....	56
表 3.21 販售商日誌基本資料表 .....	56
表 4.1 WSDL 文件設計要點與對應元素 .....	68
表 4.2 WSDL 與 JAVA 程式碼繫結規則 .....	72
表 5.1 BUSINESSENTITY 使用元素說明 .....	83
表 5.2 BUSINESSSERVICE 使用元素說明 .....	83
表 5.3 BINDINGTEMPLATE 使用元素說明 .....	84
表 5.4 TMODEL 使用元素說明 .....	84
表 5.5 UDDI4J 主要 API 結構與功能對照表 .....	85
表 5.6 射頻識別讀寫器的原理所對應之 JAVA 程式碼.....	90

# 第一章 緒論

本章內容分為五個部份，分別為 1.1 節「研究動機」，1.2 節「問題界定」，1.3 節「研究目的」，1.4 節「研究方法」，1.5 節「論文架構」。

## 1.1 研究動機

近年來，國內外食品安全問題的發生日漸頻繁，從常見的食物中毒到近年來歐美狂牛症、禽流感，以及農漁產品藥物殘留等問題，讓消費者日漸重視食品的衛生與安全，而世界各國也愈來愈重視食品安全相關的要求與規範，並已陸續推動 HACCP (Hazard Analysis and Critical Control Point)[48][27]、GAP(Good Agriculture/Aquaculture Practice)與 ISO 22000 Food safety management system- Requirements for any organization in the food chain(食品安全管制系統-食品供應鏈所有業者要求)等制度[42]。歐盟為了因應狂牛症問題，自 1997 年起推動並倡導食品來源的「可追蹤性」，作為食品安全管理的重要手段之一[10]，並預計於 2010 年要求進入歐盟的食品需提供產銷履歷供查詢；日本也在 2004 年 12 月 1 日開始實施牛隻產銷履歷紀錄制度，並於 e-Japan 戰略重點政策中提出「食料品履歷情報有關係統的導入」以及「牛肉履歷情報有關係統的普及」等兩項計畫，將食品產銷履歷制度正式導入農產品實施，並成立「共同生產履歷中心」，明確指出在 2010 年實現所有食品導入產銷履歷的目標，屆時若沒有取得認證的食品均無法進口至日本；而韓國也在 2005 年推動環保型農業實施產銷履歷管理制度，由該國 GAP 農民開始導入蔬果產業的食品安全體系與回溯系統，預計於 2006 年蔬果全面實施相關措施。世界各國強制性建立食品安全控管及追蹤體系的措施，紛紛要求其農漁產品進口國須提出產品認證，以保障消費者食的權益。[3]

舉凡「從農場到餐桌」的所有食物製作流程，都必須在符合政府所頒定的食品安全標準下，做到標示清晰和資訊可追蹤的要求[22]，也就是所謂的「產銷履歷制度」。目前，結合食品產銷履歷作為食品安全風險控管的有效手段，已成為推動食品安全的核心工作。對養殖漁產品而言，產銷履歷資訊就是追蹤養殖漁產品從原物料生產、養殖直到銷售過程的相關資訊。從養殖產品的飼料與藥品來源供應、魚苗來源與養成、運銷、加工集散，一直到銷售的產銷過程中，每一階段的資訊都可以向上游追溯或下游追蹤，以瞭解養殖漁產品的來源、加工廠、運輸流程以及銷售點之間的紀錄與資訊[4][14]。

針對我國的漁業產銷供應鏈體系而言，日本是我國漁產品外銷的最大市場，為了符合日本市場的要求，因此產銷履歷制度為目前政府大力輔導推動之項目。2003 年十月台灣輸日鰻魚連續被檢測出磺胺二甲嘧啶，導致日本對台灣輸日鰻魚產品實施命令檢查(逐批檢驗才可放行)；2004 年十月輸日活鰻與加工鰻產品又遭日本檢驗出殘留禁用的抗菌劑恩諾沙星，使得甫遭日方宣布解除的命令檢查又重新啟動。屢次遭到日方檢驗出禁藥問題導致鰻農、運銷盤商以及政府主管單位互相究責與不信任，卻無法迅速精確地追查問題源頭。而 2005 年台灣輸往香港的石斑魚事件，由於遭到香港方面檢測出含有禁藥孔雀石綠，遭到香港強制下架；而我國政府後續緊急啟用的安全驗證標章制度，竟又再次遭到香港驗出孔雀石綠；而此案經我國主管單位追查月餘，仍無法查出問題石斑魚的來源。上述的事件不但使政府公權力遭受質疑，更使得台灣的國際形象遭到嚴重傷害。

而在供應鏈中物流技術的載具方面，由於射頻辨識技術(Radio Frequency Identification

Technology, RFID Technology)[2]的成熟與商品化，2003年六月美國Wal-Mart零售系統要求其前一百大供應商在2005年一月使用射頻辨識技術出貨，並於2006年底所有的供應商採用射頻辨識技術出貨[15]。受到此巨額採購量之影響，造成全球民生用品供應鏈的物流辨識技術產生重要的變革，從傳統的條碼技術轉成無線電磁波導向的射頻辨識技術。對漁業界而言，美國Beaver Street Fisheries企業(B.S.F.)已建置射頻辨識資訊系統來應付此挑戰，B.S.F.是一個經營超過50年的跨國冷凍海產專業公司，其研發導入的射頻辨識系統使得此跨國的供應商能繼續存活於Wal-Mart的市場裡。由於射頻辨識技術是以無線電波為基礎，而無線電波的頻率使用受到各種不同產業特性與外在環境的複雜因素所影響。因此，在漁產品產銷履歷系統的設計發展過程中，同步探討射頻辨識的規格標準，研究如何從射頻辨識標籤透過網際網路追蹤並獲得養殖場的產銷履歷資訊，讓國內的漁產品供應鏈的資訊通透性更加地完善。



## 1.2 問題界定

我國現行的養殖漁產品資訊系統的制度有兩項缺失：生產業者配合度不佳與無法讓供應鏈中各層級的資訊紀錄合作進行追溯與追蹤的功能。以生產業者配合度不佳的問題而言，主要是因為優良養殖場系統採用傳統三層式架構，資料庫為網路型集中式資料庫，全國參與優良養殖場系統的生產者都必須透過網頁的操作介面將生產紀錄登記在此資料庫。但是各生產者之間處於同產業競爭狀態，因此會造成對於生產紀錄的資料洩漏或是被其他競爭對手取得的疑慮；而就政策實行面而言，優良養殖場系統屬於政府漁業署監管範圍，生產業者對於所有紀錄資料完全處於政府相關單位監管下心生畏懼。在此疑慮之下造成兩項結果：生產者不願意使用優良養殖場系統或是生產者登記不實紀錄。由於優良養殖場系統是中央集中式的資料庫系統，因此只有養殖業者與政府主管單位可以使用該系統內的資訊。但是對於養殖漁產品供應鏈而言，上游原物料供應商、下游的通路銷售廠商與消費者皆無法取得相關資訊。因此，對整條供應鏈而言並沒有追溯或追蹤的功能。也導致諸如台灣鯛銷歐盟氯黴素殘留、輸日鰻魚的恩諾沙星殘留與輸往香港石斑魚孔雀石綠等事件，卻無法在有效時間內找出問題魚隻的來源，或是發生供應鏈上各端點的業者互相推卸責任，而政府主管單位卻無從分辨真假等問題

本論文將問題界定於海鱸產銷供應鏈中紀錄保存以及資訊交換的建置架構。過去在供應鏈上的資訊傳遞有眾多技術被提出與實作，例如郵寄、傳真以及網路等。由於近年來許多新技術發展成熟，所以在傳輸模式以及資料格式轉換的彈性上皆有改善的空間，因此本研究設計獨立分散式的資料庫系統與應用程式讓生產業者追蹤記錄生產履歷資料—「工作日誌系統」；同時搭配網路服務技術架構達成分散式的整合追溯功能—「履歷追溯查詢系統」；而在供應鏈上各端點之間的交換識別，則使用射頻識別技術作為工具。如此的結合效用可用以下幾點來討論：

### 1. 資訊的正確性

在資訊傳遞方面，只要是經由人工處理，即有可能產生錯誤，例如筆誤、輸入操作失誤等。因此系統在建構上減少人為處理，並且在儲存與傳送過程中加入驗證機制，乃是本研究對於資料傳輸正確性的改善方式。

### 2. 分散式架構的接受度提升

不同於傳統三層式架構的中央式資料庫系統，分散於各養殖供應鏈業者的資訊系統可以透過流程分析，建構一套標準化的資料庫系統，各業者再各自針對特殊需求對此套系統進行微調。相較之下，中央式資料庫系統由於負荷量是全國的養殖漁產品相關業者，必須耗費巨額的成本與人力維護系統的正常運作；而分散式的系統在標準化過程中，只建構追蹤與追溯功能的核心需求，此目的有二：對成本而言，各廠商只需負擔少量的費用即可加入產銷履歷的追蹤與追溯體系；而對電腦操作而言，因為系統軟體盡可能簡化操作的項目與流程，而提升業者的接受度。

### 3. 資訊的傳送成本

資訊的傳送成本對於各企業而言也是一項成本負擔，然而隨著使用的技術不同，該項花費亦有所差異。例如使用郵寄的方式，郵資就是資訊的傳送成本。本研究將以網際網路為資料傳遞的途徑，由於現今網路的普及化，資訊傳送成本將可有效現制在網路費用範圍之內。另一方面，資訊的傳送應該降低傳送端與接收端之間的耦合性。收送訊息的雙方不會因為資



訊尚未傳送完成而造成作業的停滯，使得系統不需要花費額外的時間以及金錢成本在資訊傳遞。

#### 4. 載具的高記憶容量與重複使用性

相較於傳統的條碼或是人工書寫編號，射頻識別技術除了對環境的耐用性較好、可重複讀寫、遠距離讀寫等優點外，由於傳統條碼的容量有限，在養殖漁產品供應鏈的使用上，通常都必須一次使用兩到三張條碼。有鑑於此，本研究嘗試用高容量的 RFID 標籤來記錄識別編號，大幅降低讀寫次數的操作過程。





### 1.3 研究目的

基於前述這些角度的分析，本研究提出「網路服務技術與海鱸產銷履歷系統的整合」想法，圖 1.1 為設計架構圖。由於海鱸為我國養殖漁業的重要外銷產品，而且具有高額單價的特性，因此本研究以海鱸的產銷供應鏈為主要研究對象。此系統包含兩套子系統[16]：第一套子系統是工作日誌系統(Working Log)，該系統安裝於各家海鱸養殖場，將魚隻、飼料與藥品等日常工作記錄在各自的資料庫中，工作日誌系統將在第三章中說明。第二套子系統利用網路服務技術[23]來連接工作日誌等資訊系統的資料庫，網路服務技術共有三個主要角色：服務提供者、服務仲介者以及服務需求者。服務提供者是供應鏈中各層級的廠商，使用網路服務描述語言(WSDL)文件將提供的服務規格書註冊在服務仲介者端，第四章將說明服務提供者建立各自服務的流程；而服務仲介者使用通用描述、發現與整合(UDDI)技術，是網路服務架構中的第三方仲介，目前由各大資訊廠商免費支援，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；而服務需求端整合了條碼與射頻辨識功能成為產銷履歷入口，將讀取到的條碼或射頻識別標籤內容以簡單物件存取協定(SOAP)訊息送至服務仲介端查詢得到該產品所屬廠商提供的服務位址，隨後再向該位址取得產品相關的資訊並顯示在產銷履歷入口的頁籤內。如此不斷反覆查詢並向服務提供者取得產品相關資訊，最後可以將任何產品在供應鏈各端點經過的所有相關資訊整合顯示在產銷履歷入口的各頁籤之中。第五章將說明服務仲介者與產銷履歷入口之間的互動關係，以及產銷履歷入口與射頻識別系統的整合原理。

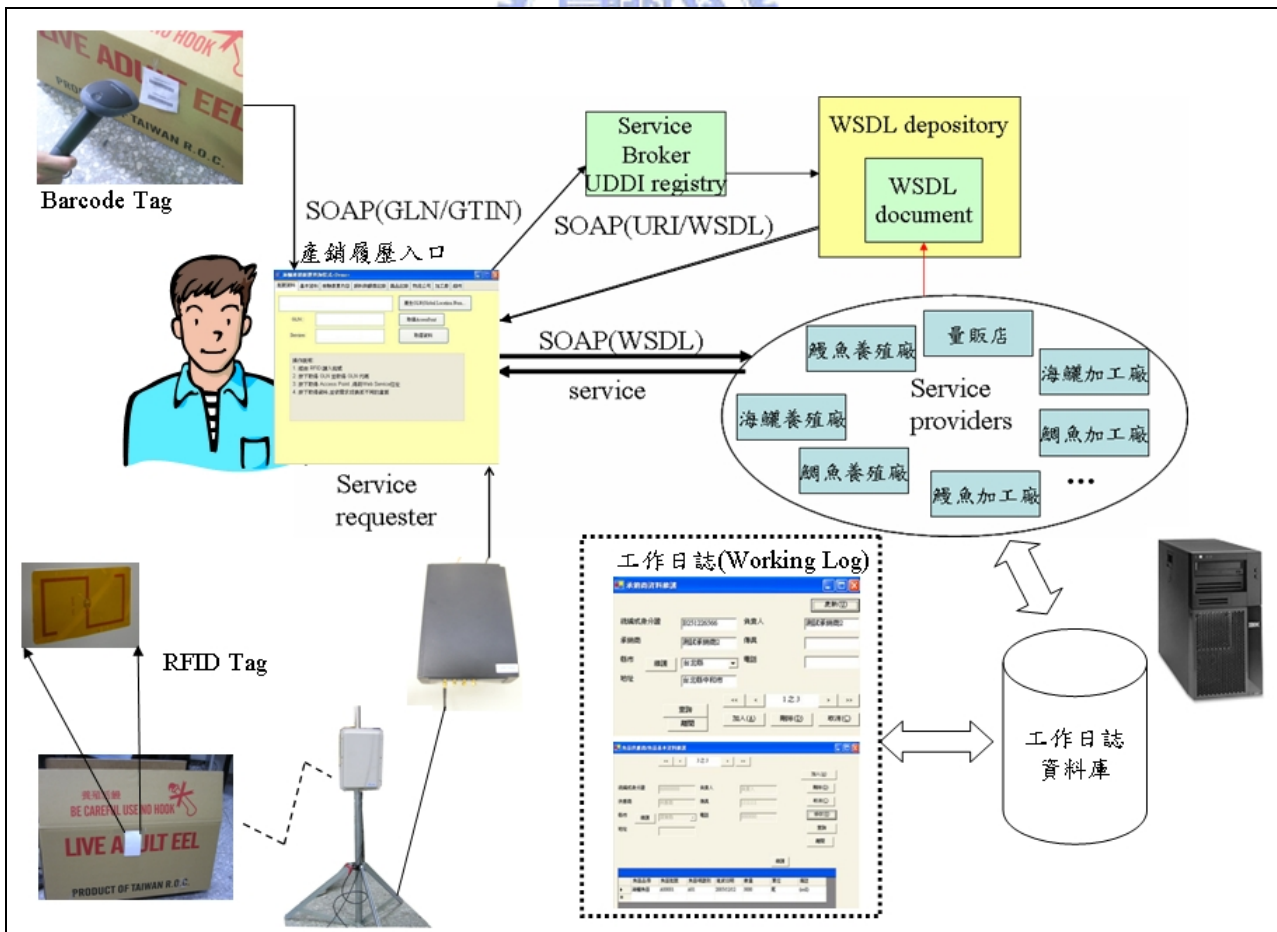


圖 1.1 網路服務技術與產銷履歷整合服務架構圖

工作日誌使用美國國家標準 IDEF0 及 IDEF1X 來建立。網路服務可連接工作日誌等子系統而形成一個完整的分散式產銷履歷系統如圖 1.2 所示，而使用者可進行產銷履歷的追蹤與追溯。本研究將分為「工作日誌系統」、「產銷履歷追溯查詢系統」兩個角度來探討可能的解決方案。就建置步驟而言，工作日誌系統屬於「點」的建立；而產銷履歷追溯查詢系統屬於「網」的形成。

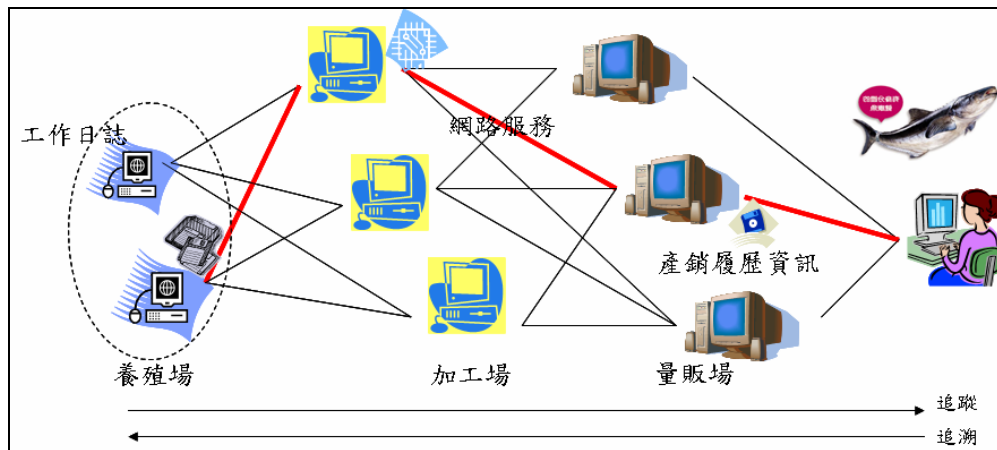


圖 1.2 網路服務連接工作日誌而形成的分散式產銷履歷系統[23]

工作日誌系統主要目的是有效率儲存業者在產銷流程的相關紀錄，對內必須作為業者接收存放生產流程的相關資訊，對外則作為網路服務技術的資料提供者，提供服務需求端的生產履歷查詢請求。因此先透過 IDEF0 規格將流程予以標準化之後，再將資訊流部份用 IDEF1X 技術進行三階正規化，接著將正規化的 IDEF1X 圖用來建立工作日誌。

產銷履歷追溯查詢系統主要目的是利用網路服務的技術，提供服務需求者相關的資料服務，網路服務技術架構主要分成服務需求端、服務伺服器端與服務仲介端三者。服務的內涵則是讀取工作日誌資料庫系統，而對於紀錄內容的開放程度則由生產業者自行在網路服務描述語言中定義。如此，生產業者不用擔心生產技術的資料安全問題，而供應鏈上各端點的業者能透過網路服務技術將生產履歷相關資訊加以串聯，提供消費者充分的產品產銷流程相關資訊。藉由這兩個子系統的建構，並透過適當的組合，期望能擘畫一套完善且高效率的海鱺供應鏈產銷履歷系統。

## 1.4 研究方法

本論文的研究步驟一共分為八個步驟，分述如下：

### 1. 確認研究目的與範疇

本論文首要步驟是確認供應鏈上各端點資訊處理的重要性，並且了解現行的系統架構是否具有改善的空間，而後衡量個人力量以及實際需求，規劃未來的研究目的。

### 2. 相關基本技術研究

確認研究目的之後，針對所有可能需要的知識與技術進行了解，一方面選擇較佳的工具，另一方面也再次確認系統建構的可行性。

### 3. 系統雛型實作

確認過系統建置的可行性之後，本研究嘗試完成一簡易的網路服務技術架構雛形系統。該雛型系統有兩項目的：首先是了解系統實際運作狀況的效能，評估是否能夠符合需求；第二個目的則是以此系統雛型作為往後與配合廠商與政府單位溝通協調的工具，以便讓合作單位對於未來新系統有所概念，同時也協助訪談過程中雙方的溝通。

### 4. 廠商訪談

為了讓理論研究與實作不至於脫軌，本研究時地訪問了澎湖海鱸養殖業者、台南嘉義地區的鰻魚與台灣鯛養殖業者、盤商、桃園地區的加工業者等，詳細了解並記錄供應鏈中各端點的業者對於生產履歷的使用狀況以及面臨的困難。

### 5. 作業流程分析

本研究根據廠商以及漁業署所提供的資料與需求規格，以 IDEF0 規格繪製系統作業流程圖，明確規劃出工作日誌系統的各種紀錄項目。

### 6. 工作日誌資料庫與應用程式實作

根據步驟五所繪製的 IDEF0 規格，將其轉換為 IDEF1X 規格，並利用此 IDEF1X 規格產生資料庫綱要。同時開發工作日誌應用程式，用來操作此資料庫。

### 7. 進階資訊技術研究與實作

根據所學習的技術，開始建構履歷追溯查詢系統。在網路服務技術架構之下，透過網路服務描述語言(WSDL)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(SOAP)加以包裝後在網路上傳輸；而仲介端則利用通用描述、發現與整合(UDDI)企業註冊(UBR)主機登錄。

### 8. 結論與建議

系統進行上線測試之後，本研究記錄分析其表現，並製成文字報告，以供往後的系統開發人員作為參考之用。

圖 1.3 為本論文的研究方法架構。

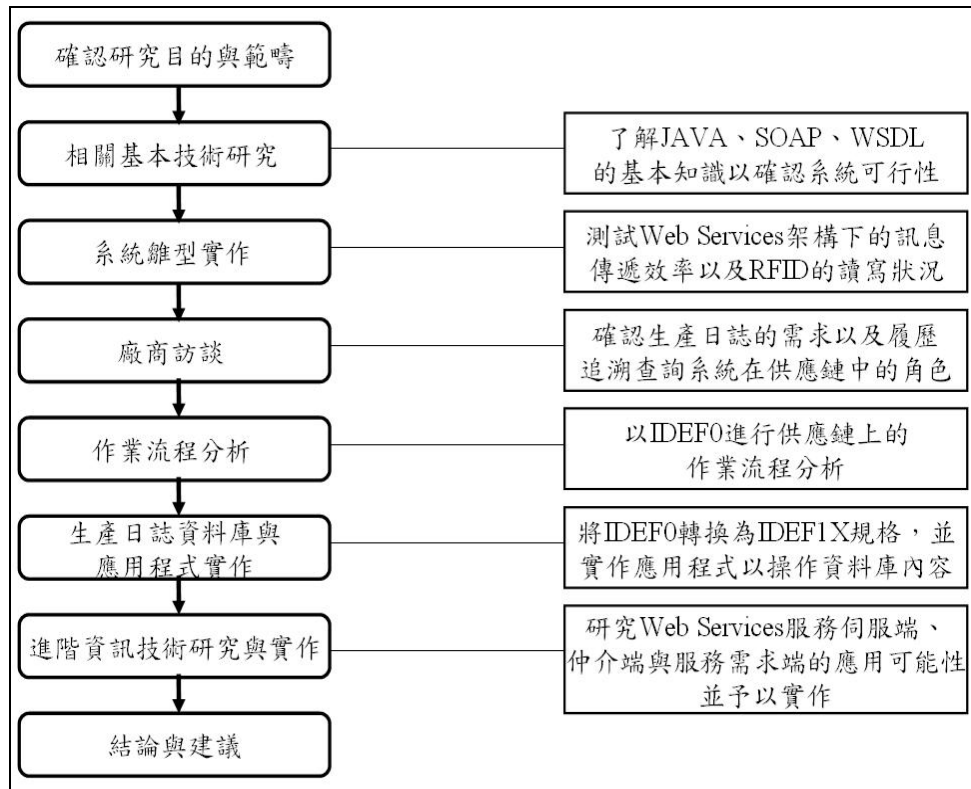


圖 1.3 研究方法架構





## 1.5 論文架構

本論文的內容包含七個章節，內容大綱如下。論文架構如圖 1.4 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：『工作日誌系統設計』，從海鱸生產供應鏈的作業流程中擷取需求資訊，分析物流的資訊內容與需求作為工作日誌資料庫系統的建構基礎。並介紹工作日誌系統的操作程序。

第四章：『履歷追溯查詢系統架構—伺服器端設計』，針對服務伺服器端程式使用 Axis 套件，提出兩種建置方法。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：『履歷追溯查詢系統架構—服務需求端與仲介端設計』，分為服務需求端與仲介端兩項。對服務需求端而言，介紹履歷追溯查詢系統的服務需求端介面設計與操作說明；而對仲介端而言，說明仲介端的操作使用以及文件規格。

第六章：『混合式產銷履歷追蹤查詢系統的操作』，介紹射頻識別辨識技術原理與編碼規則，利用射頻識別辨識技術與履歷追溯查詢系統的服務需求端予以整合，並介紹其操作說明與相關應用技術。

第七章：『結論』，結論與未來研究方向。

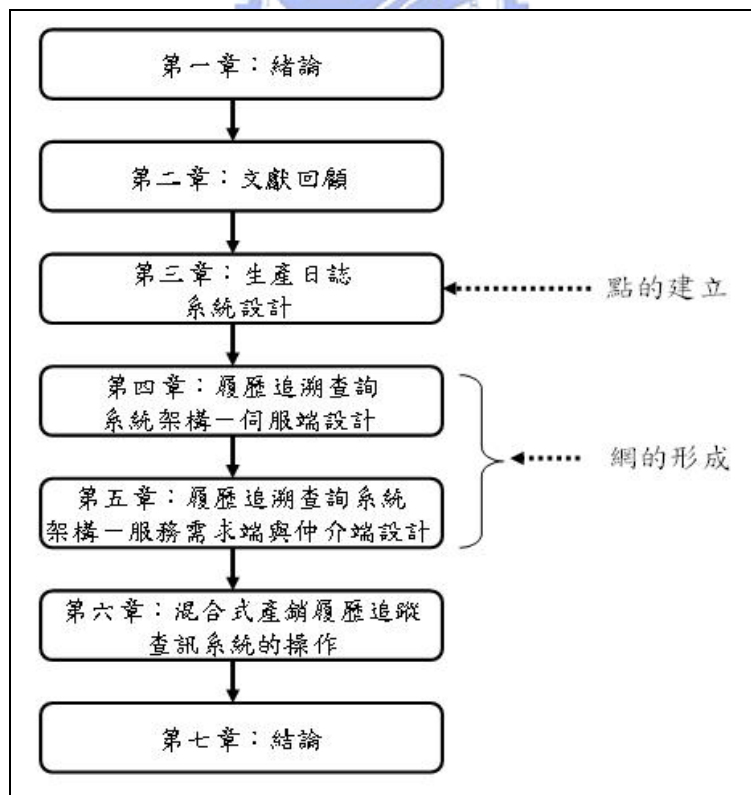


圖 1.4 論文架構圖



## 第二章 文獻回顧

本章分為兩個部份，第一部份說明產銷履歷相關的制度與流程，內容為 2.1 節「產銷履歷的流程」。第二部份則為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，內容為 2.2 節「射頻識別技術」，對於射頻識別技術從標籤、讀寫器以及後端資訊系統架構進行規格介紹。2.3 節「網路服務技術」說明服務導向架構，並逐一說明網路服務技術中使用到的細節，包括簡單物件存取協定、網路服務描述語言、通用描述、發現與整合以及後續的商業流程描述語言等技術規格。2.4 節「射頻識別與網路服務技術的分析結論」針對射頻識別與網路服務技術說明與整合。

### 2.1 產銷履歷的流程

本節說明產銷履歷的相關制度與流程。2.1.1 節說明食品安全的相關制度措施，包含危害管制點分析、ISO22000 系列等制度。2.1.2 節說明養殖水產品的供應鏈流程以及供應鏈的產銷資訊追蹤。2.1.3 節說明產銷履歷與資訊技術的關係。

#### 2.1.1 食品安全相關制度說明

危害分析管制點(Hazard Analysis and Critical Control Point, HACCP)[4]是世界公認食品安全管制的最高標準，源自於美國太空總署為了提供太空人完全安全的飲食，與食品企業 PILLSBURY 所共同開發的汙染風險排除系統以及確保食物安全的管理制度。用於判定、控制加工過程中所有可能出現危害污染，並加以預先防制。危害分析管制點系統從最初的三個原理，危害識別、確定關鍵控制點和控制任何危害、建立監視系統，發展到目前的七項原理和四個步驟，這七項原理是：對危害進行分析、確定關鍵控制點、建立關鍵值域、建立關鍵控制點的監控系統、對於監控系統顯示出某個關鍵控制點失控時必須建立應對措施、建立驗證程序以確認 HACCP 系統運行的有效性，建立文件化的流程與紀錄。而對於危害分析管制點的實施而言，四個步驟分別是：準備與規劃、HACCP 的研究與規劃發展、將 HACCP 規劃予以實作、對於 HACCP 系統的驗證與維護。將 HACCP 擁有的意義使用於一般食品衛生標準，以確保食品的安全性，是先進國家重視食品安全的重要方法。

從上述七項原理與四項步驟[48]可知，HACCP 著重的是對於可能發生問題的環節實施重點管理，因此在實施之前就應該先決定重要管制點，並進行正確記錄。就食品安全的觀點而言，HACCP 已臻完善；但是針對可追蹤性的觀點而言，在美國主導的 HACCP 制度仍有落差存在。所謂食品可追溯制度(Food Traceability System)最早是由歐盟在 1997 年第二波狂牛症爆發之後，為了消除消費者對於日常飲食的疑慮，以恢復大眾信心的手段。歐盟在食品法中將「可追溯性」定義為：「追蹤、追溯調查食品、飼料、動物與動物相關物質的加工食品，在生產、加工、流通所有階段相關訊息的能力[3][10]。」而日本農林水產省將食品追溯制度定義為：「可追溯、追蹤食品在生產、加工處理、流通、販賣等各階段的資訊。」可追溯性重視的不只是食品的衛生標準，還必須追溯其原料，例如飼料組成成分、農藥使用狀況；追查其產銷過程，以確認安全性。以牛肉為例，牛隻個體以何種方式飼養、使用何種飼料、健康狀況、施打的藥品等資訊，公開給一般消費大眾。

就農、漁產品而言，可追溯性就是追溯農產品生產與流通履歷的過程，也就是在農漁產品生產、加工處理與流通、販售過程的各個階段，由該階段的業者分別將食品產銷履歷流程等相關資訊予以記錄、保存並公開標示，讓消費者可以透過追溯食品產銷相關流程，了解在製程各階段的重要資訊。藉由食品流通鏈的可追溯系統，可追溯到產品本身特性，也可以了解產品產銷過程，包括生產者、生產地點、原料、加工製造、流通、販售等。一旦產品發生問題，能精確迅速地追溯到源頭，找出事故原因，將傷害降至最低。

而為了滿足對企業 HACCP 體系的建立和產品履歷可追蹤性的需求，國際標準化組織 (ISO) TC34 成立工作組研究制定了食品安全管理體系標準 ISO22000。ISO22000 標準不是對企業提出食品安全的最低要求，也不是食品生產法規的通常要求。該標準是組織自願遵循的管理要求，為食品鏈中的任何企業提供了一個重點更突出、連貫一致和綜合完整的食品安全管理體系。ISO22000 標準的條款編制形式與 ISO9001:2000 相同，為食品產業提供了系統化的食品安全管理體系框架，該標準整合了 HACCP 原理，可用於食品供應鏈的各階層組織，從飼料生產者、初級生產者、到食品製造者、運輸和倉儲經營者，直至零售分包商和餐飲經營者以及與其相關聯的組織（如設備、清潔劑和添加劑等生產者），其目的是將企業終產品交付到下一階段時，將其中已確定的危害控制和降低到最低水準。

而 ISO22000 其中的附屬文件 ISO22005 將規範食品可追溯至度相關內涵，預計將於 2006 年制定完成。

### 2.1.2 養殖水產品供應鏈流程

對養殖水產品而言，產銷履歷資訊就是追蹤養殖漁產品從養殖(生產)到販運以及銷售過程的相關資訊。也就是從養殖漁產品料供應(例如飼料與藥品)來源、生產(魚苗放養與養成)、運銷(流通業者)、加工(例如包裝廠與加工廠)，一直到銷售的過程中，每一階段的資訊都可以向上游或下游查詢，以瞭解養殖漁產品的來源、養殖場、加工廠、販售點之間的紀錄與資訊，如圖 2.1 養殖漁產品產銷供應鏈所示。

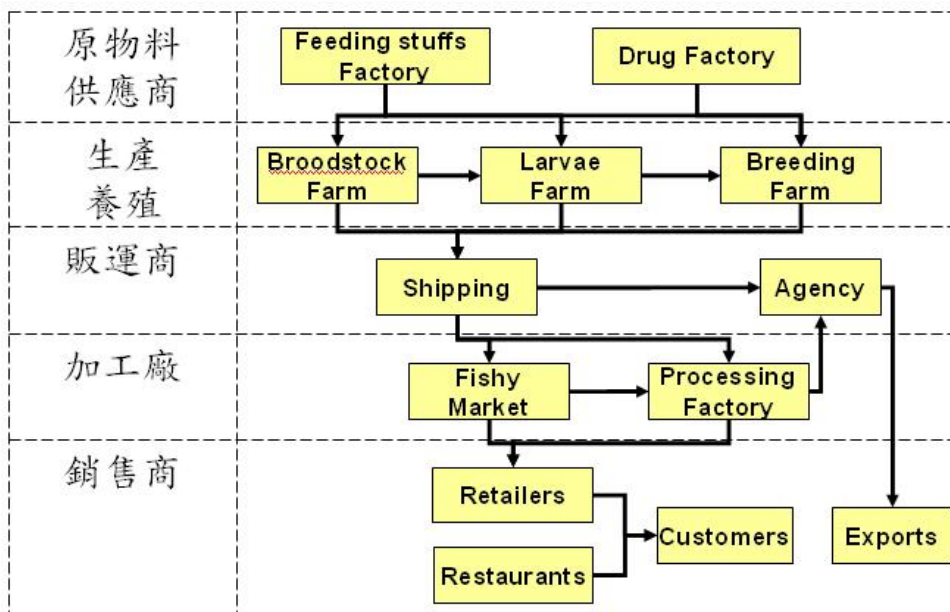


圖 2.1 養殖漁產品產銷供應鏈架構圖[3]

對於養殖漁產品供應鏈的產銷紀錄而言，以下分別針對原料廠商到養殖場、養殖場到運輸業者、運輸業者到加工廠、加工廠到消費者之間的產銷履歷追溯模式進行說明。

在原料廠商方面，主要以飼料與藥品兩項產品為主，在飼料廠方面，對於每批飼料的生產會有其對應的加工生產批號，並記錄包括生產日期、生產批號、飼料來源與配方等紀錄，同時應該有相對應的飼料檢驗報告已確定該批飼料金屬或藥物殘留狀況。而藥品廠商方面，對於藥品生產也必須有對應的生產批號，記錄生產日期、生產批號、原料以及配方比例等資訊。當養殖場向飼料商或藥品商採購時，飼料商或藥品商端會產生出貨紀錄資料，而養殖場端則登錄進貨相關資料，同時藉由追溯生產批號與生產廠商，透過產銷履歷資訊系統得到飼料或藥品的來源、成分與檢驗報告等資訊。圖 2.2 以飼料廠到養殖場的追溯模式為例說明其架構流程。

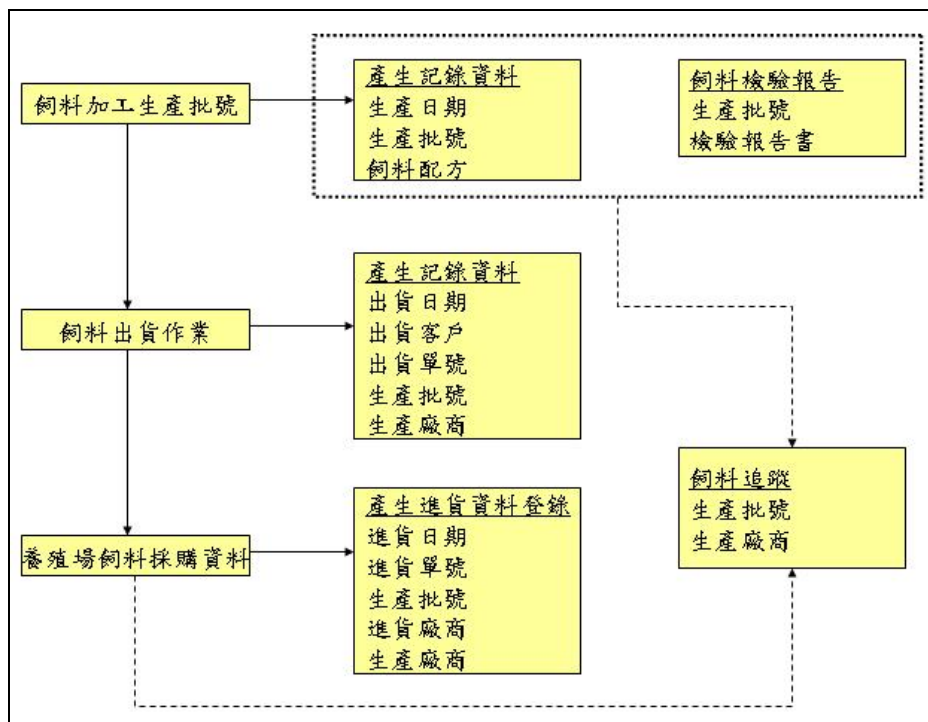


圖 2.2 飼料廠到養殖場的資訊追溯模式[3]

養殖紀錄在養殖場和物流業者之間資訊連結的應用，如圖 2.3 所示。在養殖場端，紀錄資訊包括養殖場的工作日誌紀錄，例如水質與養殖環境資料、環境與藥品檢驗報告、餵食資料等，養殖場在出貨時會記錄運送魚種名稱、規格與數量、出貨日期、出貨單號及養殖批號等資訊。另一方面，在物流運輸端則會記錄該批漁產品的所有廠商、養殖批號、出貨批號、運送出發時間等資訊，並能與養殖場的產銷履歷資訊互相連結。



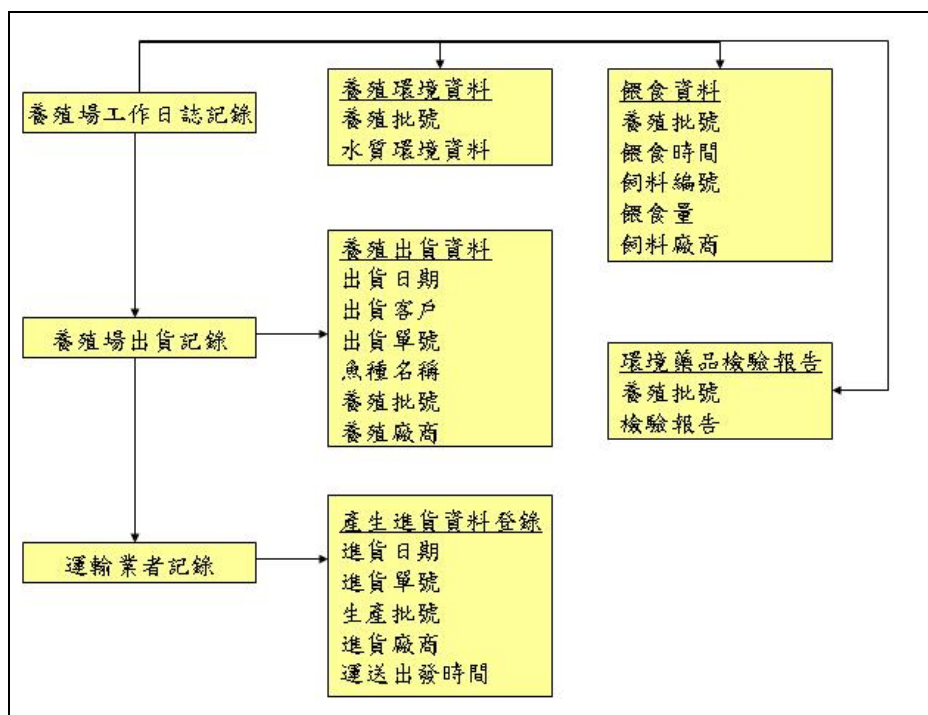


圖 2.3 養殖場到物流運輸的資訊追溯模式[3]

養殖漁產品經由運輸業者的物流配送系統之後，在物流系統端會記錄運輸環境的相關資訊例如：環境溫度、溼度、運輸工具與運輸批號等。並且在物流經過的各物流站記錄運輸過程的資料如出發縣市、經過物流站、運輸人員、作業人員等。另一方面，在加工廠端則會記錄漁產品的來源、進貨廠商、進貨批號、進貨日期、運送抵達時間等資訊，並能夠與物流業者的出貨資訊系統相連結。關於物流運輸端點到加工廠的資訊追溯架構如圖 2.4 所示。

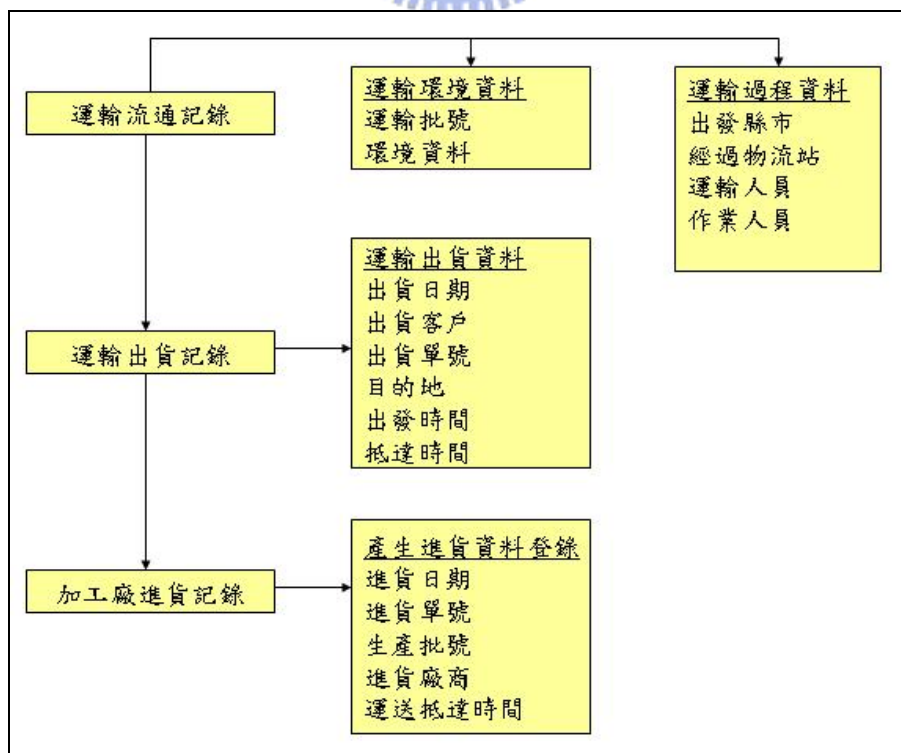


圖 2.4 物流運輸到加工廠的資訊追溯模式[3]

漁產品經由加工廠的加工與包裝處理之後，在加工廠端會記錄來源、養殖戶名稱、加工批號等資訊；在銷售端(消費者端)方面，透過銷售端點的進貨批號作為識別碼，可以進而追溯養殖漁產品的相關資訊如圖 2.5 所示。例如養殖過程中的水質環境資料、環境與藥品檢驗報告與養殖過程等相關資訊，消費者透過漁產品產銷履歷的整合可以了解漁產品的所有產銷流程詳細資訊。

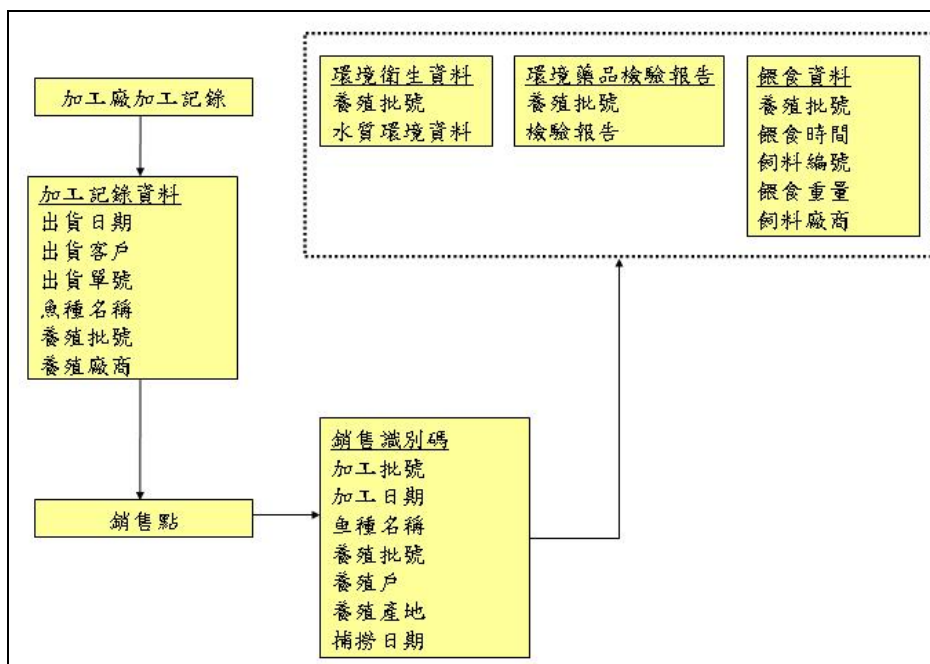


圖 2.5 加工廠到銷售端點(消費者)的資訊追蹤模式[3]



### 2.1.3 產銷履歷與資訊技術的關係

關於產銷履歷的運作方式，主要分為兩大類[2]。第一種是使用射頻識別技術，當製造商生產產品時，會將專屬的電子標籤貼在產品上。由於電子標籤可以讀寫資料，每經過一次加工、運送、組裝與販售的時候，便會將相關資訊寫入電子標籤。如此一來，該電子標籤完整記錄所黏貼產品的產銷履歷，也可以將該電子標籤視為專屬於所附著產品的電子化產銷履歷，如圖 2.6 所示。關於射頻識別相關技術將在 2.2 節中說明。

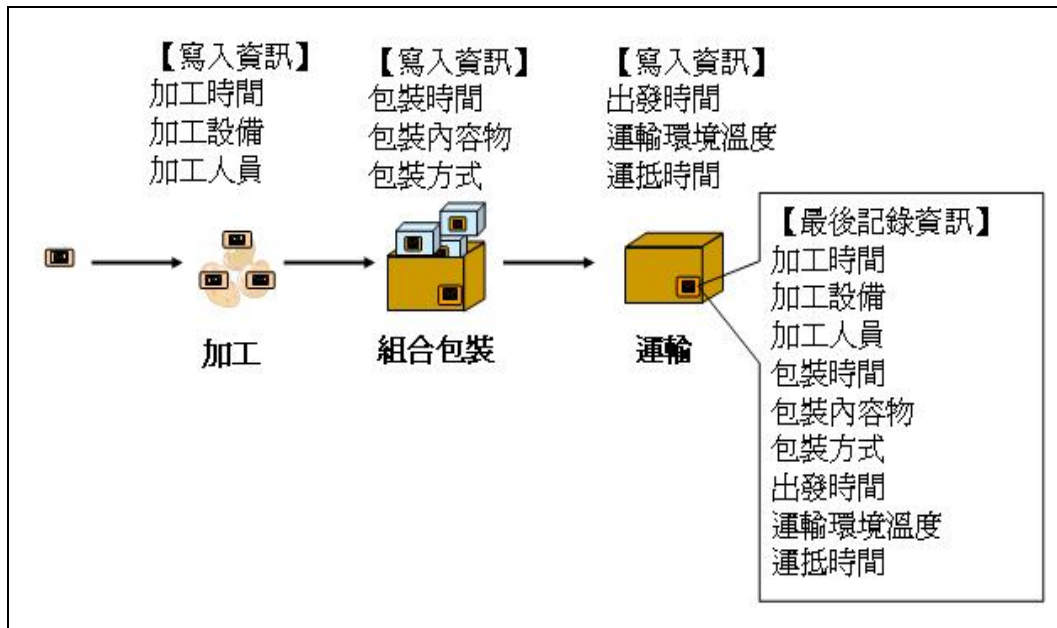


圖 2.6 射頻識別技術的產銷履歷模式

射頻識別技術目前已經逐漸在各行業測試導入，但是低單價的產品使用射頻識別技術所造成的成本太高，圖 2.7 顯示目前射頻識別技術與傳統條碼的成本差距。有鑑於成本問題，因此第二種方式採用已發展相當成熟的條碼標籤作為編碼載具，此種方式的運作成本可以得到大幅度的降低，但是由於條碼的記憶容量最大極限只有 48 個字元(EAN/UCC-128)與不能重覆寫入資料等重大缺陷，因此必須倚賴網路服務技術予以彌補。此種運作方法，條碼上只有記錄一組唯一的識別碼，此識別碼提供認證的功能用以追蹤供應鏈中該端點的產銷狀況；而該端點的產銷過程資訊則記錄在後端的資訊系統中，利用網路服務技術予以串接。換句話說，在這種模式下的產銷資訊，是利用網路服務的技術而跟著廠商移動，圖 2.8 說明此架構運作模式。而關於網路服務相關的技術將在 2.3 節中說明。

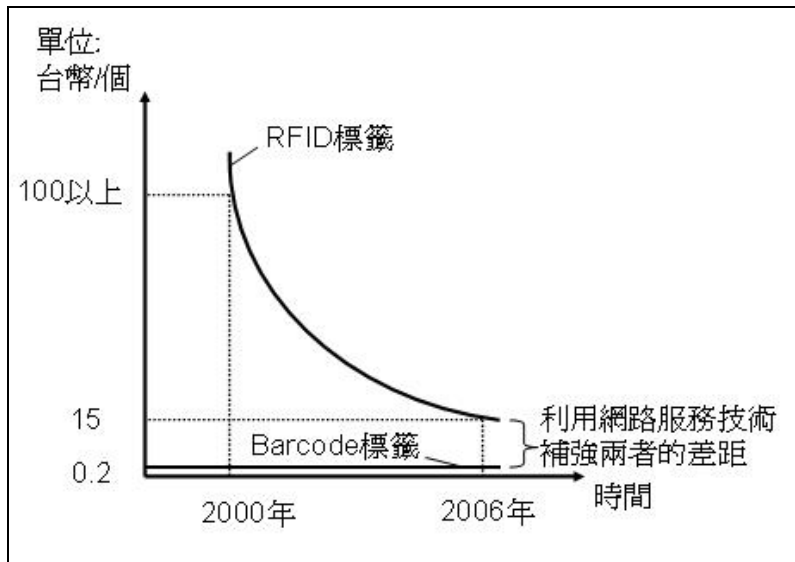


圖 2.7 射頻識別技術與條碼技術的成本比較

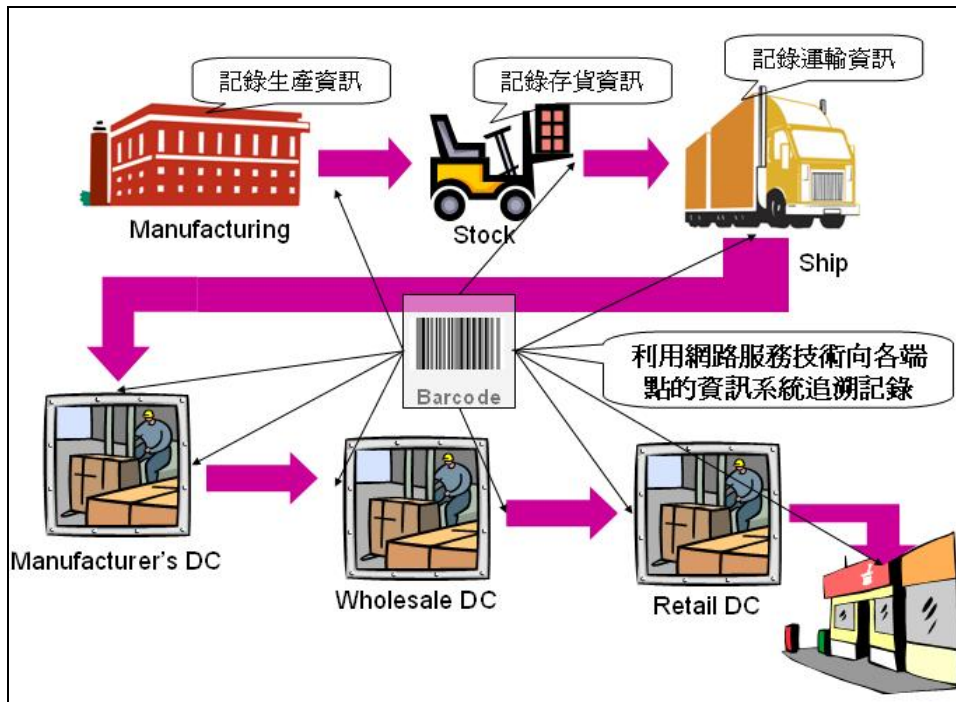


圖 2.8 條碼與網路服務技術並行的產銷履歷模式

## 2.2 射頻識別技術

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與電子化產品編碼技術、射頻識別讀寫器、中介軟體架構等元件，目前射頻識別規格標準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)，電子化產品碼是以麻省理工學院創立的 Auto ID Center 所研發的應用技術為基礎，由 EAN.UCC 所合資成立的 EPCGlobal Inc. 負責後續研究發展；國際標準組織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。而對射頻識別整合服務而言，是指如何將上述產品與原有資訊系統加以整合以提供加值服務。2.2.1 節為「EPC Network 架構」；2.2.2 節為「射頻識別標籤、標牌與編碼」；2.2.3 節為「射頻識別讀寫器」；2.2.4 節為「中介軟體架構與資訊服務系統」；2.2.5 節為「射頻識別技術的困境」。

### 2.2.1 EPC Network 架構

EPC Network[26]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，隨著物品的移動，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 2.9 說明 EPC Network 系統架構。

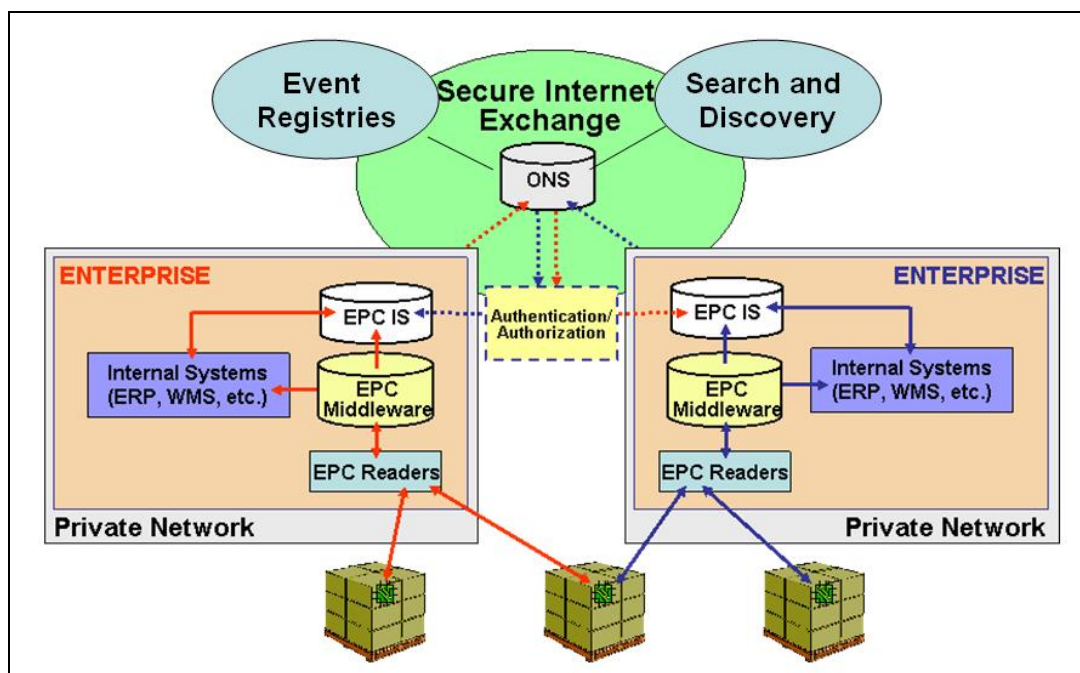


圖 2.9 EPC Network 架構圖[37]

EPC 的 RFID 網路架構主要由五個元素與三種層級所構成。五個元素分別為：電子化產品編碼、電子標籤與射頻識別讀寫器、EPC 中介軟體、EPC 資訊服務(EPC Information Services, EPCIS)與物件名稱解析服務(Object Name Service, ONS)。而三種層級分別是：實體與企業內部、企業夥伴之間的資料傳遞、其他應用服務層。對實體與企業內部層級而言，包含產品電子編碼、電子標籤、讀寫器、EPC 中介軟體。在 EPC Network 架構下，產品電子編碼儲存於電子標籤的記憶體之中，而電子標籤則附著於產品上。讀寫器則用於讀取或寫入電子標籤的產品電子編碼，將資料傳送到預定的目的地。而 EPC 中介軟體的功能是收集、過濾接收的數據資料，再進行後續個別應用。而對企業夥伴之間的資料傳遞層級而言，包含 EPC 資訊服務。每個需要與其他企業傳遞資訊或利用射頻識別蒐集資訊的企業，需要在企業內部放置 EPC 資訊服務的伺服器，EPC 資訊服務的內容透過中介軟體過濾收集，以便繼續與其他商業夥伴與企業內部系統交換資訊。對於其他應用服務層而言，包含物件名稱解析服務。物件名稱解析服務的功能在於連結製造商或相關的 EPC 資訊服務位置。目前全球物件名稱解析服務由 EPCGlobal 委託 VeriSign 營運，用以提供物件名稱解晰的連結、搜尋等服務。

## 2.2.2 射頻識別標籤、標牌與編碼

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。目前 RFID 標籤的標準規格有國際標準組織規範的 ISO/IEC 15963 以及 EPC 所區隔的六個等級的電子化標籤。

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，但是發行團體必須必須保證 ID 登錄編號為獨一無二的號碼。

EPC 將 RFID 標籤分為六個等級，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 2.1。

表 2.1 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

基礎編碼方式將 EPC 碼結構分為四區塊：標頭、一般管理者代碼、物件類別碼以及序號。標頭為 EPC 碼的第一部份，主要定義該 EPC 碼的長度、識別類型和該標籤的編碼結構。一般管理者代碼具有獨一無二的特性，為一個組織代號，也是公司代碼，並負責維護結構中最後兩組連續號碼。物件類別碼在 EPC 編碼結構的角色為辨識物件的形式以及類型，也具有獨一無二的特性。序號連續號也同樣具有單一的特性，賦予物件類別中物件的最後一層，使得同一種物件得以區分不同個體。



### 2.2.3 射頻識別讀寫器

對於 RFID 讀寫器規格而言，分為 EPCglobal 與國際標準組織(ISO/IEC 18000)系列兩種。EPCglobal 使用 EPC 碼、無線射頻識別技術與資訊網路等科技，建立 RFID 全球標準架構，為供應鏈自動化、追蹤與追溯管理增進資訊能見度，提供高效率 and 資訊準確性的物件資訊交換。秉持當初 Auto-ID「務必達成最低成本的支出」的宗旨，產生 EPC 規格的標籤和讀寫器。目前 EPC 所使用的頻率主要為高頻和超高頻。而 ISO/IEC 18000 系列涵蓋工廠的製程管理、流通運籌、回收等整個物流系統的國際標準。為了能夠對應物流系統各種不同的運用場景，而將通訊規格分為 135kHz 頻帶(ISO/IEC 18000-2)、13.56MHz 頻帶(ISO/IEC 18000-3)、2.45GHz 頻帶(ISO/IEC 18000-4)、5.8GHz 頻帶(ISO/IEC 18000-5)、860MHz~960MHz 頻帶(ISO/IEC 18000-6)、433MHz 頻帶(ISO/IEC 18000-7)六種頻帶。RFID 會因為使用不同的頻帶電波特性和顯現出不同的特性，而各種頻帶的特性各有優劣，必須針對使用領域與周邊環境對應使用。

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。在 EPC 標準中，無線電波由讀寫器發射，目的為讀取 RFID 標籤內的資料，EPC 標準規格的讀寫器可支援低頻與超高頻兩種頻帶，分別是 13.56MHz 和 860-930MHz 兩頻帶。目前 EPC 規格的讀寫器設計必須搭配被動式 EPC 規格標籤。讀寫器讀取範圍會隨頻率的高低呈現正向的變化，在低頻時讀寫器僅能辨識數十公分以內的物品；而超高頻辨識距離可達數公尺。

ISO/IEC 18000-2 由德國規格學會(Deutsches Institut Fur Normung, DIN)提案，以未含電池的被動式 RFID 為前提，使用 135kHz 以下的頻帶，通信最大範圍約數十公分左右。通常讀寫器發出的電波訊號會受到金屬或液體的影響，產生折射、繞射、反射等物理現象，使電波訊號受到干擾而衰減。而對電波而言，越長的電波波長能夠對障礙物產生轉折或是穿透的特性。135kHz 頻帶的波長通常在兩千公尺以上，因此其最大特性是不容易受到周圍金屬與液體的訊號干擾，且沒有天線方向性的限制。目前 135kHz 常應用的案例大都為容易遭遇水分或金屬的環境，例如日本迴轉壽司店的盤子底下、工廠的生產管理、汽車防盜識別裝置、金屬機殼的回收作業等。ISO/IEC 18000-2 的主要缺點是天線元件的長度很難縮短以及容易受到雜訊干擾。由於電波波長的特性，造成讀寫器的天線元件必須加長才能夠順利讀寫訊號。在雜訊干擾方面，由於主要採用 ASK 調變電路，會視資料而將載波的振幅產生調變，若載波的雜訊太大會造成辨識資料與雜訊的困難。對於機器內部的馬達或電源開關等雜訊源環境，通常都必須使用屏障物隔開雜訊源。

ISO/IEC 18000-3 分為荷蘭 Philips 公司與美國德州儀器公司所提出的 Mode1 以及澳洲 Magellan Technology 公司所提出的 Mode 2。兩者均採用未含電池的被動式 RFID 為前提，使用 13.56MHz 的頻帶，通信最大範圍約六十公分左右。Mode1 與 Mode2 最顯著的差異，在於 Mode1 的通信速率為 26.48kbit/sec 或 26.69kbit/sec；而 Mode2 的通信方式採用八個通信頻道，因此最高通信速率可以達到 848kbit/sec，缺點是必須使用大型天線，而且為了包覆住上下左右的天線，只有隧道式讀寫器才能運作。而對於電波波長而言，13.56MHz 頻帶的波長只有廿公分左右，優點是天線只有約五圈左右的短線圈，可以對形狀任意加工。缺點則是比 ISO/IEC 18000-2 容易受到液體與金屬的干擾。在應用案例方面 Mode1 的應用較廣，例如門禁系統、物流系統、圖書館藏書管理等。而 Mode2 最適合的環境是大量且需要高速讀寫的航空行李託



運管理。

ISO/IEC 18000-4 分為美國 Intermec Technologies 公司所提出的 Mode1 以及德國西門子公司與荷蘭 Nedap 公司所提出的 Mode 2。Mode1 與 Mode2 最顯著的差異是 Mode1 採用被動式 RFID；而 Mode2 採用內建電池的半被動式 RFID，利用電池驅動 RFID，但是與讀寫器之間的資料傳輸運作仍然與被動式 RFID 相同。由電池提供的驅動電力可以使通信距離達到五至十公尺，且資料的傳輸速率達到 384kbit/sec。此模式適合使用在高速移動且通訊距離長的環境之下，唯一的缺點是內建電池的成本遠高於被動式 RFID。兩種模式均使用 2.45GHz 的頻帶，但是此頻帶也被微波爐、無線網路等機器使用，機器之間會產生訊號互相干擾的問題。對於電波而言，波長大約十二公分左右，因此直線前進性強，缺點是容易被各種障礙物遮蔽訊號，對於金屬或液體的穿透性很差。優點是因為波長短而可以使用較小型的天線，而讀寫器製作也可同步小型化。由於 ISO/IEC 18000-4 的長距離讀寫、小型化讀寫器與天線等特性，因此常用於貨櫃車的進出管理、汽車生產線等環境。

ISO/IEC 18000-5 是由挪威 Q-Free 公司提出，採用未內建電池的被動式 RFID，使用頻帶為 5.8GHz。因為和 ISO TC204 的交通用資訊控制系統對象重疊，在 2002 年 12 月的審議會議中遭到否決，目前此標準的審議已經中止。

ISO/IEC 18000-6 分為法國 Tagsys 公司與美國德州儀器公司所提出的 Type A 以及美國 Intermec Technologies 公司與荷蘭 Philips 公司所提出的 Type B。兩者均採用半被動或被動型 RFID，使用頻帶為 860MHz~960MHz。Type A 與 Type B 最顯著的差異在一次讀寫多個 RFID 標籤時的防止衝突技術。Type A 採用 Slotted Aloha 方式，防止衝突的成功機率較低，但是價格較低廉；Type B 採用 Binary Tree 方式，讀寫時間較長但是準確度較高，建置成本也因為準確度提升而增加。而對電波而言，此頻帶的波長約卅公分左右，其特性介於 13.56MHz 與 2.45GHz 之間。與 2.45GHz 頻帶相比，訊號較不容易被液體或金屬干擾；而天線的體積則比 2.45GHz 頻帶大數倍。目前常被使用的案例為美國的 Wal-Mart 與英國 Tesco 等大型流通業的倉儲貨品管理。

ISO/IEC 18000-7 由美國 Savi Technology 公司提案，以內建含電池的半被動式 RFID 為前提，使用 400kHz 的頻帶，通信最大範圍約數十公尺以上。此標準規格的特點是平常為了省電的目的，而將 RFID 標籤設定在休眠的狀態，只有通訊的時候供給電力。目前主要使用在美國軍方的物資與車輛的管理、海關貨櫃的電子封條等需要長距離通訊的環境。

## 2.2.4 中介軟體架構與資訊服務系統

對於中介軟體架構而言，國際標準組織並沒有針對此部分建立明確標準規範；而電子化產品編碼已經發展了 EPC 中介軟體架構。EPC 中介軟體架構是電子化產品編碼標準的重要元件，主要功能是資訊交換與處理。EPC 中介軟體架構是在網際網路的基礎上，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。

對Savant技術而言，專門處理巨量資料與過濾數據。在EPC系統中，讀寫器將收集到的EPC碼傳送給Savant，接著Savant向散落各處的ONS提出查詢請求，由ONS搜尋對應的產品資料位址，再回傳答覆給Savant，最後Savant藉此向相關單位的資料庫或資訊系統查得產品相關資訊。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。

PML是以XML為基礎所發展出新的網路通用標準，PML的目標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，可廣泛應用在存貨追蹤、自動處理事務、供應鏈管理、機器控制等方面。主要功用在相異資訊系統例如企業資源規劃或是製造執行系統中，提供可以交換資訊的共用介面。

對於物件名稱服務而言，因為RFID標籤的電子編碼需要對應到產品相關資訊，因此必須提供類似網域名稱解析服務(Domain Name Service, DNS)的物件名稱服務。物件名稱服務的根目錄是依照EPC標準，儲存註冊使用的RFID產品編碼，因此物件名稱服務可以得到全世界註冊使用的產品資訊。但是缺點是使用物件名稱服務查詢必須支付EPCGlobal訂閱費用。根據EPCGlobal評估，未來五年內每日的查詢次數將達到數十億次。

## 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[31]、干擾性、批量讀取精確度三項[2]。對價格而言，由於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。對IC晶片而言，其製造流程先以蝕刻技術裁切出大量的錐狀無線通信IC，澆注到液態中的薄膜底板上，使得底板與液體共振，讓重力將IC一起陷入至底板上的錐狀凹陷處；然後再將保護用的薄膜覆蓋上去，用鍍銀的方式印刷連接天線用的端子，最後再將端子和天線以異方性導電薄膜(Anisotropic Conductive Film)壓合。IC晶片是射頻識別標籤中最單純的部份，受限於晶圓製程設備以及相關物理特性的影響，晶片幾乎沒有大幅度降低成本的空間，藉由製程技術的改善，日本預估以「億」為生產批量的前提下，晶片價格將維持在5日圓左右。而對於天線而言，目前天線主要有兩種製造方法：蝕刻法與凸版印刷。蝕刻法因為使用金屬(通常是銅)，因此成本較高；而凸版印刷雖然可以讓材料成本降低到四分之一左右，但是由於天線尺寸在印刷時規格不穩定，而略微的差異便造成天線無法共振，而需要額外的切削費用；目前研發的技術目標朝向印刷與噴鍍，因為這兩種技術的成功才有大幅降低成本的希望。而包裝的型態對於價格有著大幅的影響，單純用透明塑膠片封住天線的鑲嵌式做法成本最低廉；但是其耐熱度與抗折、抗磨能力也最差。工廠自動化與物流管理所使用的射頻識別標籤，為了承受工廠內的髒污以及搬運過程的風雨損傷，均以較堅固的塑膠等材料進行封合，因此成本會上漲數十到數百倍之多。

對於干擾性而言，由於射頻識別技術採用電波通信，因此很容易遭到金屬或液體的干擾。對金屬干擾而言，將射頻識別標籤貼近金屬表面時，流過標籤的電流所產生的磁場，也同時在金屬上產生電流；這種電流方向是往抵銷射頻識別標籤磁場的方向所流動，造成標籤無法取得電力。而且射頻識別技術所使用的電波頻率越高就越難折射，高頻率的電波幾乎以直線方式前進，很容易因為金屬的干擾而造成無法讀取的狀況。而水分會吸收電波，造成許多含有液體的物品，使用射頻識別技術的時候會產生讀取失敗率提高或是有效讀取距離縮短。其中最容易遭到水分影響的是2.45GHz頻段的射頻識別標籤，此頻段的電波特性和幾乎完全被水所吸收。因此遇到液體的環境下，只有約50%吸收率的低頻率射頻識別頻段是勉強能採用的

方案。表 2.2 為工研院對於含水分讀取距離的測試，根據統計檢定，不同含水率的有效讀取距離有顯著的差異性存在。

表 2.2 含水分木棧板的有效讀取距離測試結果(單位：公尺)[1]

實驗次數	含水率 12.5% (低含水率)	含水率 18.5% (高含水率)
1	1.7	1.1
2	1.7	1.1
3	1.5	1.1
4	1.5	1.1
5	1.5	1.0

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兩項原因影響：射頻識別標籤與讀寫器的距離太遠與射頻識別標籤與讀寫器的方向不正確。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，造成其中某些物品的角度無法接收到讀寫器的電波以產生磁場，因此無法進行資料的讀寫交換。





## 2.3 網路服務技術

網路服務技術[17][6][45]是由Microsoft、IBM、Sun、HP的科技廠商聯合推動，是一項以網際網路為基礎，完整且模組的運作模式。此技術能自動被一個或多個應用程式所啟動，完成一個或一組特定的工作。網路服務的優點是提供簡單而順暢的機制，讓應用程式間不必經由人力操作介入，也不會受到雙方環境平台的架構，能自動地透過網際網路互相溝通。主要目的在異質的電腦整體架構之下，能夠穩定且迅速地使用網路傳遞訊息。此標準使用到的技術及應用分別在以下小節中介紹：2.3.1 節說明服務導向架構；2.3.2 節說明簡單物件存取協定；2.3.3 節說明網路服務描述語言；2.3.4 節說明通用描述、發現與整合；2.3.5 說明業務流程執行語言。

### 2.3.1 服務導向架構

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

服務導向的優點在於以服務的思維重新將所有現有的系統資源以服務型式看待，服務之間的溝通以訊息的形式來進行交換，以鬆散耦合的型態解決了以往在企業內部與企業之間的跨系統整合問題。透過元件服務化把分散式元件架構的限制去除，只留下呼叫端與伺服器端之間的使用協定以及呼叫與回傳參數。元件透過重新封裝後，以單元服務的型態存在，透過服務組裝形成更複雜的整合服務，或是更貼近商業流程的商業服務。以服務導向為基礎的系統分析與企業的經營管理分析十分接近，不但可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；而且又不受限於特定廠商的產品功能或是平台，達到真正的軟體開放性。

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，來源可以是區域網路或網際網路網路等。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。

服務導向架構的效益會在所有的資源逐步服務化之後顯現。當可供使用的服務達到一定的數量後，透過服務的重新組裝使新的應用系統開發時間大幅縮短，也能隨著需求彈性的調整與改變。目前服務導向架構最普遍的實作是網路服務技術，透過網路服務架構提供訊息處理的框架。網路服務架構的發展，可以分為三個階段：簡易型網路服務、組外型網路服務、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。而第二階段的組外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。

### 2.3.2 簡單物件存取協定

簡單物件存取協定(Simple Object Access Protocol, SOAP)[49]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。藉由劃分好的區塊模組來放置應用程式的呼叫程序，而在放置時必須遵守規定的編碼方式。其前身是XML-RPC(遠端程序呼叫)為基礎，「遠端」是指發出程序呼叫的位置，這也意味發出呼叫與接收呼叫執行程序的機器是不同的。但是當這兩端的機器是異質基礎平台，例如Windows、Mac、Unix等，平台之間的溝通會出現障礙。SOAP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。而SOAP傳遞的文字訊息，則使用XML文件[13][19]。這主要是著眼於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。[20]

目前版本是W3C組織在2003年六月公佈的1.2正式版本[39]，SOAP規範的內容包含SOAP Envelope、SOAP Encoding Style、SOAP RPC。SOAP Envelope定義了整體架構，在服務導向架構之下的Find、Publish、Binding等都必須使用SOAP訊息傳遞。而SOAP訊息本身就是一份XML文件，包含三個部份：SOAP Envelope、SOAP Header、SOAP Body，如圖 2.10 所示。



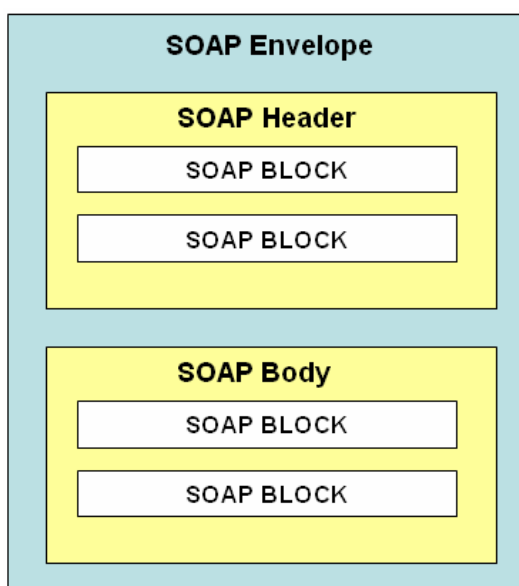


圖 2.10 SOAP Envelope 架構[23]

SOAP Envelope 是 SOAP 訊息必要的根節點，表示這份文件是 SOAP 訊息；Envelope 元素包含名稱空間(Namespace)的宣告與其他額外的屬性宣告。而 SOAP Header 為非必要的子元素，若使用 SOAP Header 必需定為第一項子元素，其主要功能是安排此文件經過特定節點，已取得附加的服務，同時交換所需的資訊。SOAP Header 內有四項屬性：encodingStyle、mustUnderstand、role、relay，其屬性值如表 2.3 所示。

表 2.3 SOAP Header 的屬性值[17]

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。

在 SOAP 訊息中 SOAP Body 是必要的子元素，功能等同一個承載資料的容器，其內容包含所要傳送的資料。所有在 Body 元素下的子元素都是資料項目，這些子元素的內容就是傳

送的訊息。至於資料應該放在 SOAP Header 亦或 SOAP Body，則必須視資料的使用者而定。

### 2.3.3 網路服務描述語言

網路服務描述語言目前使用的版本為 2001 年 3 月由 W3C 公佈的 1.1 版。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。網路服務描述語言的基本格式是以 XML 為基礎，這意味這是一份給機器或是程式閱讀的文件。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這種統一規格的說明書，其最大效益是大幅度降低網路服務應用程式開發的難度與時間。

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[17][24]。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 2.11 所示。

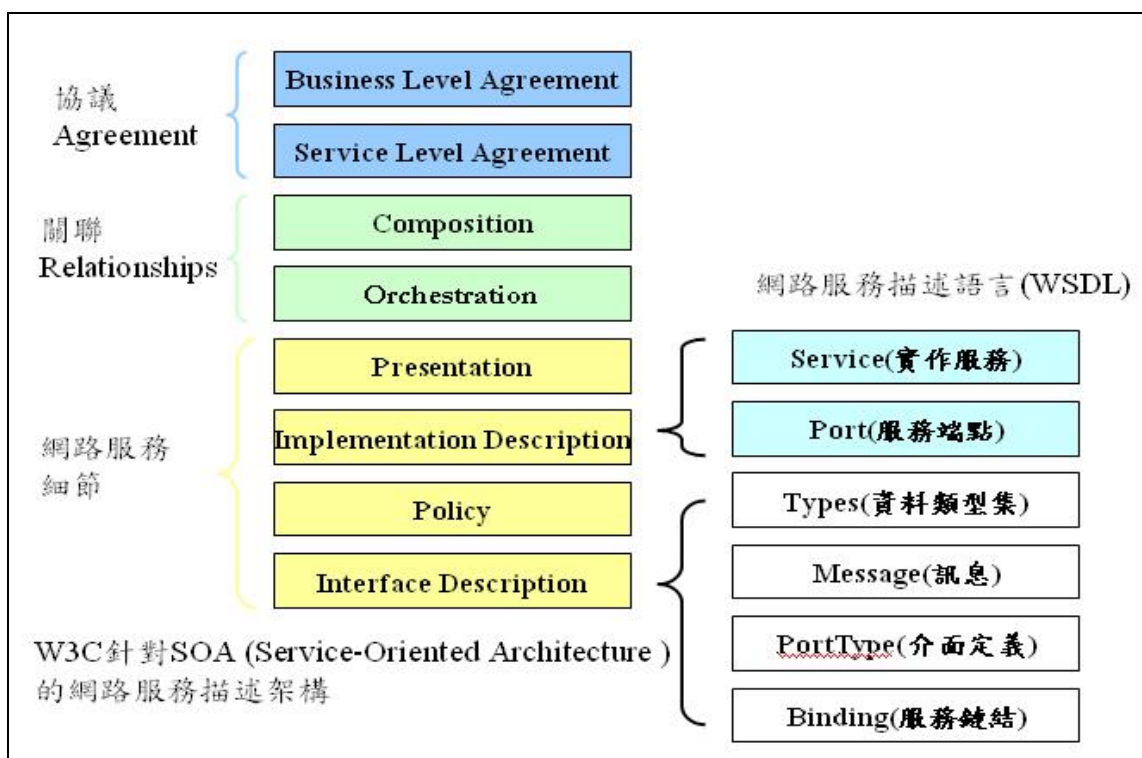


圖 2.11 網路服務描述架構與網路服務描述語言對應圖[17]

Port 指示服務所存在的網路位址；而 Service 主要用來描述所提供的服務，通常與 Port 並用，已表示開該 Service 所在位址。而 Types 用來存放資料交換的資料型態定義標準，例如 XML Schema 中對於資料型態的定義；Message 則用來定義服務需求者與提供者之間的互相溝通所使用的資料型態；而對 portType 而言，可視其為作業(operation)的集合，用抽象的方式書名網路服務的功用。portType 用於定義一連串傳送接收作業，包括資料輸入項與資料輸出項，因此常做為檢視網路服務描述語言文件的起點；Binding 區塊繼承由 portType 與 Message 所提

到的資料元素，繼續描述更具體的實作資訊，使用協定有 SOAP Binding、HTTP Get/Post Binding、MIME Binding 三種協定。

### 2.3.4 通用描述、發現與整合

通用描述、發現與整合(Universal Description, Discovery & Integration, UDDI)[17][24]是由 IBM、Microsoft、Ariba 等廠商於 2000 年第三季所發起的計畫。通用描述、發現與整合提供網路服務架構中註冊服務的角色。網路服務中除了註冊服務之外，還有服務提供者與服務需求者。對服務提供者來說，將服務內容以網路服務描述語言描述，並將其位址公佈於通用描述、發現與整合處，以供服務需求者尋找。而對服務需求者而言，可以透過通用描述、發現與整合提供的查詢功能找到滿足所需的服務位址，然後利用簡單物件存取協定，呼叫該項服務，其運作圖如 2.12 所示。

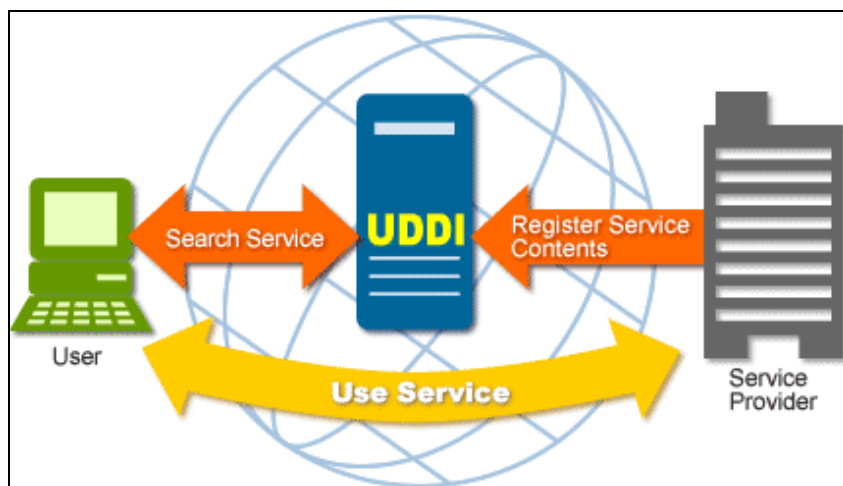


圖 2.12 通用描述、發現與整合在網路服務架構的角色[34]

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；而對「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；而對「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內容來說，可分為三大類：白頁(White pages)、黃頁(Yellow pages)與綠頁(Green pages)[17]。白頁用來登錄企業的名稱、服務項目、電話號碼及其他商業資訊等，讓其他使用者可以根據名稱搜尋企業；而黃頁將企業依照三種方式分類：美國政府所訂定的NAICS工業標準碼、聯合國UN/SPSC碼與地理區域，登錄相關的商業分類資訊；綠頁則定義呼叫介面的技術細節，讓企業之間知道應該使用何種格式作為交談的標準。

在通用描述、發現與整合註冊中較重要的資料內容有：Business Relationship、Business information、Service information、Binding information、Information about specifications for services，而這五項資料內容分別在描述、發現與整合文件中有相對應的節點來儲存。Business Relationship 在通用描述、發現與整合服務中對應的節點為 PublisherAssertion，由於有些服務

並非單純由單一業者提供，此時可藉由這項資料描述服務提供者之間的關係；Business information 在通用描述、發現與整合服務中對應的節點為 BusinessEntity object，包含關服務、分類、聯絡方式、URL、以及其他所需要與給予的服務互動的資料；而 Service information 對應的節點為 BusinessService object，用來描述網路服務的群組；Binding information 對應的節點則是 BindingTemplate object，內容為所需要叫用網路服務的技術細節；Information about specifications for services 對應的節點是 TModel object，內容包含有關網路服務所實作各種規格書的中繼資料。這些資料節點的對應關係如圖 2.13 所示。

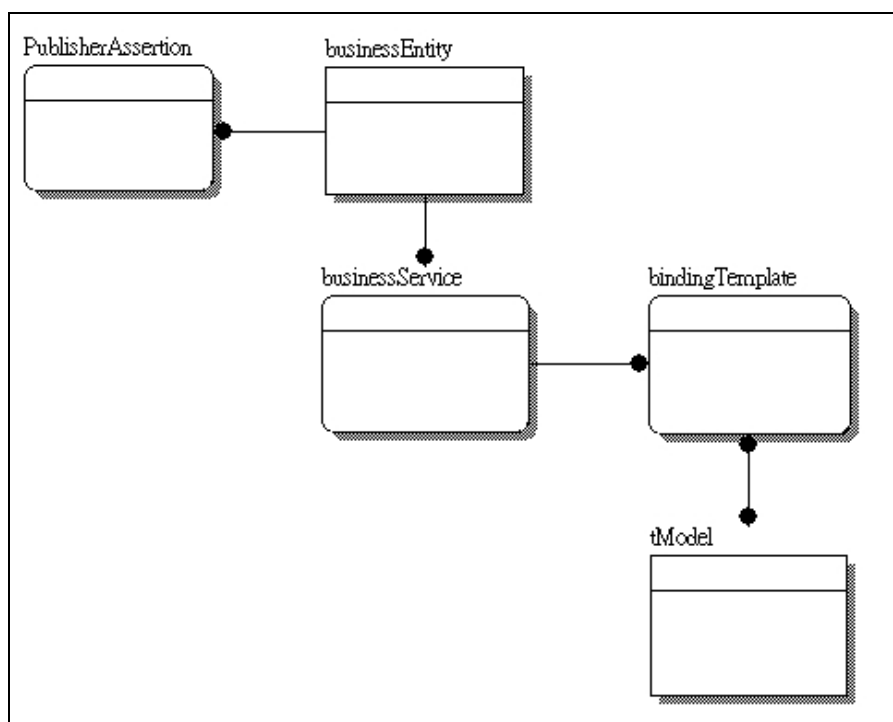


圖 2.13 重要資料節點的對應關係[45]

### 2.3.5 業務流程描述語言

在強調電子化的時代，以業務流程的調整與變動的彈性與否已成為影響企業成功的關鍵因素。因此以流程進行整合的業務流程管理(Business Process Management, BPM)是企業進行系統整合時的一種工具。業務流程管理是在商業邏輯設計層面而非技術實作層面上的功能，能夠歸納現有流程邏輯、設計新流程邏輯、佈署流程系統、自動執行流程工作任務、分析流程執行績效等涵蓋一連串流程生命週期的步驟。而業務流程管理與服務導向架構的中介橋樑就是網路服務技術，可以讓跨平台及跨語言的資訊系統整合根據網路服務標準加上一層介面，封裝成一個共用的服務元件讓其他應用程式呼叫以傳遞資訊。而其效益在於一方面能將企業內系統整合流程化以及業務流程電子化，也同時能將客戶與廠商的業務流程結合，以提高供應鏈中協同運作的程度。目前普遍被運用的實作技術以業務流程語言[9](Business Process Execution Language, BPEL)為主，業務流程語言是一種以 XML 來描述企業內部流程的語言，使原本建立在不同產品上的商業流程能夠以網路服務技術為基礎跨平台溝通，該標準也得到OASIS 標準組織的認證。

網路服務業務流程語言(Business Process Execution Language for Web Services, BPEL4WS)



是由BEA、IBM、Microsoft等公司於2002年提出，定位為網路服務技術的整合[47]。BPEL4WS能完成網路服務的調用、存取資料、拋出障礙或終止某一流程等工作，藉由對網路服務的串接或再編排，創造出更複雜的流程功能。BPEL4WS是利用流程圖表達，圖中每個步驟稱為一個活動(Activity)，而將網路服務的portType定義為每個活動的進入點，藉由進入點對於訊息的接收與傳送，達成BPEL流程的佈置與執行。

BPEL4WS把與流程互動的其他服務稱為夥伴(partner)[44]，每個夥伴都以網路服務的型態呈現，在互動過程中扮演不同的角色(role)。兩服務之間有夥伴連結(partner link)，定義關係中每個服務所扮演的角色並指定每個角色所提供的portType；而每個服務則可以包含多個portType。BPEL4WS主要可實現兩項功能：可執行的業務流程與不可執行的抽象流程。可執行的業務流程是指存在於系統模型中的企業體之間互動行為；而不可執行的抽象流程是指企業間協定，實作上是利用已存在的網路服務描述語言的portType組合成流程邏輯，定義出新的網路服務。利用BPEL4WS整合網路服務的架構圖如圖2.14所示。

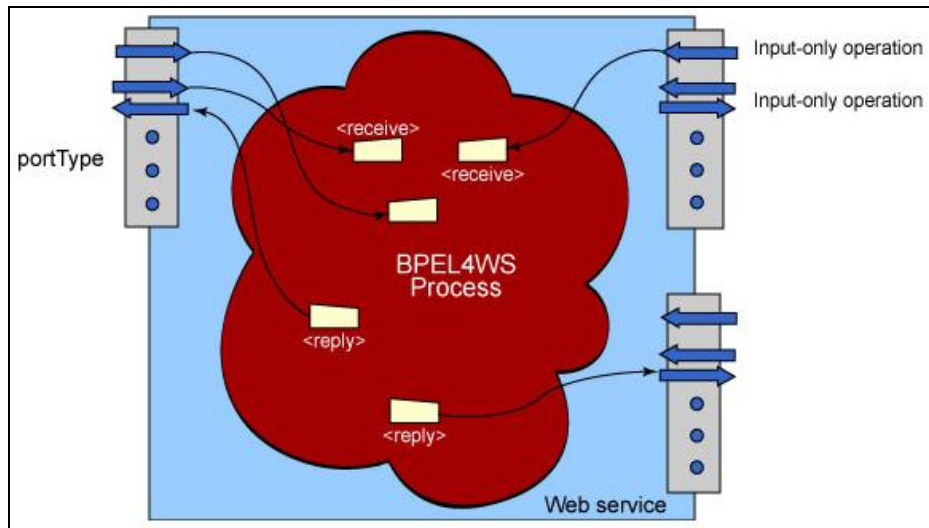


圖 2.14 BPEL4WS整合網路服務的架構圖[41]



## 2.4 射頻識別與網路服務技術的分析結論

射頻識別標籤與傳統條碼式標籤的比較如表 2.4 所示。最大的差異在於射頻識別標籤利用無線電波讀寫產品資訊，而且使用自動化讀寫設備同時執行多個標籤的內容辨識；不像傳統條碼式標籤需要人工操作，一次讀取一個標籤內容。射頻識別標籤的記憶內容相對條碼標籤而言非常龐大，因此產銷履歷的內容直接寫入射頻識別標籤，不需要再透過外部的資訊系統支援，提供相關的產銷資料。

表 2.4 射頻識別標籤與條碼式標籤比較表[21]

功能	條碼式標籤	射頻識別標籤
讀寫能力	唯讀	可讀可寫
方向性	必須近距離對準條碼	可遠距離讀寫，但有天線方向性的問題
同時讀取數量	一次讀取一個	可同時讀寫多個標籤
堅固性	易破損	耐天候與抗磨損
使用性	不可重複使用	可重複回收使用
安全性	無密碼保護	可加密
資料判讀	移動讀取速度與距離均受限制	可在較快的移動速度下讀寫

根據 2.1.3 節可知，雖然條碼式標籤比射頻識別標籤有諸多缺點，但是可以利用網路服務技術的分散式特性來彌補部份的缺陷。射頻識別標籤將完整的產銷履歷資料置入其內，跟隨所附著的產品移動與增修產銷資料；而條碼式標籤則將產銷資料分散於供應鏈各端點的廠商資訊系統中，利用網路服務技術串聯這些資料集。

有鑑於這項種模式的優點與缺失，本研究嘗試使用混合型的產銷履歷追溯模式：在資料載具使用射頻識別技術，擁有可重複多次讀寫以及遠距離的優勢；而產銷履歷資訊則記錄在供應鏈各端點的廠商日誌系統內，並使用網路服務技術將這些分散式的履歷資訊予以串聯。

## 第三章 工作日誌的設計

本章說明工作日誌的設計流程，3.1 節「海鱸供應鏈作業流程分析」利用 IDEF0[8][7][29] 分析海鱸生產作業流程的物流與資訊流；3.2 節「工作日誌系統三階正規化分析與設計」將 3.1 節的 IDEF0 圖轉換為 IDEF1X 規格[30]；3.3 節「工作日誌系統」則說明工作日誌系統的設計與操作；3.4 節「其他工作日誌設計」則說明養殖場以外的其他相關廠商工作日誌規格。

### 3.1 海鱸供應鏈作業流程分析

本節主要目的是從海鱸生產供應鏈的作業流程中擷取需求資訊，分析物流的資訊內容與需求作為工作日誌資料庫系統的建構基礎，共分為兩小節。3.1.1 節為「海鱸供應鏈作業 IDEF0 規格」；而 3.1.2 節將 3.1.1 節的 IDEF0 規格轉換為裴氏圖，分析其中物流與資訊流的動向。

#### 3.1.1 海鱸供應鏈作業 IDEF0 規格

經由對於海鱸養殖業者與供應鏈上下游的廠商勘查訪談，並結合專家與業者的意見建立海鱸供應鏈生產作業流程規格。由於 IDEF0 技術的優勢在流程圖形化的呈現清晰且容易瞭解，本節使用 IDEF0 表達海鱸供應鏈的生產作業流程與說明。

圖 3.1 為海鱸供應鏈生產作業流程 IDEF0 表達法。圖中方框代表海鱸養殖生產作業；方框左邊箭頭包括輸入海鱸養殖生產作業的物流與資訊流，分別有魚苗、飼料、藥品、添加物；方框右邊為輸出海鱸養殖生產作業的物流，也就是準備外銷的魚貨成品；方框上方箭頭表達控制整個海鱸養殖生產作業的市場需求；而方框下方箭頭為參與作業的人員與設備、網具、儲運容器。此為海鱸供應鏈生產作業流程的概觀圖。

**錯誤！物件無法用編輯功能變數代碼來建立。**

圖 3.1 海鱸供應鏈生產作業流程 IDEF0 表達法[23]

圖 3.2 為海鱸生產養殖作業主要流程項目 IDEF0 圖。圖中方框代表海鱸生產供應鏈依序進行的作業流程，分別從最初開始的魚苗與原物料進貨驗收作業，再經由原物料的儲藏作業後接著魚苗的養殖作業，最後到包裝出貨作業。方框作業下方的箭頭為參與海鱸供應鏈作業流程的人員與設備，包含飼料與藥品等原物料的驗收人員、倉管人員、養殖場餵食人員以及出貨包裝人員。方框左邊箭頭為各作業流程的輸入；方框右邊箭頭為各作業流程輸出。粗線為物流例如魚苗、飼料或是藥品；細線則是資訊流如飼料或藥品供應商進貨內容，其中包括飼料供應商、藥品供應商、飼料供應商代號、藥品供應商代號、數量、單位、成分描述等資訊。圖 3.2 中出現在資訊流的括號內文字描述表示資訊流內容的集合或是實體表單名稱；而無括號的文字描述表示現有的資訊流內容。例如從魚苗及原物料驗收到養殖作業，其中有括號的文字描述為魚苗進貨資料，就是下方魚苗供應商、供應商代碼、魚種、數量、單位的集合表單名稱。各作業更詳細的作業內容可由其往下一層展開圖來說明，例如圖 3.3 就是魚苗及原物料驗收作業的展開圖。

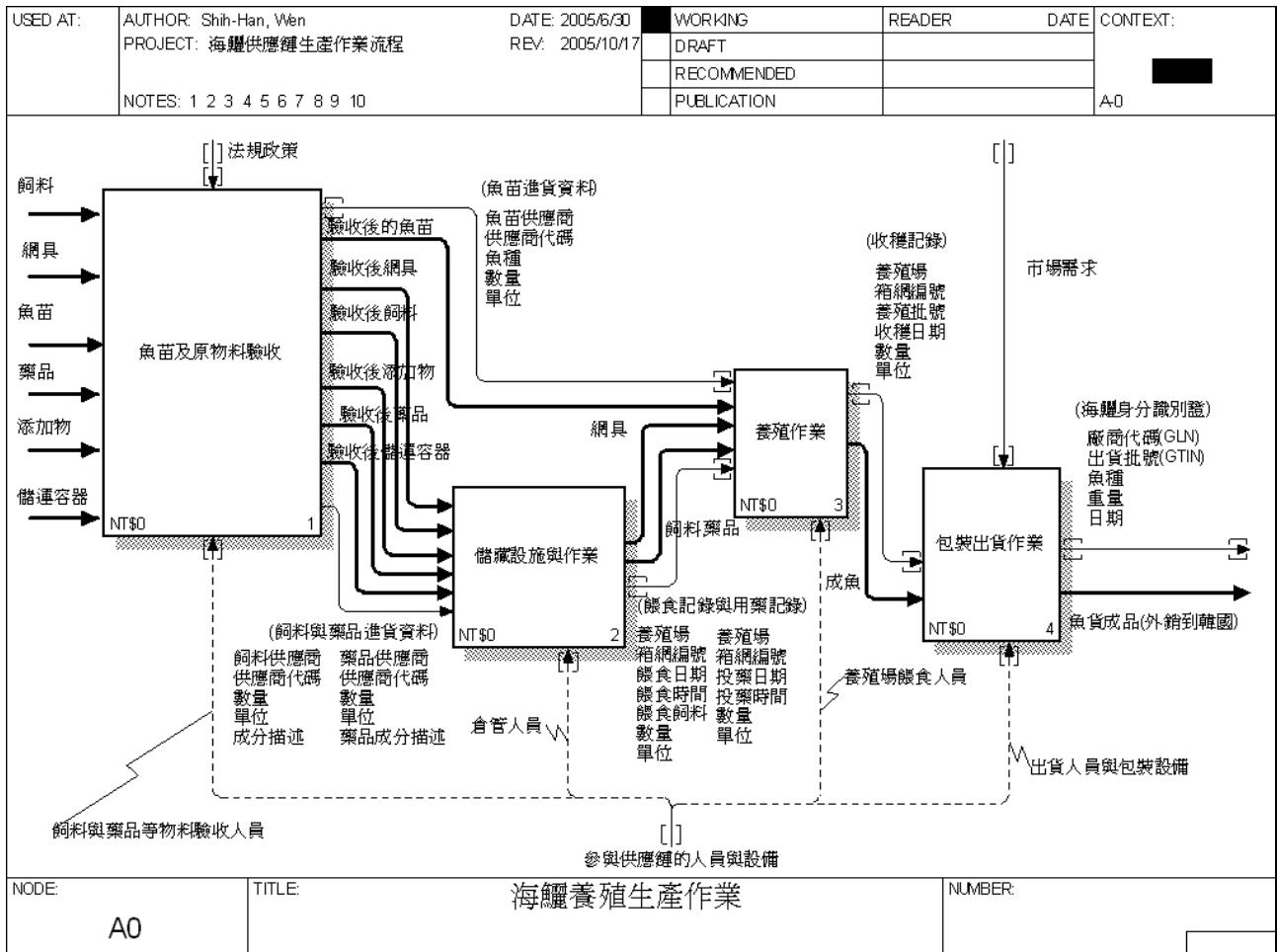


圖 3.2 海鱸供應鏈生產作業四大流程IDEF0 圖[23]

圖 3.3 為海鱸供應鏈魚苗及原物料驗收作業流程 IDEF0 圖，這是圖 3.2 中最左邊的魚苗及原物料驗收作業方框的展開圖，目的在說明魚苗及原物料驗收的細部工作。魚苗及原物料驗收作業可以細分為驗收魚苗、驗收網具、驗收飼料、驗收飼料添加物、驗收藥品以及驗收儲運容器六項工作。各項作業傳遞的物流分別從左方箭頭進入，經過各別的檢驗作業之後，再從右方箭頭輸出驗收完成的物流。各項方框作業中填滿灰色者是根據 HACCP 原則所制定的生產流程重要點(Critical Point)，驗收魚苗、驗收飼料、驗收飼料添加物以及驗收藥品四項作業是魚苗及原物料驗收作業流程中的重要點，也就是一定要在工作日誌資料庫系統中予以記錄的部份。

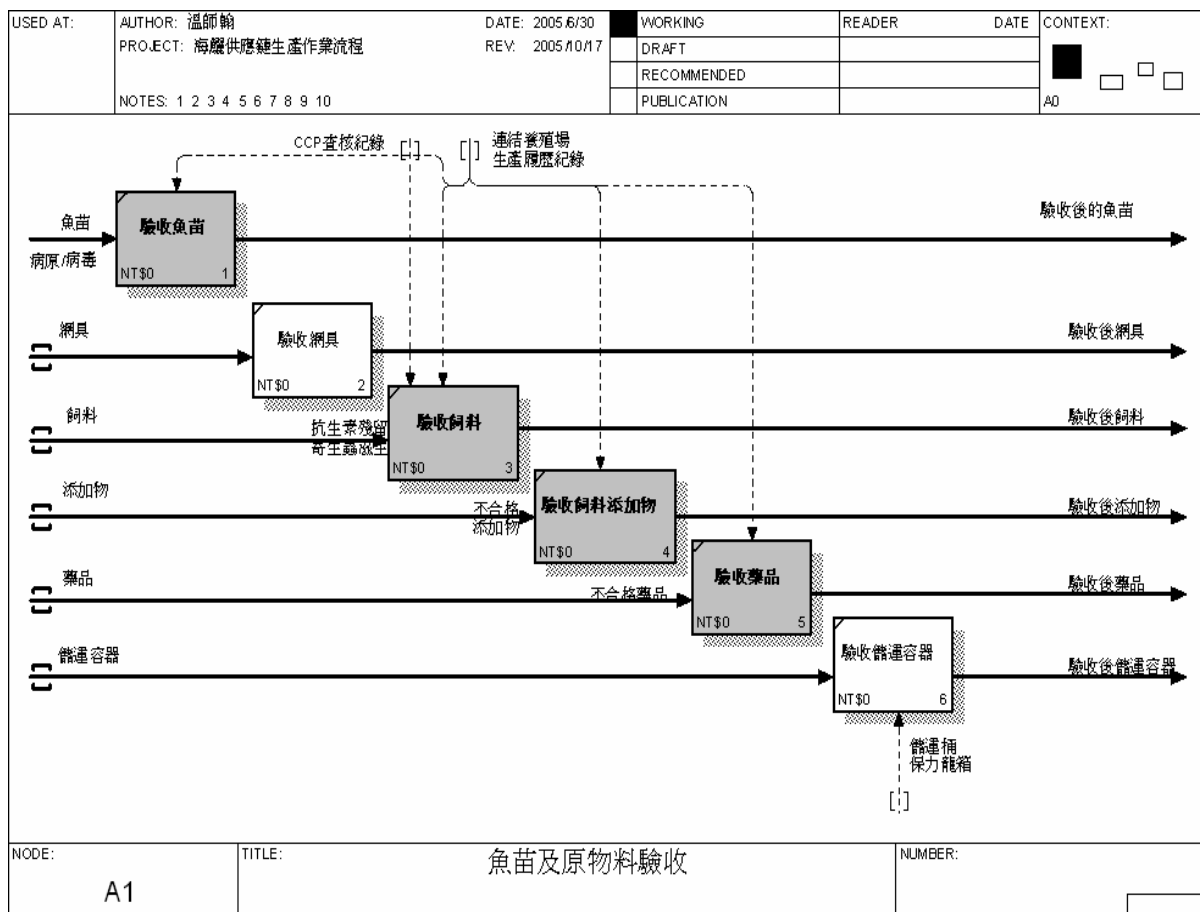


圖 3.3 海鰻供應鏈魚苗及原物料驗收作業流程IDEF0 圖[23]

圖 3.4 是海鰻供應鏈儲藏設施與作業流程 IDEF0 圖，為圖 3.2 中儲藏設施與作業的展開圖，目的在說明儲藏設施與作業的細部工作內容。儲藏設施與作業是承接魚苗與原物料驗收作業之後的程序。在圖 3.3 的魚苗及原物料驗收作業中，除了魚苗以外，其餘五項作業如驗收網具、驗收飼料、驗收飼料添加物、驗收藥品以及驗收儲運容器，在驗收完成後均會進入儲藏設施與作業流程中個別進行儲藏，分別是網具儲藏作業、容器儲藏作業、飼料儲藏作業、添加物儲藏作業以及藥品儲藏作業。針對儲藏設施與作業流程中監控的重要點有三項，分別是飼料儲存、添加物儲存以及藥品儲存。而儲藏設施與作業的支援人員與設備主要是倉儲設施以及倉儲管理人員。

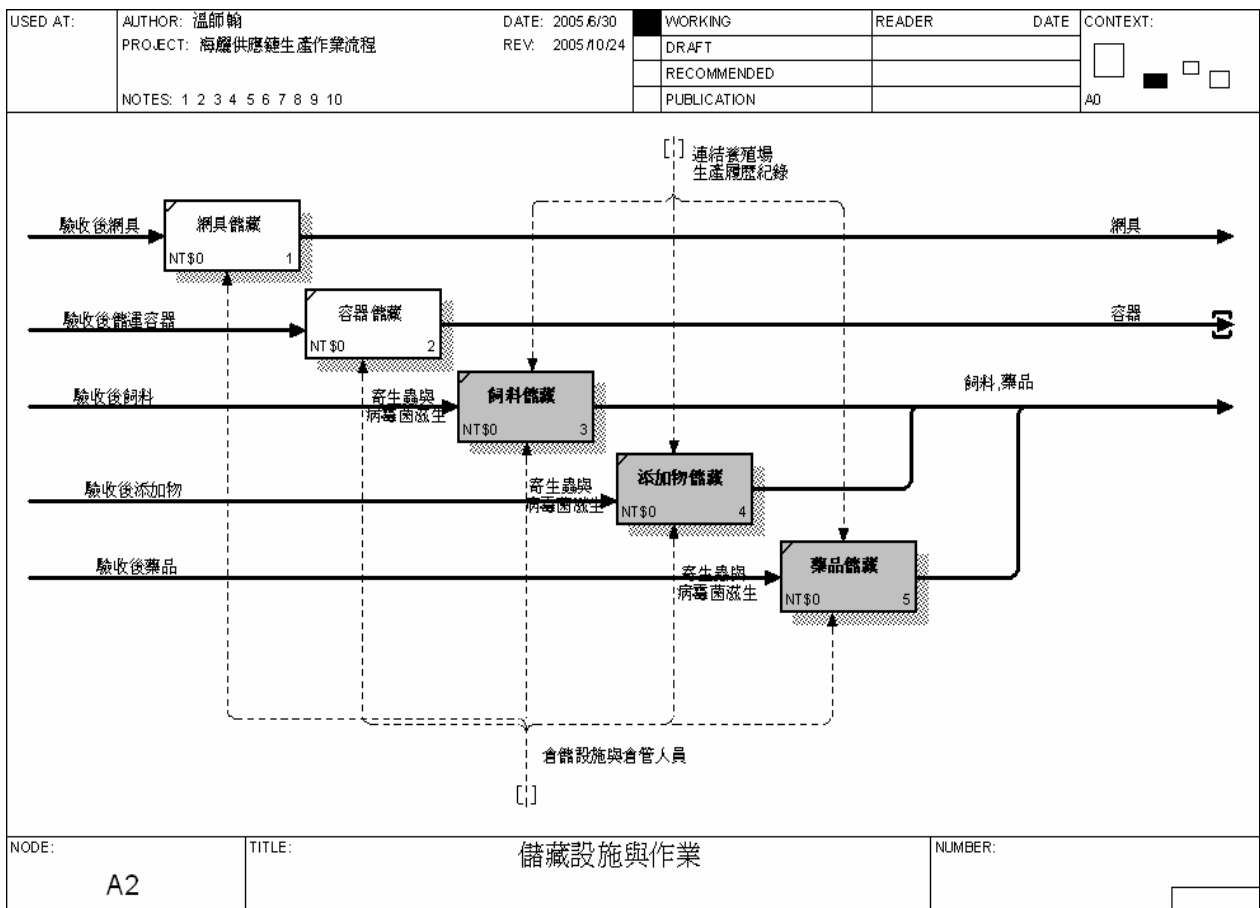


圖 3.4 海鱺供應鏈儲藏設施與作業流程 IDEF0 圖 [23]

圖 3.5 為海鱺供應鏈養殖作業流程 IDEF0 圖，是圖 3.2 中的養殖作業方框展開圖，目的在於說明養殖作業的細部工作內容。養殖作業可分為下網換網、分魚、養殖、撈魚等四項工作。養殖作業傳遞的物流為經過驗收完成的魚苗；輸出的物流則為被捕撈的成魚。傳遞的資訊流則是放養紀錄表單、飼養紀錄表單、用藥紀錄表單的內容，放養紀錄表單內容包括養殖場、箱網編號、放養日期、魚苗規格、魚苗批號、數量以及單位；飼養紀錄表單則包含養殖場、箱網編號、餵食日期、餵食時間、餵食飼料、數量與單位；用藥紀錄表單則包含養殖場、箱網編號、投藥日期、投藥時間、數量以及單位。而輸出的資訊流則為收穫紀錄表單，收穫紀錄表單內容包含養殖場、箱網編號、養殖批號、收穫日期、數量以及單位。針對養殖作業流程的重要點是養殖作業。而養殖作業流程的支援人員與設備主要是各項作業的工作人員。



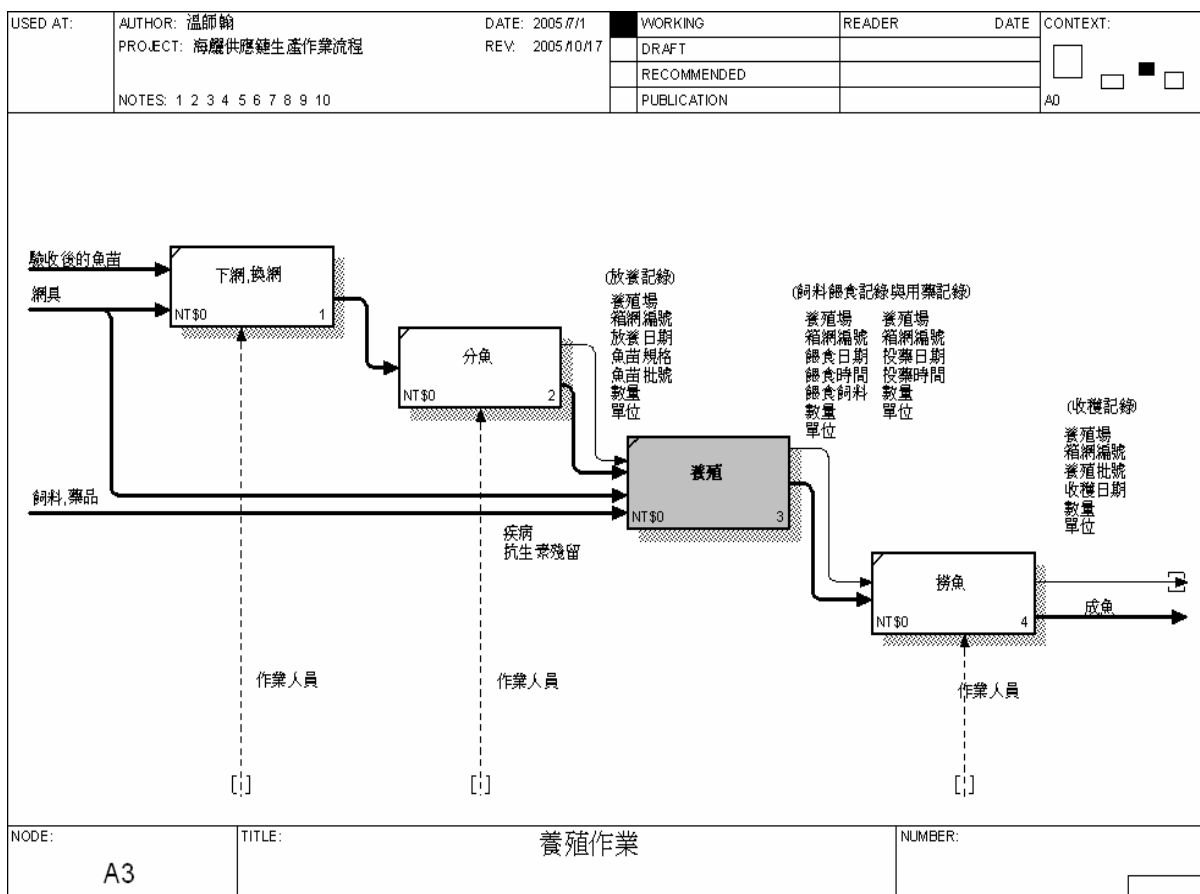


圖 3.5 海鱺供應鏈養殖作業流程IDEF0 圖[23]

圖 3.6 為海鱺供應鏈包裝出貨作業流程 IDEF0 圖，是圖 3.2 中的包裝出貨作業方框展開圖，目的在於說明包裝出貨作業的細部工作內容。包裝出貨可分為檢驗、放血、冰鎮包裝、運輸與銷售五項工作。包裝出貨作業傳遞的物流是捕撈的成魚及容器，經過作業流程中的包裝，最後輸出準備外銷的魚貨成品。而傳遞的資訊流分為檢驗表單與出貨紀錄表單兩部份，檢驗表單的內容包括檢驗表單序號、檢驗國家、檢驗中心、檢驗類別、檢驗項目、檢驗值與單位；出貨紀錄表單內容則包含養殖場、出貨日期、出貨批號、販運商、運輸工具、包裝方式、數量及單位。最後的資訊流輸出則是海鱺身分識別證，此識別證含有條碼資訊，內容包括廠商代碼、出貨批號、魚種、重量以及日期。針對包裝出貨作業流程的重要點有三項，分別是放血、冰鎮包裝與銷售三項工作，以填滿灰色的方框表示。包裝出貨作業支援的人員與設備包括專門從事檢驗的人員與設備、運輸魚貨的運輸工具以及魚貨的買主。

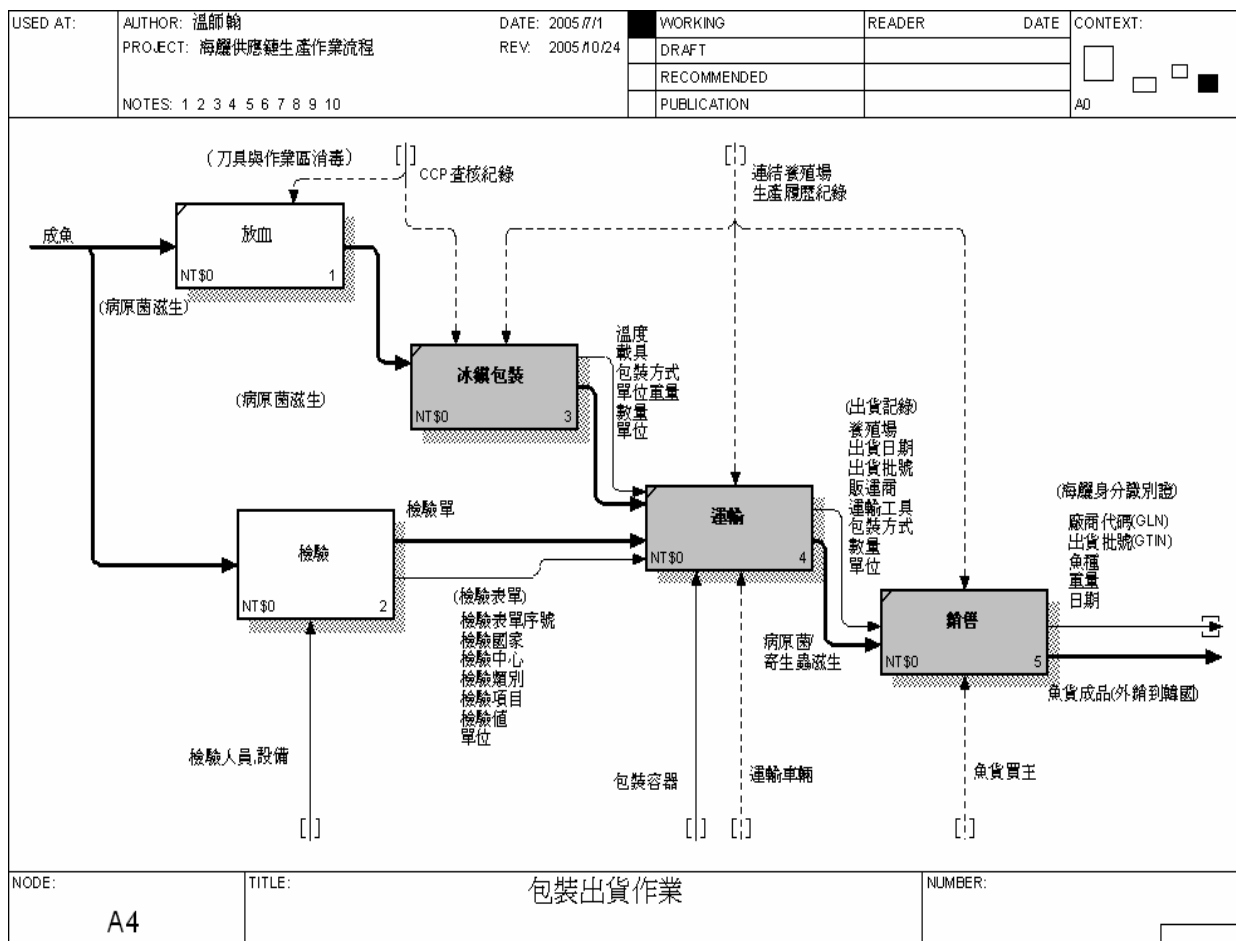


圖 3.6 海鱺供應鏈包裝出貨作業流程IDEF0 圖[23]

### 3.1.2 海鱺供應鏈作業流程裴氏圖

分析海鱺供應鏈作業流程的目的在於研究供應鏈作業中的物流與資訊流動向。從 3.1.1 節的 IDEF0 流程圖中可以看到四大作業流程的靜態細部內容，而完整的海鱺供應鏈作業流程則可以利用裴氏圖(Petri Net)[46]來表達。IDEF0 流程圖與裴氏圖之間的轉換過程透過階層轉換法[18]，把 IDEF0 中的作業方格(Activity Box)視為裴氏圖中的轉移點(Transition)，代表正在進行的動作；而將 IDEF0 中的箭號(Arrow)視為裴氏圖中的暫存點(Place)，代表該物流或資訊流的屬性狀態。關於階層轉換法的流程如圖 3.7 所示。當裴氏圖確定整個作業流程的物流與資訊流轉換流程沒有發生鎖死(Dead Lock)的情況，接續才設計整個工作日誌資料庫的內容，以建置接下來要進行的工作日誌系統。

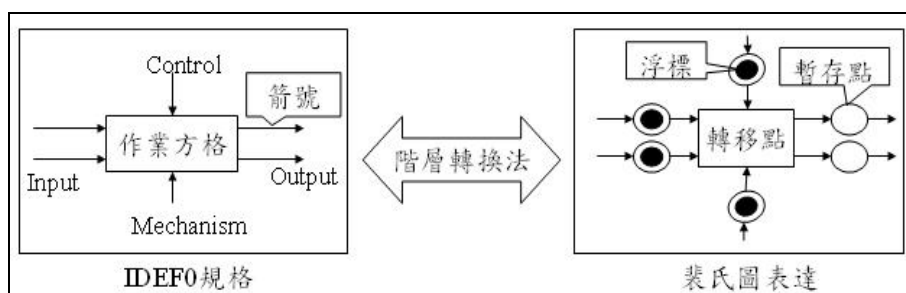


圖 3.7 階層轉換法[7]

本研究以HPSim軟體[40]作為設計裴氏圖的工具，HPSim是一套利用C++語言開發的裴氏圖設計環境軟體，提供網路上免費下載。該軟體具備圖形化介面來進行基本的裴氏圖編輯與執行：以空心圓形圖示代表暫存點；黑色填實矩形代表轉移點；實心圓形代表浮標。根據3.1.1節海鱸供應鏈IDEF0圖將作業方格與箭號轉換後，可以得到如圖3.8所示的裴氏圖。包含25個轉移點以及29個暫存點，其動態行為由轉移點的激發來控制。

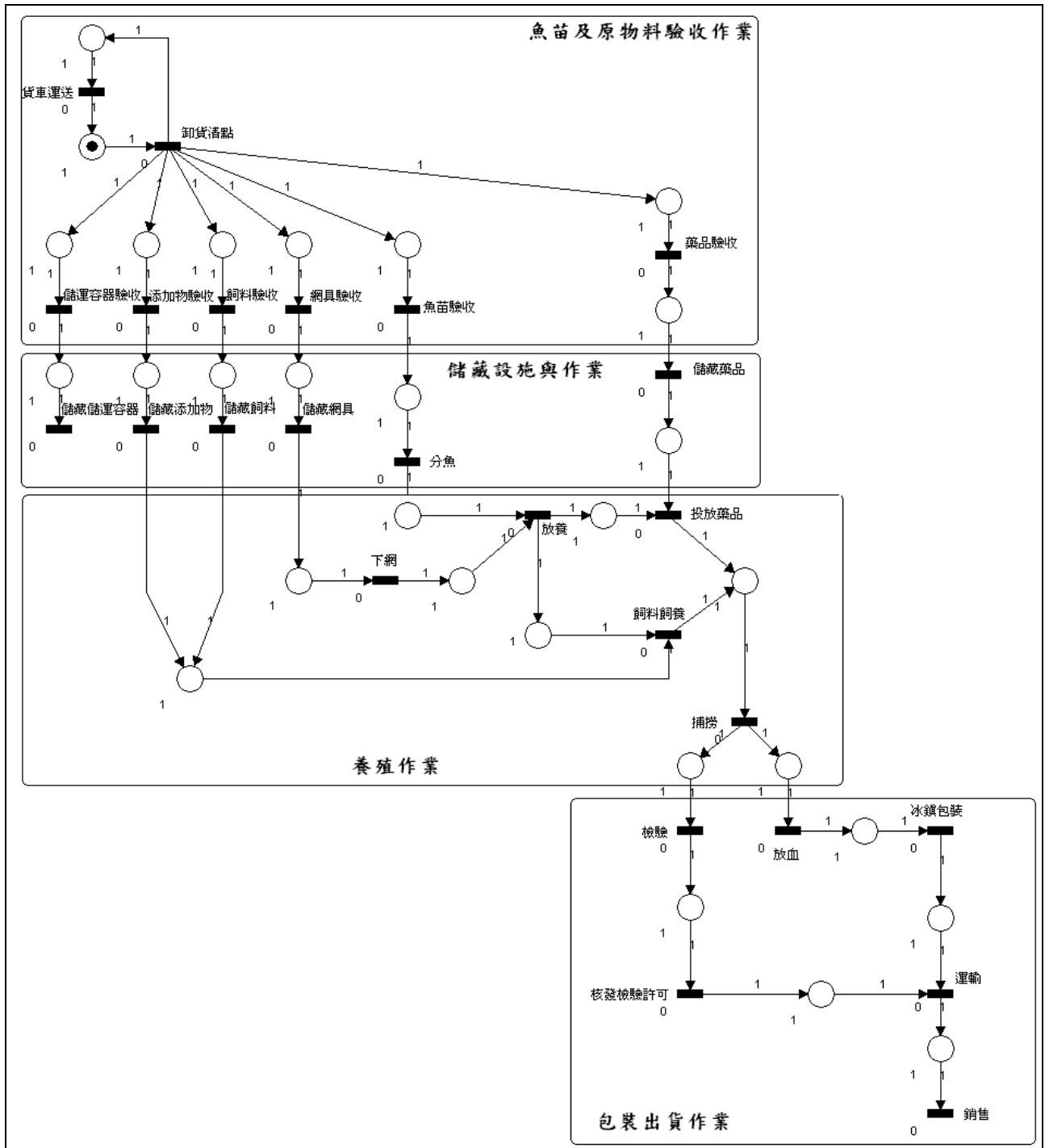


圖 3.8 海鱸供應鏈作業流程裴氏圖

## 3.2 工作日誌系統三階正規化分析與設計

本節主要目的在對海鱸工作日誌進行關聯式資料庫三階正規化分析與設計，共分為三小節。3.2.1 節說明在「設計階層的模型架構」；3.2.2 節說明「工作日誌系統從 IDEF0 到 IDEF1X 規格」的整合方法；而 3.2.3 節說明「工作日誌系統三階正規化資料庫規格」的架構。

### 3.2.1 設計階層式架構模型

設計階層是用來設計應用發展程序的目的而做的資料模型，在不同的系統架構之下，資料模型可能是單一的或是一組。階層可以視為建構資料模型時期的階段性目標。例如建置企業資源規劃系統時，分拆為數個單一資料模型設計階層的模組，例如：財務會計模組、詢價與報價模組、客戶管理模組、生管模組等。每個設計階層都是階層式架構模型的一部份，如圖 3.9 所示。第一階層是邏輯模型，第二階層是實體模型，而第三階層是特定資料庫的實體模型。圖 3.10 表示設計階層與工作日誌系統資料庫設計的關係對照圖。

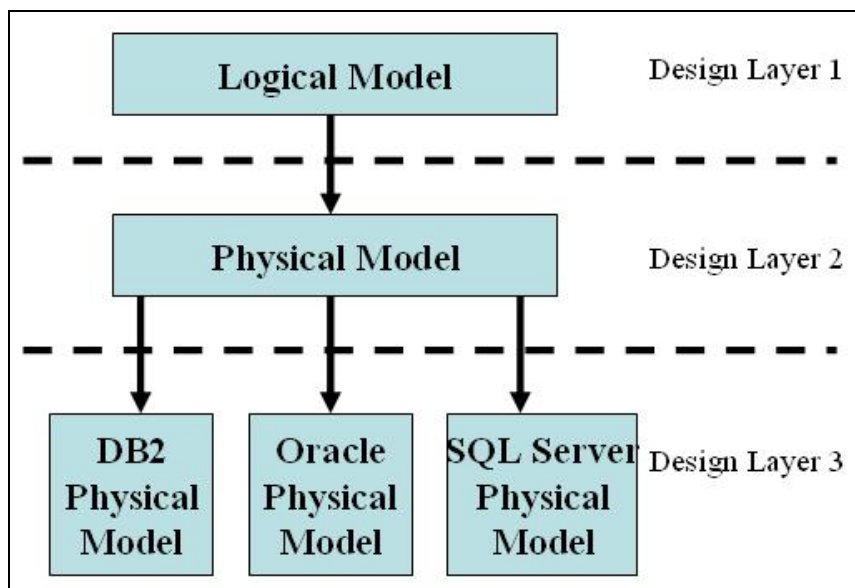


圖 3.9 階層式架構模型圖[7]

第一階層是概念性邏輯資料模型，目的是擷取公司在設計應用發展程序時的需求。對工作日誌的資料庫系統而言，第一階層必須討論海鱸養殖業者對於此資料庫系統的基本需求，例如資料庫設計的目的在於提高記錄效率、降低風險成本等；對資訊方面的需求則以整條供應鏈為出發點，資料內容包含養殖戶資料、供應商資料、販運商資料、飼料資料、藥品資料、相關檢驗資料、出貨資料；對系統特性分成多個應用系統，包含養殖記錄系統、供應商管理系統、出貨管理系統等。在概念性邏輯資料模型中就是把這些需求建立實體關係模型，並將資料間的關聯性與資料屬性建立實體關聯圖。

第二階層一般稱為實體模型，目的是將邏輯資料模型中的需求架構轉換為資料庫可執行的規則。對工作日誌資料庫系統而言，第二階層必須把第一階層設計的實體關係模型轉換為關聯式資料庫模式並建立三階正規化的關聯式資料庫模型。



第三階層為特定資料庫實體模型[43][28]，目的是把第二階層設計的關聯式資料庫模型轉換到不同的資料庫系統介面上實作。對於工作日誌資料庫系統而言，第三階層的工作是將第二階層建立好的三階正規化關聯式資料庫模式轉換到SQL Server、Oracle、DB2 等不同廠牌的資料庫系統介面中，為了配合工作日誌系統整體軟硬體系統介面，選擇SQL Server2000 實作特定資料庫實體模型。

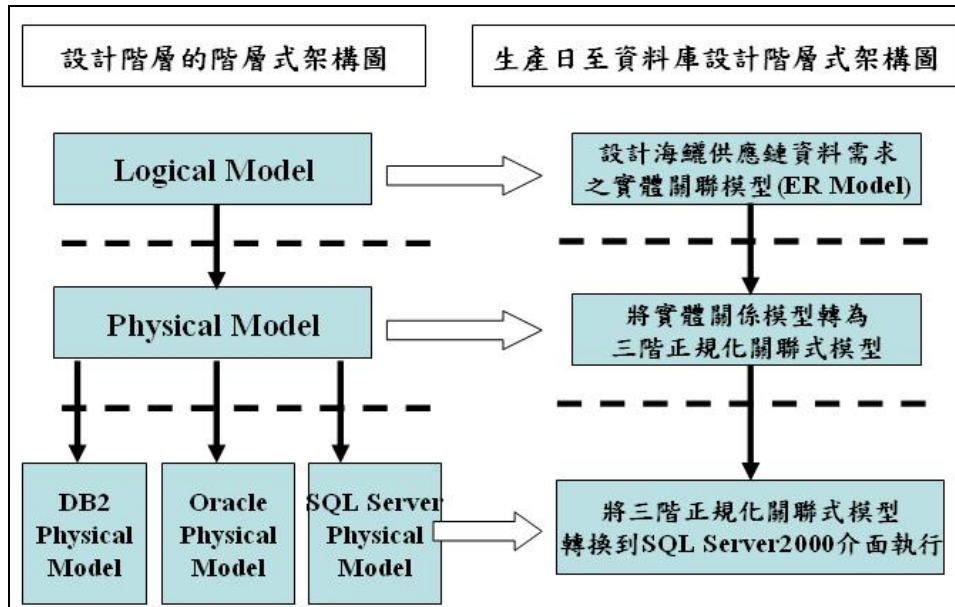


圖 3.10 設計階層與工作日誌資料庫設計關係對照圖

因為每個階層各代表不同時期的模型系統，系統無法在同一個模型中呈現設計程序內的各個階層，反而必須強調階層內容的一致性與同步性，使其能夠在同一套系統中具備三項功能：第一項是建構不同階層間的關聯性並將之連結；第二項是在每個階段裡面做不同的設計決策，並記錄各階段的轉換過程；第三項是當進行維護動作，對於不同階層內容有所改變時，能夠同步進行更動。而 Computer Associates 公司的軟體 ERwin Data Modeler 錯誤! 找不到參照來源。能夠提供這三項環環相扣的關係連結、轉換與維護。故本章會使用 ERwin Data Modeler 軟體設計工作日誌資料庫系統。

### 3.2.2 工作日誌系統從 IDEF0 到三階正規化 IDEF1X 規格

對於設計工作日誌資料庫而言，依照三綱要模型(3-schema model)可以分為概念性綱要、內部綱要與外部綱要三種，概念性綱要是從IDEF0 初步轉換到IDEF1X的規格，使用對象為資料庫分析設計人員；內部綱要是經過三階正規化[28]之後的IDEF1X規格，其對象是資料庫內部；而外部綱要主要是配合應用程式模組的呈現，使用對象是直接面對應用程式的使用者。三綱要模型如圖 3.11 所示。

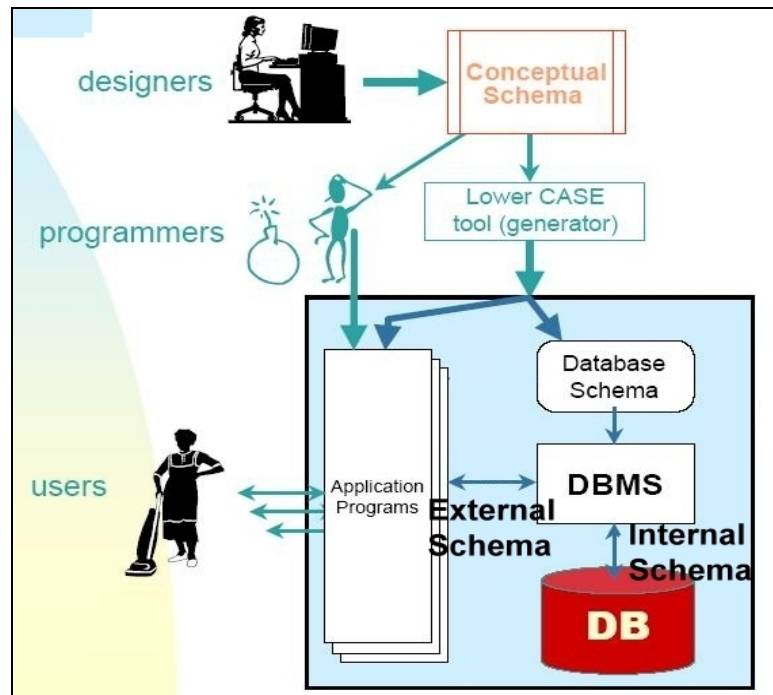


圖 3.11 三綱要模型圖[23]

海鱺供應鏈生產作業流程在物流與資訊流的IDEF0 與IDEF1X轉換技術[18][25]中，首先在現場與專業人士訪談時取得作業需求的相關資訊，在 3.1.1 節的海鱺供應鏈作業IDEF0 階層圖，將系統中各活動之間的關係與其相關資訊清楚表達，完成海鱺供應鏈生產作業流程IDEF0 規格設計。接著從海鱺供應鏈生產作業四大流程中相關的物流與資訊流擷取每項資料屬性，並將這些屬性集合生成一張整體資料表，再初步將整體資料表分解成滿足三階正規化形式條件的數個子關聯表。如圖 3.12 所示，此步驟相當於工作日誌資料庫設計階層中的概念性邏輯模型。IDEF0 是圖形化的動作流程描述工具，其在雙元性方面可以把動詞與名詞互換變成計算模式轉為IDEF1X圖。而IDEF1X規格代表資料庫基本表格的組成，基本表格上半部放置主鍵，下半部則放置非主鍵。基本表格用實線連接表示二階正規化；而虛線連接則代表三階正規化。線段的末端以實心圓形表示關聯性的「多」；而線段末端沒有任何特殊符號則表示關聯性的「一」；因此關聯性有多對多關聯、一對多關聯、一對一關聯三種可能性。圖 3.12 的IDEF1X 的規格乃是直接經由IDEF0 的物流與資訊流屬性直接產生，並未經過任何的正規化，此時的資料表以及對應關係雖然可以在概念層透過IDEF1X規格表示，但是在實體層的資料庫是無法做到的。例如圖 3.12 的左下方的藥品供應商與用藥紀錄兩張表格，關聯性是多對多關係，但是實體資料庫規格沒有辦法做到多對多的關係，必須要經過後續的正規化步驟，才能加以拆解成為資料庫可以產生的資料表格式。

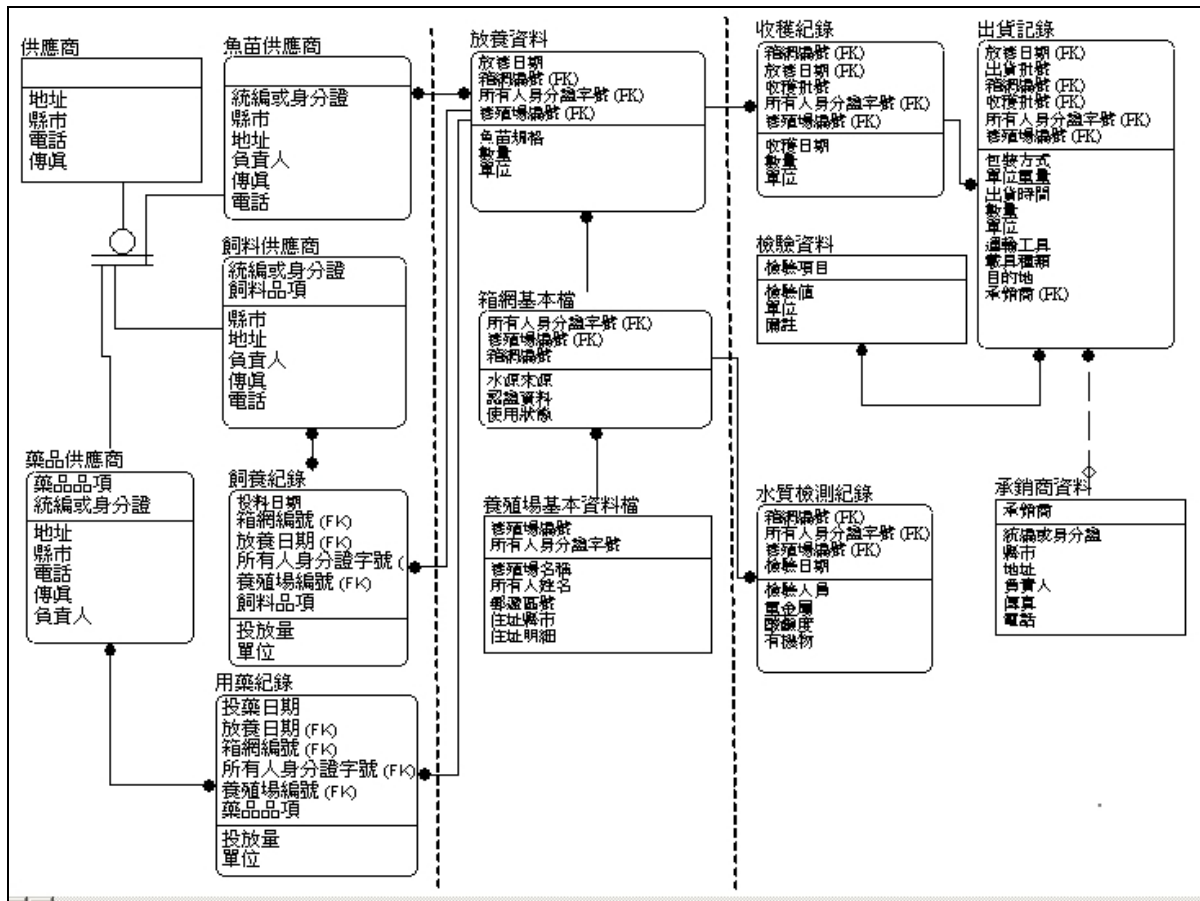


圖 3.12 概念性綱要IDEF1X規格[23]

接著透過分解法把未經過三階正規化的 IDEF1X 圖依照功能相依特性逐步拆解，建立工作日誌資料庫正規化。圖 3.13 以藥品供應商與用藥紀錄兩張多對多關聯的表格為拆解範例，將藥品供應商資料表與用藥紀錄資料表新增一張具有關聯性的合成資料表，命名為藥品明細檔資料表，並建立新的關聯主鍵「藥品品項」使其具備唯一的識別性，組成三張關聯式資料表。由於表格眾多，其餘的表格關聯性拆解過程同理可得。依照分解法最後可以將圖 3.12 的概念性綱要拆解為圖 3.14 的出海鱸供應鏈生產作業內部綱要 IDEF1X 規格，經過三階正規化的內部綱要 IDEF1X 規格共計有十八張資料庫基本表格。

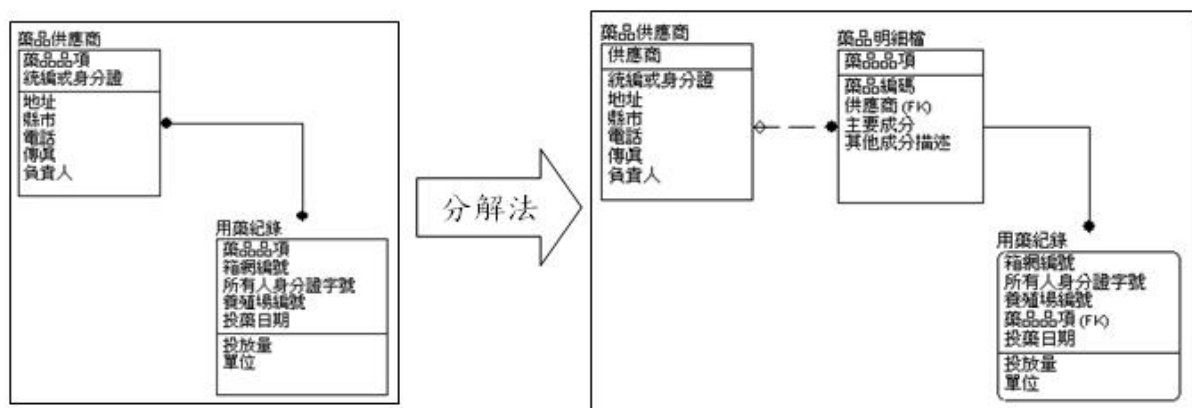


圖 3.13 「藥品供應商」與「用藥紀錄」多對多關聯表的拆解

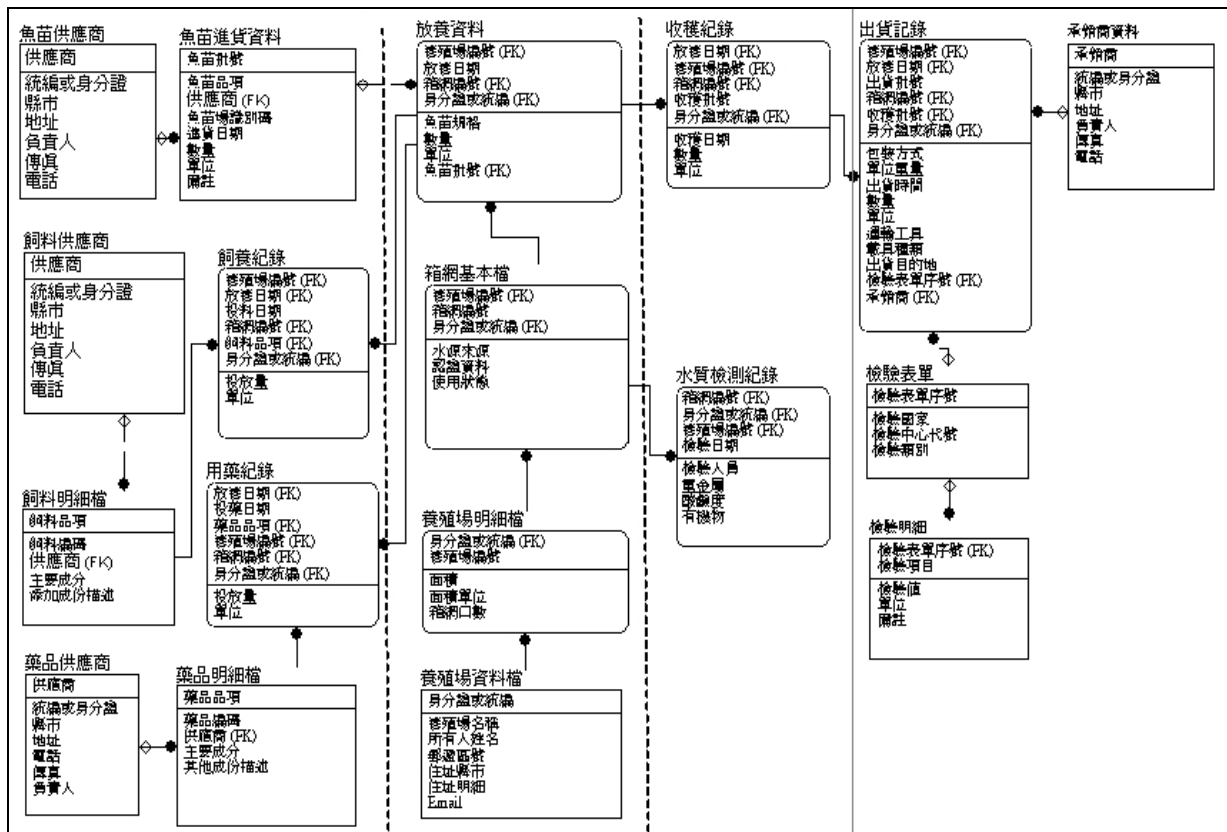


圖 3.14 內部綱要IDEF1X規格[23]

根據圖 3.14 的三階正規化 IDEF1X 規格，接著建立外部綱要。外部綱要透過應用程式與程式使用者互動，總共有五大模組：供應商基本資料模組、養殖場基本資料模組、養殖日製模組、檢驗紀錄資料模組與出貨紀錄模組。圖 3.15 同步顯示外部綱要五大模組與工作日誌應用程式模組的對應關係。在 3.1 節的海鱷生產供應鏈作業流程 IDEF0 規格中把整個作業流程分為四大項：魚苗及原物料驗收作業、儲藏設施與作業、養殖作業與包裝出貨作業。在轉換為 IDEF1X 的外部綱要時，魚苗及原物料驗收作業以及儲藏設施與作業兩項合併成為供應商基本資料模組；另外由於第四章追溯系統的需求，因此新增加養殖場基本資料模組以便追溯養殖場負責人相關資訊。圖 3.15 中間填滿灰色的大方框是根據 HACCP 定義所制定的重要管制點，由於檢驗紀錄資料模組是整個供應鏈作業流程中最重要監控部分，因此特別獨立開發一模組強化相關工作日誌項目的追蹤與追溯。



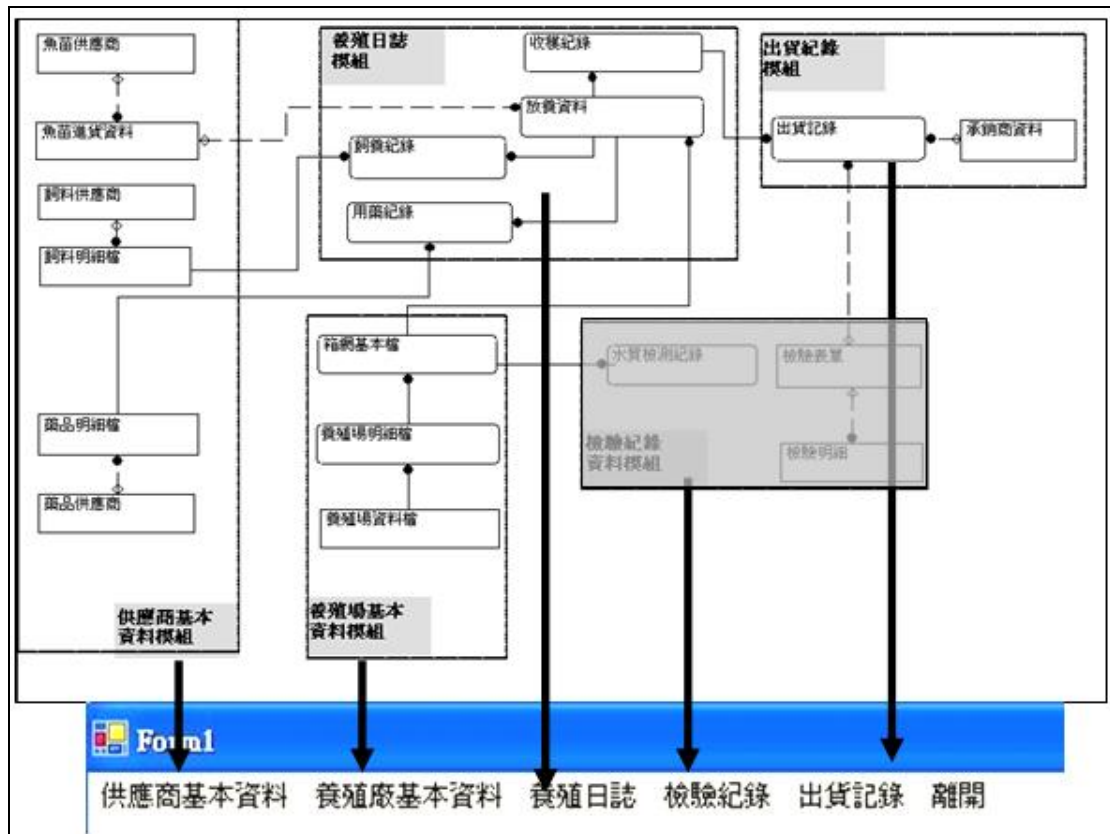


圖 3.15 外部網要五大模組與應用程式模組對應圖[23]

### 3.2.3 工作日誌系統三階正規化資料庫規格

由圖 3.14 工作日誌系統 IDEF1X 表達法，可以在 SQL Server 2000 [11][12] 建立三階正規化資料表共計十八張。表 3.1 為養殖場資料檔資料表，記錄養殖戶相關的基本資料；表 3.2 為養殖場明細檔資料表，記錄關於養殖場的面積或編號等資訊；表 3.3 為箱網基本檔資料表，此表為養殖場明細檔資料表經過二階正規化之後分出的資料表，目的在記錄關於箱網編號、水源、使用狀態等資訊；表 3.4 為水質檢測紀錄資料表，主要記錄重金屬、酸鹼度與有機物的檢測資訊；表 3.5 為放養資料資料表，針對每個箱網，以放養日期為主鍵值記錄魚苗放養的相關資訊；表 3.6 為飼養紀錄資料表，記錄每次的餵食飼料工作的日期、時間、飼料品項、投放量等資訊；表 3.7 為飼料明細檔資料表，目的是記錄飼料成分與飼料供應商等資訊；表 3.8 為飼料供應商資料表，此表為飼料明細檔資料表經過三階正規化之後分出的資料表，用途是記錄飼料供應商名稱、地址、電話、負責人等飼料供應廠商的相關資料；表 3.9 為用藥紀錄資料表，記錄每次的藥物投放工作的日期、時間、藥品品項、投藥量；表 3.10 為藥品明細檔資料表，目的是記錄藥品的成分以及藥品供應商等資訊；表 3.11 為藥品供應商資料表，此表為藥品明細檔資料表經過三階正規化之後分出的資料表，用途為記錄藥品供應商名稱、地址、電話、負責人等藥品供應廠商的相關資料；表 3.12 為魚苗進貨資料資料表，記錄魚苗進貨的批號、日期、數量與供應商等資料；表 3.13 為魚苗供應商資料表，此表從魚苗進貨資料資料表經過三階正規化之後分出，用途為記錄魚苗供應商名稱、地址、電話、負責人等魚苗供應廠商的相關資料；表 3.14 為收穫紀錄資料表，目的是針對箱網的每次收穫記錄收穫日期、收穫數量等資訊；表 3.15 為檢驗表單資料表，目的是記錄檢驗表單序號、檢驗國家、檢驗中

心、檢驗類別等資訊；表 3.16 為檢驗明細資料表，此表是從檢驗表單資料表經過三階正規化之後分出來的資料表，目的是記錄檢驗項目、檢驗值與單位等資訊；表 3.17 為出貨記錄資料表，目的是記錄出貨批號、出貨日期、包裝方式、載具、目的地、承銷商與檢驗表單序號等資訊；表 3.18 為承銷商資料資料表，此表是從出貨記錄資料表經過三階正規化之後分出的資料表，用來記錄承銷商名稱、地址、電話、負責人等相關資訊。其中各資料表的主鍵以及外鍵以「◎」標示；而允許空值以「○」表示允許狀態。

表 3.1 養殖場資料檔資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		養殖場名稱	varchar		40	公司名稱或是養殖戶
		所有人姓名	varchar		30	負責人
		身分證或統編	varchar		20	公司統編或自然人身分證
		郵遞區號	varchar	◎	5	三碼或五碼郵遞區號
		住址縣市	varchar	◎	20	台灣地區各縣市
		住址明細	varchar	◎	40	
		Email	varchar	◎	40	電子郵件信箱

表 3.2 養殖場明細檔資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	身分證或統編	varchar		20	公司統編或自然人身分證
◎		養殖場編號	varchar		30	
		面積	varchar	◎	20	
		面積單位	varchar	◎	20	養殖場面積使用的單位
		箱網口數	varchar		10	該養殖場內的箱網數量

表 3.3 箱網基本檔資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	養殖場編號	varchar		30	
◎		箱網編號	varchar		30	
◎	◎	身分證或統編	varchar		20	
		水源來源	char	◎	20	鹹水、淡水
		認證資料	varchar	◎	20	通過認證的項目
		使用狀態	char	◎	1	目前使用狀態

表 3.4 水質檢測紀錄

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	箱網編號	varchar		30	
◎	◎	身分證或統編	varchar		30	

◎	◎	養殖場編號	varchar		20	
◎		檢驗日期	varchar		4	進行水質檢測的日期
		檢驗人員	varchar	◎	30	進行水質檢測的人員
		重金屬	varchar	◎	40	重金屬檢測描述
		酸鹼度	smalldatetime	◎	20	酸鹼度檢測描述
		有機物	varchar	◎	20	有機物檢測描述

表 3.5 放養資料資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	養殖場編號	varchar		30	
◎	◎	身分證或統編	varchar		20	
◎	◎	箱網編號	varchar		30	
◎		放養日期	smalldatetime		4	
		魚苗規格	varchar		20	記錄放養魚苗的規格
		數量	varchar		20	放養的數量
		單位	varchar		20	放養數量使用的單位
		魚苗批號	varchar		20	該批魚苗批號

表 3.6 飼養紀錄資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	養殖場編號	varchar		30	
◎	◎	放養日期	smalldatetime		4	
◎		投料日期	datetime		8	飼料餵食的日期
◎	◎	箱網編號	varchar		30	
◎	◎	飼料品項	varchar		30	
◎	◎	身分證或統編	varchar		20	
		投放量	varchar		20	該次餵食飼料的數量
		單位	varchar		20	餵食飼料數量的單位

表 3.7 飼料明細檔資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		飼料品項	varchar		30	
		飼料編碼	varchar		20	
	◎	供應商	varchar		30	飼料的供應商名稱
		主要成分	varchar	◎	40	飼料成分描述
		添加成份描述	varchar	◎	40	飼料添加物成分描述

表 3.8 飼料供應商資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		供應商	varchar		30	
		統編或身分證	varchar		20	供應商統編或身分證
		縣市	varchar	◎	20	所在縣市
		地址	varchar	◎	40	
		負責人	varchar	◎	30	供應商的負責人
		傳真	varchar	◎	20	
		電話	varchar	◎	20	

表 3.9 用藥紀錄資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	放養日期	smalldatetime		4	
◎		投藥日期	datetime		8	該次藥品投放的日期
◎	◎	藥品品項	varchar		30	投放的藥品品項
◎	◎	養殖場編號	varchar		30	
◎	◎	箱網編號	varchar		30	
◎	◎	身分證或統編	varchar		20	
		投放量	varchar		20	藥品投放數量
		單位	varchar		20	藥品投放數量的單位

表 3.10 藥品明細檔資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		藥品品項	varchar		30	
		藥品編碼	varchar		20	藥品的商品識別碼
	◎	供應商	varchar		30	藥品供應商名稱
		主要成分	varchar	◎	40	藥品的主要成分描述
		其他成份描述	varchar	◎	20	

表 3.11 藥品供應商資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		供應商	varchar		30	藥品供應商名稱
		統編或身分證	varchar		20	
		縣市	varchar	◎	20	供應商所在縣市
		地址	varchar	◎	40	
		電話	varchar	◎	20	
		傳真	varchar	◎	20	



		負責人	varchar	◎	30	藥品供應商負責人
--	--	-----	---------	---	----	----------

表 3.12 魚苗進貨資料資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		魚苗批號	varchar		20	購入的魚苗編號
		魚苗品項	varchar		30	魚苗品名
	◎	供應商	varchar		30	
		魚苗場識別碼	varchar		30	魚苗場的 GLN
		進貨日期	smalldatetime		4	該批魚苗購入日期
		數量	varchar		20	購入該批魚苗數量
		單位	varchar		20	魚苗數量的單位
		備註	varchar	◎	40	魚苗的補充描述

表 3.13 魚苗供應商資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		供應商	varchar		30	
		統編或身分證	varchar		20	
		縣市	varchar	◎	20	供應商所在縣市
		地址	varchar	◎	40	
		負責人	varchar	◎	30	供應商負責人名稱
		傳真	varchar	◎	20	
		電話	varchar	◎	20	

表 3.14 收穫紀錄資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	放養日期	smalldatetime		4	
◎	◎	養殖場編號	varchar		30	
◎	◎	箱網編號	varchar		30	
◎		收穫批號	varchar		40	收穫時的識別碼
◎	◎	身分證或統編	varchar		20	
		收穫日期	smalldatetime		4	該次收穫的日期
		數量	varchar		20	捕撈漁獲的數量
		單位	varchar		20	漁獲數量的單位

表 3.15 檢驗表單資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		檢驗表單序號	varchar		20	檢驗表單識別碼

		檢驗國家	varchar		20	檢驗單位所在國家
		檢驗中心代號	varchar		20	檢驗單位名稱
		檢驗類別	varchar		20	

表 3.16 檢驗明細資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	檢驗表單序號	varchar		20	
◎		檢驗項目	varchar		20	執行檢驗的項目
		檢驗值	varchar		20	該項目的檢測值
		單位	varchar		20	檢測值的單位
		備註	varchar	◎	20	

表 3.17 出貨紀錄資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎	◎	養殖場編號	varchar		30	
◎	◎	放養日期	smalldatetime		4	
◎		出貨批號	varchar		40	出貨識別碼
◎	◎	箱網編號	varchar		30	
◎	◎	收穫批號	varchar		40	
◎	◎	身分證或統編	varchar		20	
		包裝方式	varchar	◎	20	產品包裝的方式
		單位重量	varchar	◎	20	
		出貨時間	datetime	◎	8	出貨的日期
		數量	varchar		20	該批號出貨數量
		單位	varchar		20	出貨數量的單位
		運輸工具	varchar	◎	20	
		載具種類	varchar	◎	20	
		出貨目的地	varchar	◎	20	
	◎	檢驗表單序號	varchar		20	
	◎	承銷商	varchar		30	販運商等買方名稱

表 3.18 承銷商資料資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		承銷商	varchar		30	販運商等買方名稱
		統編或身分證	varchar		20	買方統編或身分證
		縣市	varchar	◎	20	買方所在縣市
		地址	varchar	◎	40	

		負責人	varchar	◎	30	買方負責人
		傳真	varchar	◎	20	
		電話	varchar	◎	20	



### 3.3 工作日誌系統

本節主要針對 3.2 節所設計的海鱷工作日誌資料庫開發相對應的應用程式，讓養殖場可以依照自己的需求，將生產履歷予以電子化保存。3.3.1 節說明「工作日誌系統設計裴氏圖」；而 3.3.2 節說明「工作日誌系統操作說明」

#### 3.3.1 工作日誌系統設計裴氏圖

根據實作的工作日誌系統，得到如圖 3.16 的裴氏圖。包含 19 個轉移點以及 26 個暫存點，其動態行為由轉移點的激發來控制。由裴氏圖的模擬運作可證明本系統不會產生死結狀態。

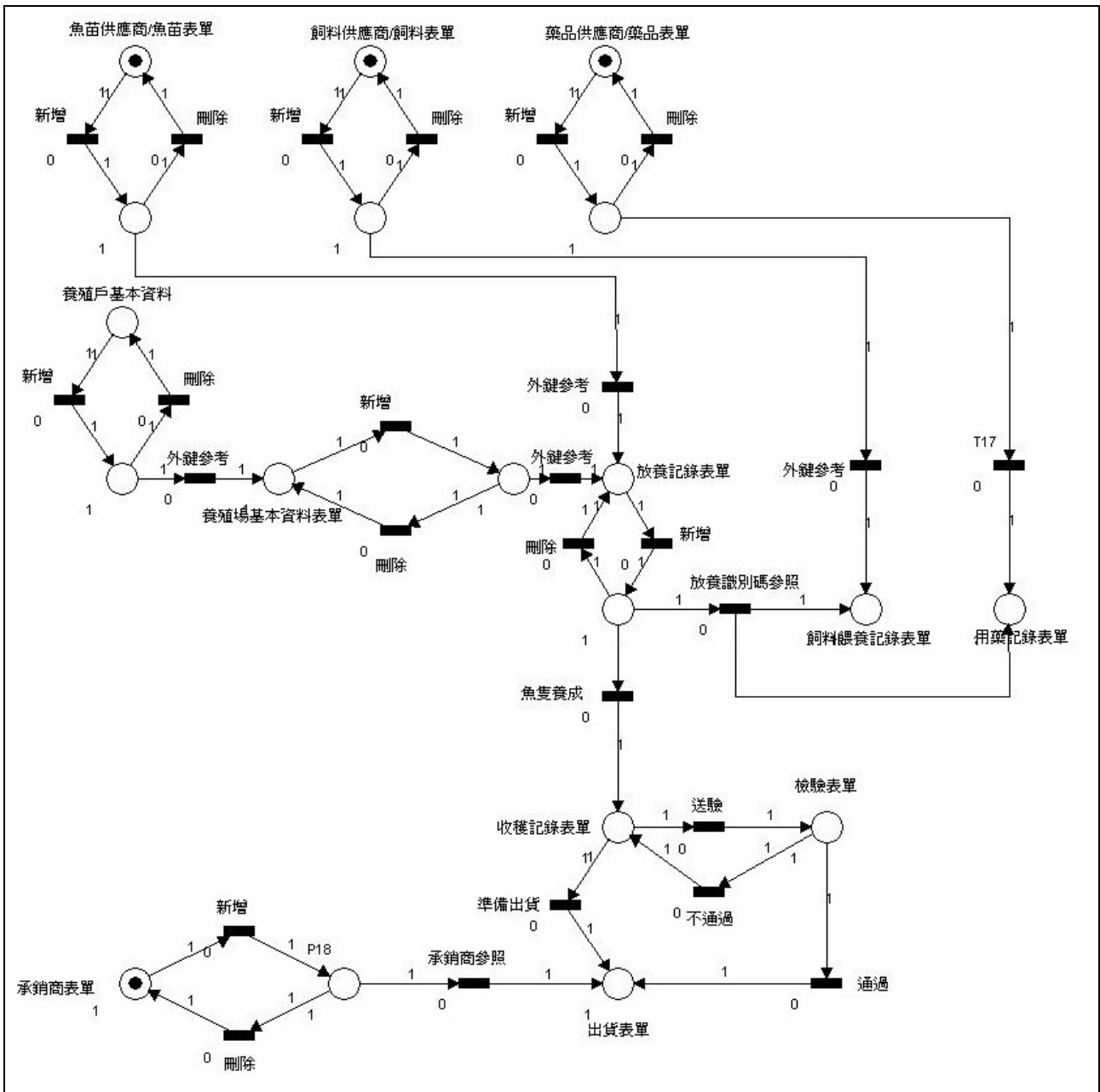


圖 3.16 工作日誌系統裴氏圖



### 3.3.2 工作日誌系統操作說明

工作日誌系統主畫面主要分為「供應商基本資料」、「養殖廠基本資料」、「養殖日誌」、「檢驗紀錄」、「出貨紀錄」五大項。以下將分別對各項內容予以說明。對「供應商基本資料」而言，其內有魚苗供應商/魚苗資料維護、飼料供應商/飼料資料維護、藥品供應商/藥品資料維護三項功能。魚苗供應商/魚苗資料維護為資料庫系統中魚苗供應商表及魚苗進貨資料表，記錄供應商的基本資料，紀錄時間，數量等，如圖 3.17 所示。

魚苗品項	魚苗批號	魚苗場識別	進貨日期	數量	單位	備註
海鱸魚苗	A00001	A01	2005/12/12	3000	尾	(unit)

圖 3.17 魚苗供應商/魚苗資料維護介面[23]

飼料供應商/飼料資料維護為資料庫系統中飼料供應商表及飼料明細表，記錄供應商的基本資料，飼料種類等，如圖 3.18 所示。

飼料編號	飼料品項	主要成分	添加成份
1249999999A	濃縮式飼料1	濃縮式飼料1	濃縮式飼料1
1249999999B	濃縮式飼料2	濃縮式飼料2	濃縮式飼料2

圖 3.18 飼料供應商/飼料資料維護介面[23]

藥品供應商/藥品資料維護為資料庫系統中藥品供應商表及藥品明細表，記錄供應商的基本資料，藥品種類，如圖 3.19 所示。



圖 3.19 藥品供應商/藥品資料維護介面[23]

對「養殖戶基本資料」而言，又分為養殖戶基本資料與養殖場基本資料兩項。養殖戶基本資料表單對應的是資料庫系統中的養殖場資料檔，為記錄養殖場負責人的資本資料，如圖 3.20 所示。



圖 3.20 養殖戶基本資料介面[23]

養殖場基本資料表單對應到資料庫系統中養殖場明細表及箱網基本資料表，記錄養殖場面積與箱網數等，如圖 3.21 所示。



圖 3.21 養殖場基本資料介面[23]

對「養殖日誌」而言，其內有放養紀錄檔、飼料餵養紀錄檔、用藥紀錄檔、收穫紀錄檔四項功能。放養紀錄檔表單所對應的資料庫系統為放養資料表，用以記錄魚苗規格、數量、批號等，如圖 3.22 所示。

圖 3.22 放養紀錄介面[23]

飼料餵養紀錄檔所對應的資料庫系統資料表為飼養紀錄表，功能是記錄飼料餵食時間、飼料品項等資訊，如圖 3.23 所示。

投料日期	飼料品項	投放量	單位	檢護

圖 3.23 飼料餵養紀錄介面[23]

用藥紀錄檔對應到資料庫系統的用藥資料表，功能是記錄投藥量、投藥時間與藥品品項等資訊，如圖 3.24 所示。

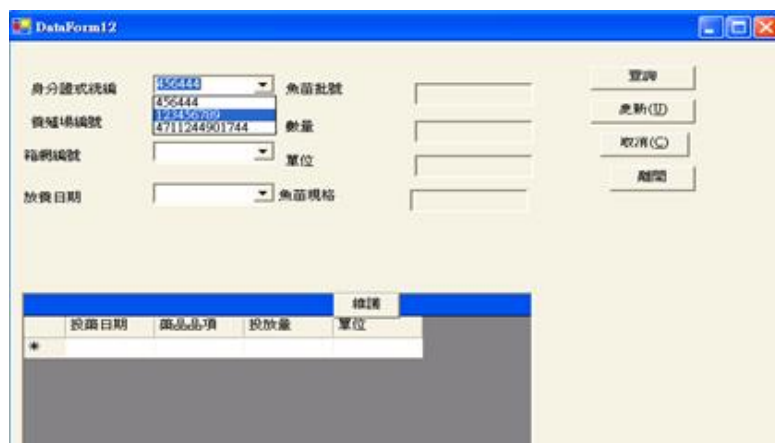


圖 3.24 用藥紀錄介面[23]

收穫紀錄檔對應資料庫系統中的收穫紀錄表，主要功能是記錄收穫日期、收穫數量、單位等資訊，如圖 3.25 所示。



圖 3.25 收穫紀錄介面[23]

對「檢驗紀錄」而言，其內容有水質檢測紀錄以及漁產品檢驗紀錄兩項。水質檢測紀錄所對應的資料庫系統資料表是水質檢測紀錄表，用途是進行各養殖池或箱網的水質檢查，記錄重金屬、酸鹼度與有機物的含量，如圖 3.26 所示。



圖 3.26 水質檢測紀錄介面[23]

漁產品檢驗紀錄則是用於漁產品出貨之前的抽驗，各種漁產品必須通過相關規定的抽檢，才能讓產品進入消費市場。漁產品檢驗紀錄對應於資料庫系統中的檢驗表單及檢驗明細



表，用以記錄各國標準的檢驗項目、檢驗單位、檢驗值及明細資訊等，如圖 3.27 所示。



圖 3.27 魚產品檢驗紀錄介面[23]

對「出貨紀錄」而言，其內容分為承銷商資料維護與出貨單維護兩項功能。承銷商資料維護對應於資料庫系統中的承銷商資料表，功能是記錄承銷商的相關資料，如圖 3.28 所示。



圖 3.28 承銷商資料維護介面[23]

出貨單維護表單對應於為資料庫系統的出貨紀錄表，功能是記錄出貨日期、承銷商、運輸方式、檢驗表單等資訊，如圖 3.29 所示。



圖 3.29 出貨單維護介面[23]

### 3.4 其他工作日志設計

關於供應鏈中其他端點的工作日志設計，將於本節說明其需求架構。在海鱈產銷供應鏈上，除了養殖場之外，主要還有運輸物流、加工廠與販售等廠商的參與。為了產銷履歷施行的完善，必須分別針對這些端點設計工作日志系統，以方便廠商將其作業流程電子化，同時使得產銷履歷的資訊透明度更加完整。表 3.19 為運輸物流日志基本資料表；表 3.20 為加工廠日志基本資料表；表 3.21 為販售商日志基本資料表。

表 3.19 運輸物流日志基本資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		發貨批號	varchar		30	發貨時的識別碼
◎		發貨公司	varchar		30	
		出發時間	smalldatetime	◎	4	
		運輸工具	varchar	◎	30	物流運輸交通工具
		抵達時間	smalldatetime	◎	4	
		運輸公司代碼	varchar	◎	10	
		出貨批號	varchar		30	出貨的識別碼
		轉運站代碼	varchar	◎	10	途經的轉運站代碼

表 3.20 加工廠日志基本資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		進貨批號	varchar		30	進貨的識別碼
◎		進貨公司	varchar		30	
		進貨時間	smalldatetime	◎	4	
		加工動作	varchar	◎	30	進行加工的動作
		包裝方式	varchar	◎	30	採用何種方式包裝
		出貨時間	smalldatetime	◎	4	
		出貨公司代碼	varchar	◎	10	
		出貨批號	varchar		30	出貨的識別碼

表 3.21 販售商日志基本資料表

主鍵	外鍵	欄位名稱	資料型別	允許空值	長度	說明
◎		進貨批號	varchar		30	進貨的識別碼
◎		進貨公司	varchar		30	
		進貨時間	smalldatetime	◎	4	
		進貨數量	varchar	◎	30	
		販售架號	varchar	◎	30	銷售的置物架編號
		上架時間	smalldatetime	◎	4	

## 第四章 履歷追蹤查詢系統架構－ 伺服器端設計

本章主要分成兩個部份，第一部份描述本案例－「海鱺生產供應鏈追溯與追蹤系統」架構的設計以及在此架構之下使用 Axis 套件建構 Web Services 的特性以及服務細節，內容為 4.1 節的「使用 Axis 實作 Web Services 專案」。第二部份則針對服務伺服器端程式使用 Axis 套件，提出兩種建置方法。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。4.2 節「JWS 建置簡單 Web Services 伺服器端程式」以服務類別的 Java 檔案為基礎，透過 Axis 套件特有的 JWS 迅速部署(Instant deploy)方式，在 Axis Servlet 內自動產生服務伺服器端程式。而 4.3 節「WSDL-Oriented 建置 Web Services 伺服器端程式」則以 WSDL 文件為基礎，透過 WSDD 一般部署(custom deploy)方式，建置服務伺服器端程式。在各廠商支援 Web Services 的套件之間，服務需求端與服務伺服器端的建置方法可以互相通用；唯有 JWS 建置簡單 Web Services 伺服器端程式的方法僅限於 Axis 套件中才能使用。

### 4.1 使用 Axis 實作網路服務專案

本節分為三部份，第一部份描述海鱺生產供應鏈追溯與追蹤系統的架構，第二部份說明使用網路服務實作此系統時使用的軟體套件以及其特性，第三部份說明本專案提供的服務內容架構。

#### 4.1.1 海鱺生產供應鏈追溯與追蹤系統架構設計

近年來國際間相繼發生禽流感、狂牛病等疫病，且部分農產品被檢驗出藥物殘留過量，以及基因改造食品或黑心食品充斥，一再引發生產者、流通業者以及消費者的恐慌與不信任，食品安全衛生問題已成為產銷的最大議題。歐盟與日本等先進國家為了確保消費安全，已實施食品可追溯性制定，又稱為食品產銷履歷制度。

2.2 節已經說明目前水產品供應鏈運作情況，但是現有的資訊系統[3][5]有兩項缺失：生產者配合度不佳以及無法讓供應鏈中各層級使用追溯與追蹤的功能。以生產者配合度不佳的問題而言，主要是因為優良養殖場系統採用三層式架構，而資料庫為中央集中式資料庫，全國參加優良養殖場系統的生產者都必須透過操作介面將生產紀錄登記在此資料庫。但是各生產者之間處於同產業競爭狀態，因此會造成對於生產紀錄的資料洩漏或是被其他競爭對手取得的疑慮；另一方面，優良養殖場系統屬於政府漁業署監管範圍，生產業者對於所有紀錄資料完全處於政府相關單位監管下心生畏懼。在此疑慮之下會造成兩個結果：生產者不願意使用優良養殖場系統或是生產者登記不實紀錄。而對於無法讓供應鏈中各層級使用追溯與追蹤系統而言，由於優良養殖場系統僅限於生產業者與政府監管單位能夠進入，因此只有這兩種身分的使用者可以使用該系統內的資訊，上游原物料供應商或是下游的通路銷售廠商與消費者均無法取得相關資訊，因此對整條供應鏈而言並沒有追溯或追蹤的功能。

針對以上兩項問題的解決方法，本研究設計獨立分散式的資料庫系統與應用程式讓生產業者追蹤記錄生產履歷資料，同時搭配 Web Services 架構達成分散式的追蹤功能，其架構如圖 4.1 所示。



圖 4.1 追溯與追蹤系統實作架構[16]

此架構主要分成從供應鏈上游往下游的追蹤系統以及從供應鏈下游往上游的追溯系統。對於追蹤系統而言，由分散式的資料庫與農場工作日志組成，可獨立安裝在個別生產業者處，生產業者可以建立生產履歷紀錄於各自的資料庫內，沒有機密生產技術外流或是被監控的風險。對於追溯系統而言，主要利用 Web Services 的技術，提供服務需求者相關的資料服務，Web Services 技術架構主要分成服務需求端、服務伺服器端與服務仲介端三者，如圖 4.2 所示。而資料服務的內涵則是使用追蹤系統的農場工作日志資料庫，但是對於紀錄內容的開放程度則由生產業者自行在服務中定義。如此，生產業者不用擔心生產技術的資料安全問題，而供應鏈上各層級均能夠使用 Web Services 取得生產相關資訊。關於追蹤系統的農場工作日志應用程式與資料庫架構在第五章說明。

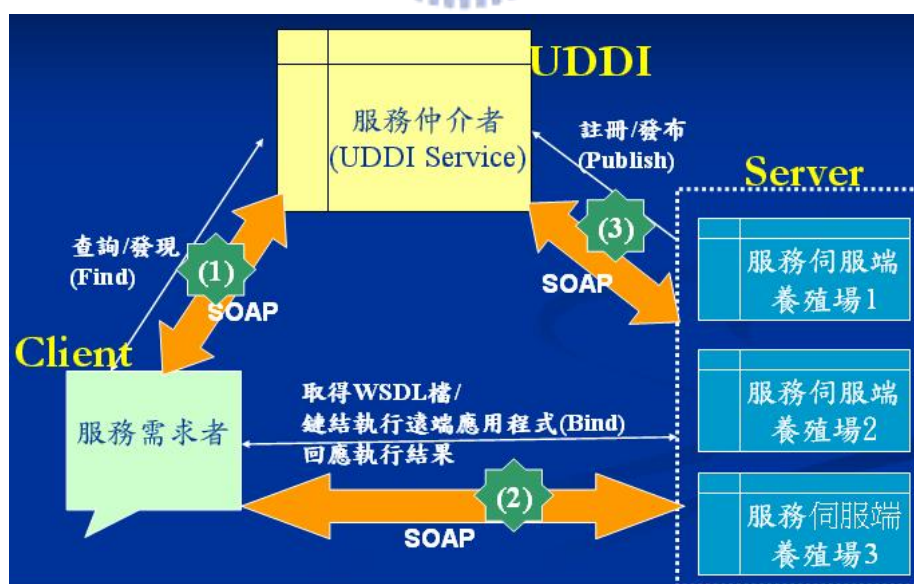


圖 4.2 Web Services 技術架構圖[16]

Web Services 技術的流程有依序為服務伺服器端向服務仲介端註冊服務、服務需求端向服務仲介端查詢服務位址、服務需求端向服務伺服器端取得服務內容三項。關於服務需求端向服務



伺服器端取得服務內容將在本章 4.2 節與 4.3 節進行說明。而對於服務伺服器端向服務仲介端註冊服務以及服務需求端向服務仲介端查詢服務位址兩者則在第四章說明。

### 4.1.2 Axis 的運作模式

本節介紹本專案實作使用的套件Axis(Apache Extensible Interaction System)[35]，其組成元件的結構如圖 4.3 所示。Axis使用的核心技術為JAX-RPC API，可以支援遠端程序呼叫(RPC，Remote Procedure Call)。而SOAP是W3C所規範使用XML格式的協定機制，可以將遠端程序呼叫放置在其文件區塊中。由於在程式中處理SOAP格式是一件複雜的工作，程式開發人員透過使用JAX-RPC API的程式碼，可以不必自行撰寫程式碼解析SOAP格式，而依然可以操作遠端程序呼叫。透過SOAP的定義，可以用XML的格式描述遠端程序呼叫的內容，使呼叫的過程以XML方式進行。不同語言與技術實作方式的遠端程序呼叫間之差異被XML格式所包裝，在異質的系統之間也可以互相溝通，因此可以實現跨平台的分散式計算，整合相異語言與技術所實作的系統。

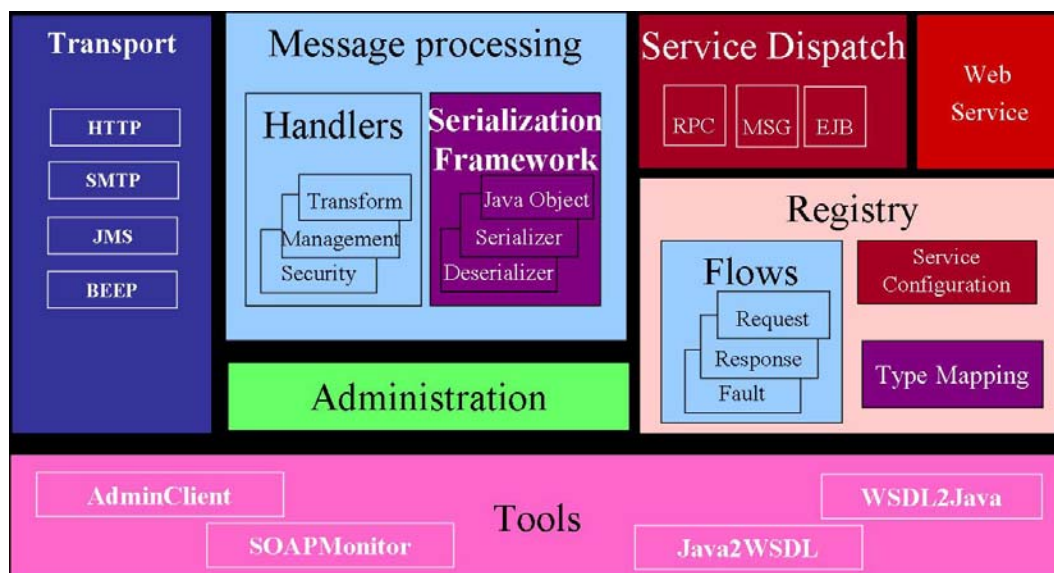


圖 4.3 Axis 組成元件架構圖[35]

Axis 是 Apache 組織實現 SOAP 技術的第三代專案，本質上是建立在 JAX-RPC 基礎架構上的 SOAP 引擎，開發語言為 Java 或是 C++，本專案使用版本為 2005 年 6 月 15 日該組織所釋出的 1.2.1 Final 版本。Axis 的優勢特點是速度、彈性、穩定度、元件導向的佈署方式、傳輸框架以及對 WSDL 文件的支援性。對速度而言，Axis 內部使用 SAX 技術進行解析，達到比早期 Apache SOAP 版本更高的速度；而對彈性而言，是指 Axis 對於程式開發人員是一個完全開放的自由架構，可以自行擴充自製的處理程序或是管理系統；對穩定度而言，是指 Axis 定義了一套公開介面，此介面的部份幾乎不會改變，相對而言較穩定；而對元件導向的佈署方式而言，Axis 內部的處理器(Handler)使所有導向功能的元件以處理器的方式很容易地組合與重覆使用；對傳輸框架而言，是指 Axis 的傳輸部分是獨立的抽象層，可以和 Axis 核心引擎完全分離；對 WSDL 文件的支援性而言，Axis 不僅支援 SOAP 標準，還附有 WSDL 相關支援工具，可以快速開發服務伺服器端以及服務需求端程式，並且可以在服務伺服器端根據部署

內容自動產生 WSDL 文件。

Axis 是一個開放的系統，提供開發 Web Services 的架構。由於 Axis 套件的介面是透過 Web 方式存取，因此 Axis 必須依附在 Web 伺服器之下，透過 Web 伺服器處理其介面才能正常運作，Axis 可以廣泛與各種廠牌的 Web 伺服器結合使用，本專案使用 Jakarta Tomcat 4.1 版的 Web 伺服器軟體與 Axis 搭配。Axis 不僅是伺服器端的工具，也可以作為服務需求者端的實作工具，當 Axis 用於服務需求者端時，可以存取其他符合 SOAP 協定的服務，圖 4.4 與 4.5 表示在這兩種模式下 Axis 元件之間的運作情況。Axis 的設計架構把伺服器端與需求端有效結合，在使用時可以分別以服務伺服器端與需求端執行。

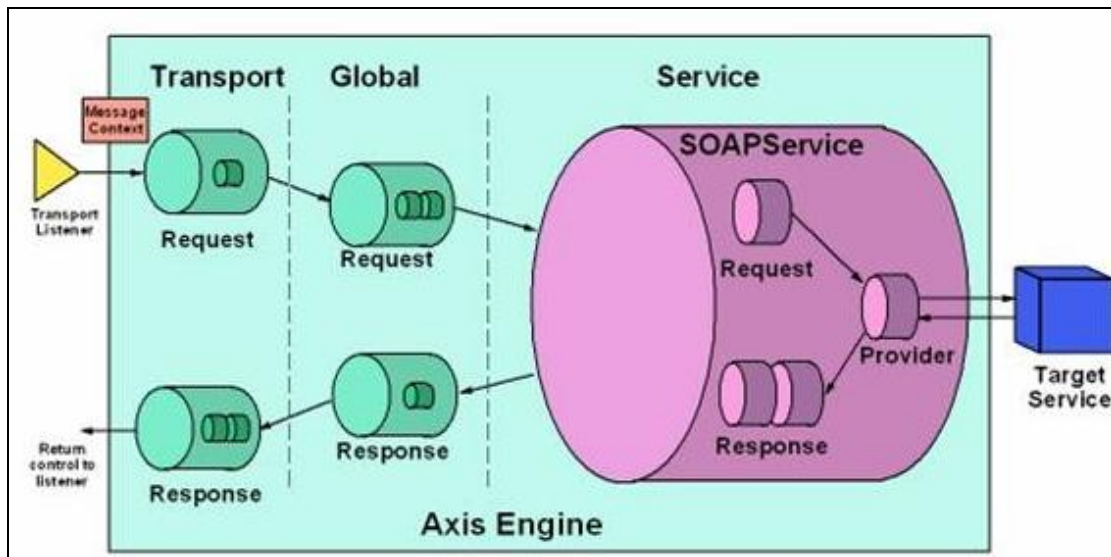


圖 4.4 Axis用於伺服器端的運作架構[35]

Axis 在伺服器端處理服務需求端呼叫時，會將呼叫請求的相關資訊封裝在訊息上下文 (MessageContext) 物件中，讓訊息上下文依次透過伺服器端的處理器。處理器被分為三個層次：傳輸層 (Transport)、基本層 (Global) 與服務層 (Service)。當使用者端的呼叫請求被轉換為訊息上下文並且依次通過三層處理器之後，服務提供者 (Provider) 負責呼叫服務物件，將傳回的結果按照相同路徑傳回最後被需求端接收。

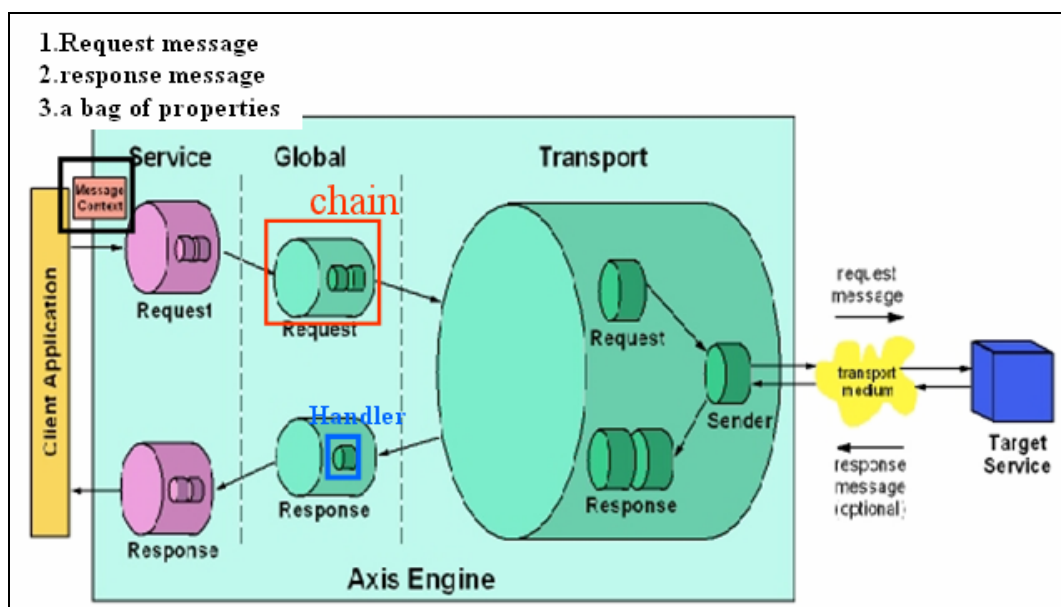


圖 4.5 Axis用於服務需求端的運作架構[35]

Axis 用於服務需求端時仍然透過與伺服器端相同的三層處理器的架構，相異之處是訊息上下文由服務需求端產生。接著依次透過服務層、基本層與傳輸層的處理器，最後將訊息發送到目標 Web Services 所在的伺服器。等待遠端伺服器接收到結果之後，在依循原本的路徑回傳給服務需求端程式。

#### 4.1.3 網路服務內容架構

本節說明實作的網路服務內容架構，而 4.2 與 4.3 節將會使用本節所描述的服務內容加上與服務需求端的溝通介面以及程式部署，完成服務需求端向服務伺服器端取得服務內容。Web Services 服務內容分為三部份：供應鏈中追溯與追蹤的唯一識別碼定義、服務程式與資料庫的連線建立以及工作日誌資料庫資訊在追溯系統的提供。

養殖水產供應鏈從上游到下游包括飼料與藥物供應、種魚、孵化、養殖、宰殺、加工、批發與零售。此供應鏈中無論是交易過程中對產品的追蹤或是交易完成後的追溯，都必須透過記錄過的識別證明，以資訊技術的運用達成追蹤產品在供應鏈中所在位置、使用情況、原始出處以及處理過程等資訊。歐盟在 2002 年 1 月 1 日的 EC 2065/2001 命令中規定採用 EAN.UCC 系統作為歐盟國家漁品貼標管理與追蹤的工具。條碼標籤上需有來源國、加工製出處批號、品名、數量、淨重、生產方式、漁勞區、日期、漁船、飼養地區、保存期限等資料。並於 2005 年 1 月 1 日起實施食品訊息追蹤紀錄，若無法提供食品生產履歷，將無法進入歐盟市場。

EAN.UCC 的識別標準原本就是追溯與追蹤系統使用的通用語言，且其標準受到國際組織法律規章的採納。使用 EAN.UCC 標準化識別代碼，可以克服因為個別公司、部門、國家或是不同語言的限制所引發的資訊溝通障礙，在本專案中則可以克服不同生產業者間的識別與通訊障礙。圖 4.6 使用 EAN.UCC 標準下的供應商識別碼/全球位址碼(Global Location Number, GLN)、全球商品碼(Global Trade Item Number, GTIN)以及配送包裝識別碼(Serial Shipping Container Code, SSCC)作為養殖水產供應鏈的唯一識別系統。

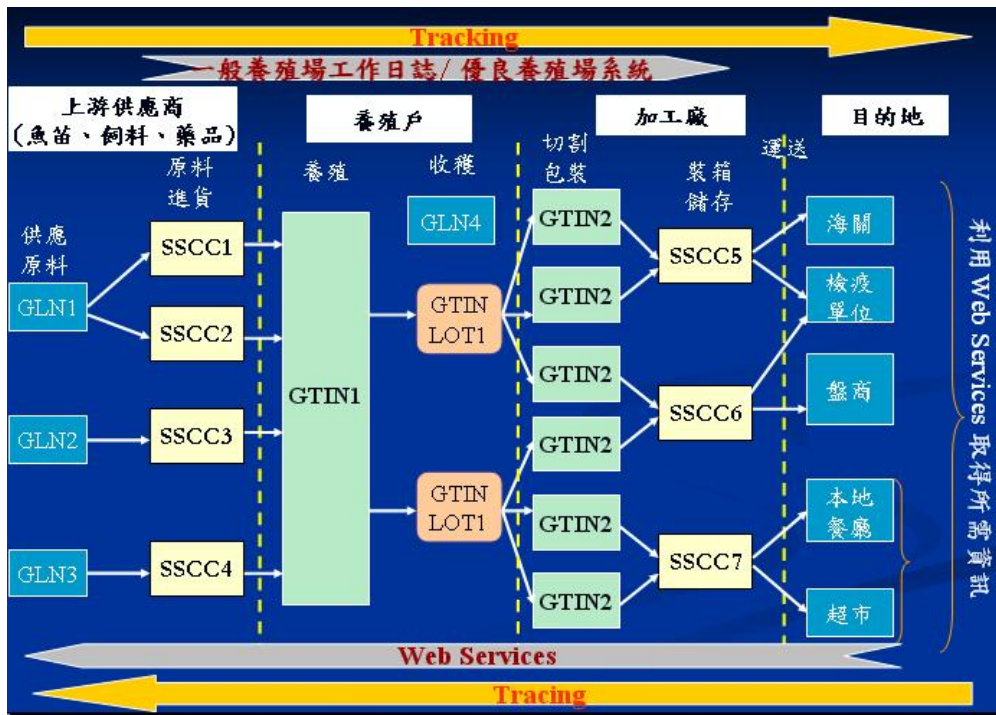


圖 4.6 EAN.UCC系統使用於養殖水產供應鏈[16]





## 4.2 JWS(Java Web Service)建置簡單網路服務伺服器端程式

在眾多提供網路服務的套件中，JWS的建置方法是Axis獨有的。其優點在於使用方便，能夠用最簡便的方式以最快的速度建立網路服務。但是缺點在於這是一種不正式的網路服務建置方法，也只能在Axis的套件支援下運作，無法達到Web Services分散式、跨平台的特性。此外，因為缺少正式的服務部署描述，因此JWS方法只能提供單一的服務，若服務數目超過一個則此方法並不適用。本節主要分成兩部份，第一部份說明服務類別的設計；而第二部份則說明服務部署的方式。關於JWS方法的開發流程如圖 4.7 所示[24]。

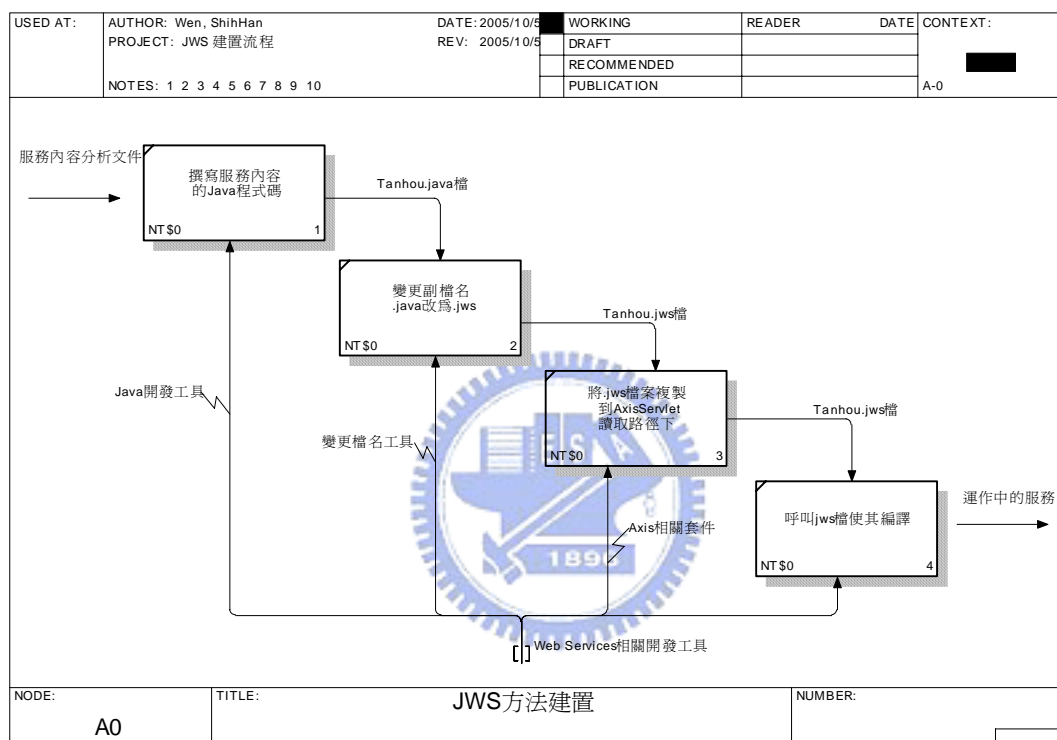


圖 4.7 JWS 方法建置流程 IDEF0

### 4.2.1 伺服器端服務類別設計

服務程式以Java撰寫，而農場工作日志資料庫則使用Microsoft SQL Server 2000。服務程式與資料庫的連線建立，必須使用JDBC API達成。JDBC(Java Database Connectivity)可以讓程式開發人員建構SQL敘述，並且將其嵌入Java API的呼叫中，因此JDBC可以使資料庫與Java應用程式之間的資訊互相傳遞，如圖 4.8 所示。

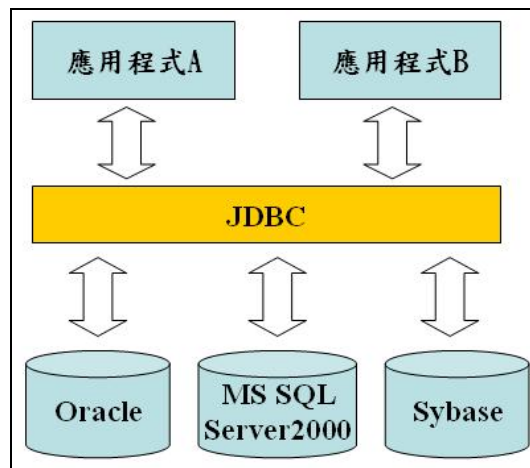


圖 4.8 JDBC 架構圖

JDBC是透過一組Java介面來達成其目標，而每一種Java介面可由不同廠商實作而成。針對特定的資料庫引擎，實作JDBC介面的一組類別，稱為JDBC驅動程式。建立資料庫程式時只要先建立JDBC驅動程式，就可以對資料庫連線進行資料庫的操作。目前JDBC種類可分為四類：Type1、Type2、Type3與Type4。Type1是JDBC-ODBC橋接驅動程式，這種型態的JDBC並沒有和資料庫直接溝通，而是先由JDBC和ODBC做溝通，再由ODBC和資料庫溝通。這一類的JDBC效能較差，而且因為與ODBC配合，所以只能在Windows作業系統下運作。Type2是原生API驅動程式，會將來自JDBC的要求轉換成資料庫所指定的原生碼，再與資料庫溝通，效能比Type1快，但如果原生碼出錯，可能會造成Server的不穩定。Type3則是JDBC透過網路的純Java驅動程式，這類驅動程式透過網路協定與中介軟體溝通，JDBC對資料庫的要求與回應都透過中介軟體來處理，此類型驅動程式完全由Java撰寫，可攜性高。Type4是原生協定及純Java驅動程式，這類的驅動程式完全由Java撰寫，而且直接與資料庫做溝通，因此效能及安全性在四種JDBC類型中最高。

本研究使用全球最大自由軟體公開平臺SourceForge.net中jTDS專案的JDBC API，版本為2005年6月16日釋出的jTDS 1.1版[32]，該程式支援Microsoft SQL Server以及Sybase兩個廠牌資料庫，而且屬於效能與安全性最高的Type4。

服務架構是將工作日誌資料庫的資料表以服務為導向重新編整，描述工作日誌資料庫在追溯系統提供的資訊內容。由於供應鏈的端點以及服務數量很多，以下將針對養殖場為例，說明廠商使用網路服務建構生產履歷追溯系統的過程。根據服務導向架構，工作日誌資料庫內容劃分為五項服務，分別是基本資料服務、檢驗表單內容服務、飼料與餵養紀錄服務、藥品紀錄服務。

基本資料服務根據出貨批號為主索引鍵值，出貨批號是根據全球商品碼規則編碼，可以取得該批號養殖漁產品的名稱、重量、包裝、所在養殖場以及下游的販運商等資訊，如圖 4.9 所示。

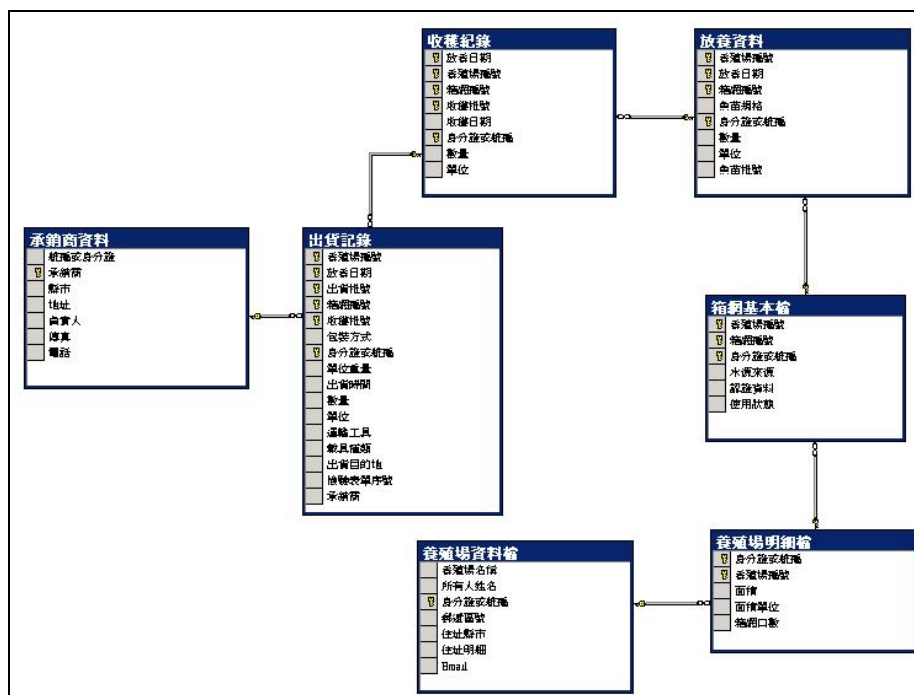


圖 4.9 基本資料服務

檢驗表單內容服務也根據出貨批號為主索引鍵值，取得檢驗表單序號，再利用檢驗表單序號讀取該檢驗表單相關檢驗項目與檢驗值，該服務相關的資料表如圖 4.10 所示。

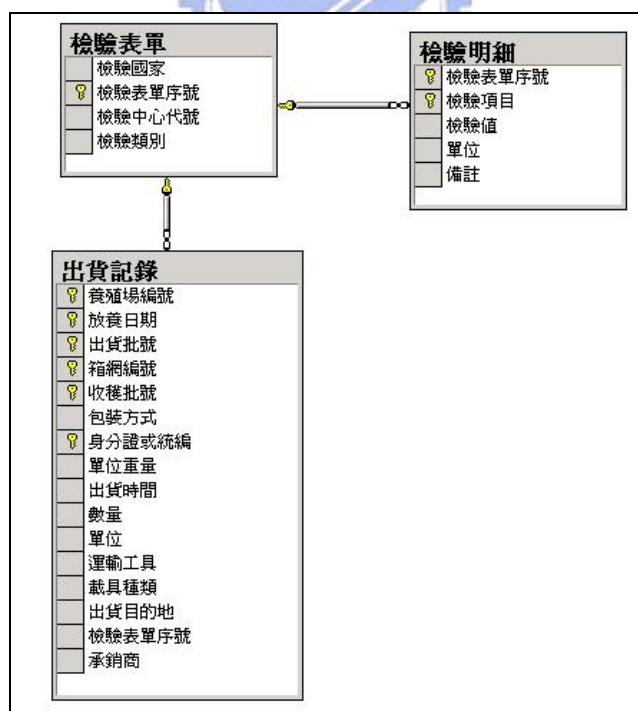


圖 4.10 檢驗表單內容服務

飼料與餵養紀錄服務利用出貨紀錄資料表的出貨批號取得養殖場編號、放養日期、箱網編號、身分證或統編四個欄位資訊，並藉由這四個欄位取得相關的餵養紀錄、飼料資訊以及飼料供應商資訊。該服務使用資料表如圖 4.11 所示。

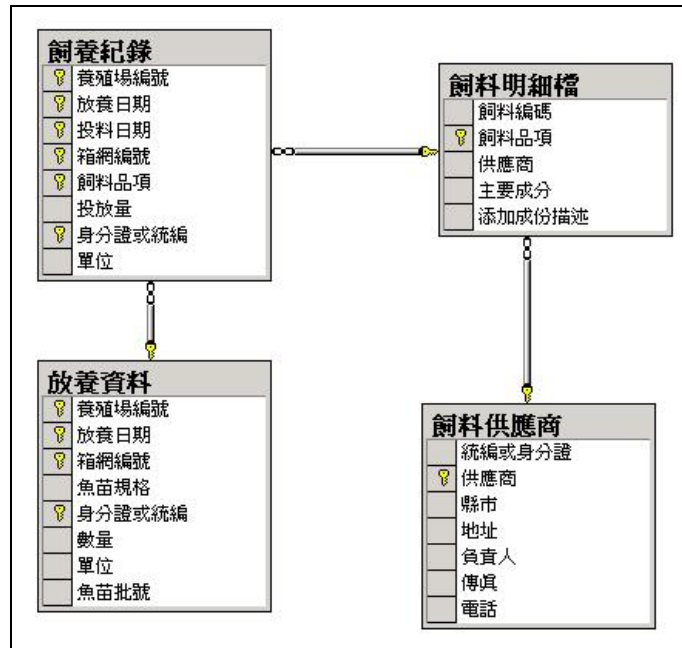


圖 4.11 飼料與餵養紀錄服務

藥品紀錄服務與飼料與餵養紀錄服務同樣利用出貨紀錄資料表的出貨批號取得放養資料表的四個主索引鍵欄位：養殖場編號、放養日期、箱網編號、身分證或統編。再藉由這四個主索引鍵欄位取得相關的用藥紀錄、藥品資訊與藥品供應商資訊，該服務相關的資料表如圖 4.12 所示。

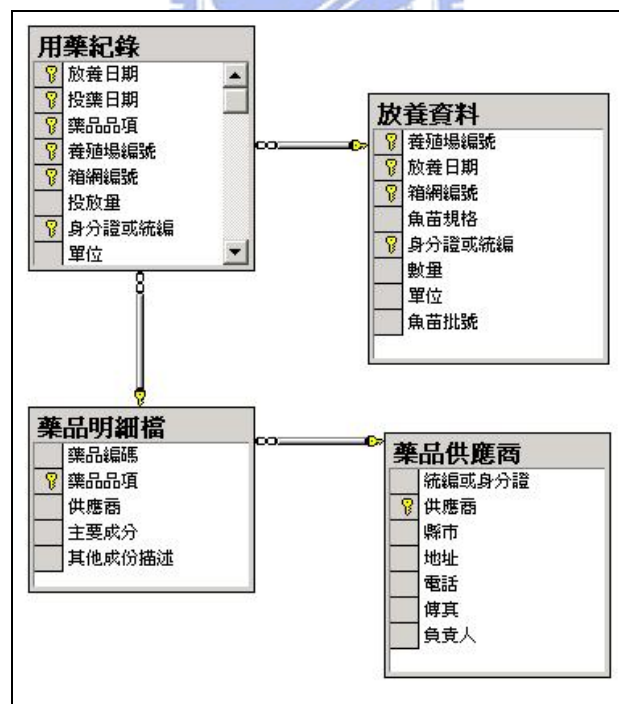


圖 4.12 藥品紀錄服務

## 4.2.2 在 Axis Server 部署服務伺服端程式

完成 4.2.1 節的服務類別之後，將此類別檔 Tanhou.javatj 重新命名為 Tanhou.jws，然後把



此檔案複製到 Jakarta Tomcat Web 伺服器目錄之下的\webapps\axis\，此時服務需求端便可以存取此服務。在服務需求端發出呼叫請求時，Axis 會自動讀取 Tanhou.jws 檔案並且編譯，然後將結果轉換為 SOAP 協定送回到服務需求端。圖 4.13 是直接向伺服器端發出呼叫請求，使用 Axis 內部的工具 TCP Monitor 進行網路封包的監聽，驗證服務已經正常運作。若服務並沒有正常運作，TCP Monitor 的監聽視窗不會有任何內容。

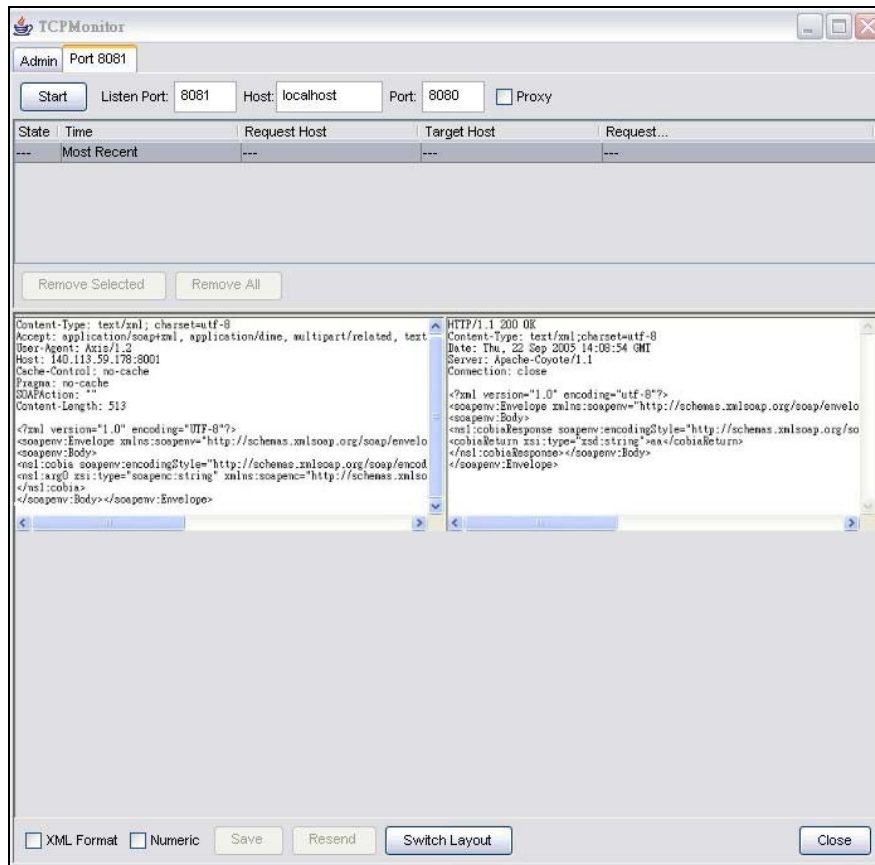


圖 4.13 TCP Monitor 驗證服務運作狀況

此方法建置 Web Services 的速度很快，耗費成本最小。但是此方法為 Axis 所單獨提供，並不適用於其他 Web Services 的套件軟體。此外，在 Axis 官方使用手冊上特別註明，JWS 方法在服務類別中不能使用 package，且因為被呼叫請求時才進行編譯，因此對於程式開發人員而言很難對其進行偵錯，一般正式運作的 Web Services 不會採用此方法建置，而僅適用於簡易的 Web Services。

### 4.3 WSDL-Oriented 建置網路服務伺服器程式

WSDL-Oriented建置方法是以WSDL文件的設計為起點，透過WSDL2Java工具的轉換，自動產生一套與WSDL文件各節點互相對應的Java類別、方法與成員。將伺服器服務程式碼加入自動產生的類別檔之中，最後利用自動產生的wsdd檔[6]進行伺服器程式的佈署。本方法的發展過程如圖 4.14 所示。

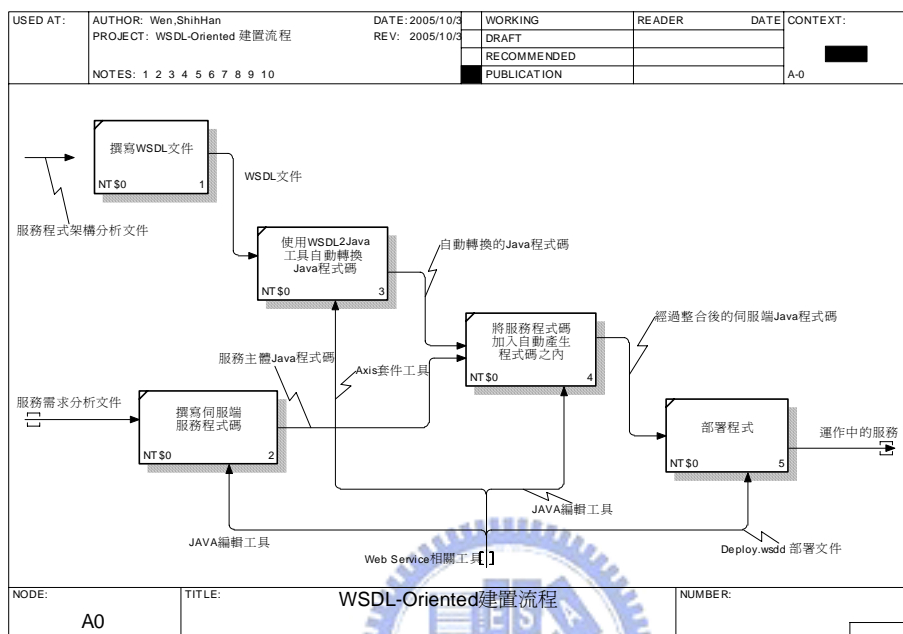


圖 4.14 WSDL-Oriented 建置流程

#### 4.3.1 WSDL 文件設計

Web Services 是服務導向的軟體架構，而 WSDL 是 Web Services 的描述語言。透過 WSDL 的技術架構，Web Services 不但可以被描述，還能夠透過描述獲取更多的使用細節、透過工具建立使用者端程式碼，或是直接形成遠端呼叫。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以及服務需求端的交換方式，表 4.1 定義出本研究設計 WSDL 文件時必要的節點資訊，同時該節點資訊將標示於圖 4.15 中。本節使用 Altova XMLSpy 軟體以及 Tanhou.wsdl 為例說明 WSDL 文件的編撰與結構分析。

表 4.1 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 4.15 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 cobiaResponse 與 cobiaRequest	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

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <wsdl:definitions xmlns:apache:soap="http://xml.apache.org/xml-soap" xmlns:impl="http://cobia.nctu.edu.tw:8080/axis/services/Tanhou" xmlns:intf="
   http://cobia.nctu.edu.tw:8080/axis/services/Tanhou" 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
   targetNamespace="http://cobia.nctu.edu.tw:8080/axis/services/Tanhou">
3      <wsdl:message name="CobiaResponse">
4          <wsdl:part name="cobiaReturn" type="xsd:string"/>
5      </wsdl:message>
6      <wsdl:message name="CobiaRequest">
7          <wsdl:part name="in0" type="xsd:string"/>
8      </wsdl:message>
9      <wsdl:portType name="Tanhou">
10         <wsdl:operation name="cobia" parameterOrder="in0">
11             <wsdl:input name="cobiaRequest" message="impl:CobiaRequest"/>
12             <wsdl:output name="cobiaResponse" message="impl:CobiaResponse"/>
13         </wsdl:operation>
14     </wsdl:portType>
15     <wsdl:binding name="TanhouSoapBinding" type="impl:Tanhou">
16         <wsdl:soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
17         <wsdl:operation name="cobia">
18             <wsdl:soap:operation/>
19             <wsdl:input>
20                 <wsdl:soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace"/>
21             </wsdl:input>
22             <wsdl:output>
23                 <wsdl:soap:body use="encoded" encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="
   http://cobia.nctu.edu.tw:8080/axis/services/Tanhou"/>
24             </wsdl:output>
25         </wsdl:operation>
26     </wsdl:binding>
27     <wsdl:service name="TanhouService">
28         <wsdl:port name="Tanhou" binding="impl:TanhouSoapBinding">
29             <wsdl:soap:address location="http://cobia.nctu.edu.tw:8080/axis/services/Tanhou"/>
30         </wsdl:port>
31     </wsdl:service>
32     <!-- WSDL created by Apache Axis version: 1.2
33     Built on May 03, 2005 (02:20:24 EDT)-->
34 </wsdl:definitions>
35

```

圖 4.15 Tanhou.wsdl 文件以 Altova XMLSpy 展開[23]

根據圖 4.15 的 Tanhou.wsdl 文件，架構大致可以分為 definitions、message、portType、binding、service 等五項元素，WSDL 文件綱要如圖 4.16 所示。以下使用 Tanhou.wsdl 針對這五項元素分別描述其內容和功能。

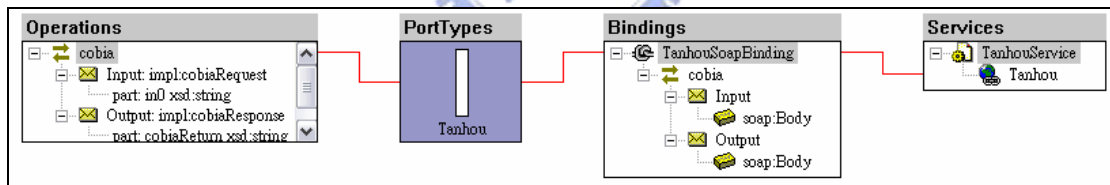


圖 4.16 Tanhou.wsdl 文件綱要

definitions 為 WSDL 文件的根元素，因此 WSDL 文件中的內容都被包含在此節點內。definitions 定義了 WSDL 的命名空間以及存放位置，如圖 4.17 所示。

```

XML
= version 1.0
= encoding UTF-8
wsdl:definitions
= xmlns:apache:soap http://xml.apache.org/xml-soap
= xmlns:impl http://cobia.nctu.edu.tw:8080/axis/services/Tanhou
= xmlns:intf http://cobia.nctu.edu.tw:8080/axis/services/Tanhou
= 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
= targetNamespace http://cobia.nctu.edu.tw:8080/axis/services/Tanhou
- wsdl:message name=cobiaResponse
- wsdl:message name=cobiaRequest
- wsdl:portType name=Tanhou
- wsdl:binding name=TanhouSoapBinding type=impl:Tanhou
- wsdl:service name=TanhouService
- Comment WSDL created by Apache Axis version: 1.2
  Built on May 03, 2005 (02:20:24 EDT)

```

圖 4.17 WSDL的definitions元素[23]

message 元素用來定義服務提供者與服務需求者兩端點之間的資訊交換，包括由提供者發送至服務需求者的輸出變數 `cobiaResponse` 以及由服務需求者發送至提供者的輸入變數 `cobiaRequest`。message 元素中可以使用 `wsdl:part` 子元素來描述訊息參數，例如 `cobiaResponse` 的 `part` 子元素定義了 `cobiaReturn` 的字串型態變數；而 `cobiaRequest` 的 `wsdl:part` 子元素則定義了 `in0` 的字串型態變數。message 元素的詳細規格如圖 4.18 所示。

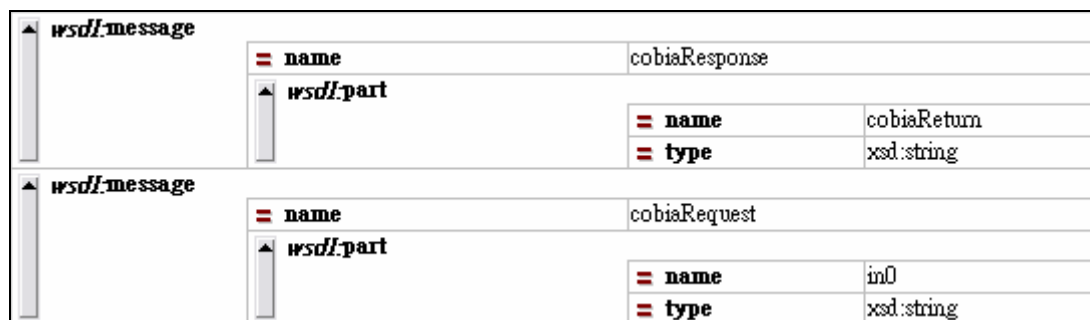


圖 4.18 WSDL的message元素[23]

portType 定義了服務的存取入口，利用子元素 `wsdl:operation` 描述服務所提供的操作，每個操作由進出的訊息所組成。由圖 4.19 的規格分析，`wsdl:operation` 子元素命名為 `cobia`，而 `parameterOrder` 定義了 `cobia` 函數內的變數名稱 `in0`。`wsdl:operation` 內部的子元素 `wsdl:input` 則定義了 message 元素中已經宣告的輸入變數 `cobiaRequest`；而 `wsdl:output` 則定義了 message 元素中已經宣告的輸出變數 `cobiaResponse`。

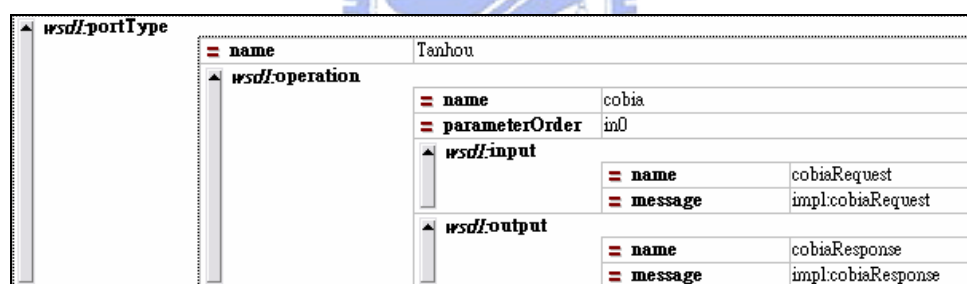


圖 4.19 WSDL的portType元素[23]

binding 元素定義了與服務實作相關的資訊。`wsdlsoap:binding` 子元素內宣告了服務的通訊協定使用遠端程序呼叫(Remote Procedure Call, RPC)，在此元素中提供了 RPC、Document、Wrapped、Messages 四種服務協定。其中 RPC 服務協定支援 SOAP 協定且支援 SOAP 格式的編碼；而 Document 服務協定無法支援任何格式的編碼；Wrapped 服務協定與 Document 服務協定大致相同，差別在於 Wrapped 服務協定會將 SOAP body 切割為許多各別的參數，而 Document 服務協定則是視其為單一結構體；Message 服務協定只能單純使用 SOAP 格式的 XML 文件來傳輸資料，沒有其他三種服務協定所提供的資料繫結(binding)與對應(mapping)。由於 `Tanhou.wsdl` 文件採用 RPC 服務協定，所以另一子元素 `wsdl:operation` 中則定義與 portType 相關的資訊，包括 `wsdl:input` 以及 `wsdl:output` 的編碼使用狀況以及對應的命名空間。binding 的詳細規格如圖 4.20 所示。



wSDL-binding	
name	TanhouSoapBinding
type	impl:Tanhou
wSDLsoap-binding	
style	rpc
transport	http://schemas.xmlsoap.org/soap/http
wSDL-operation	
name	cobia
wSDLsoap-opera...	
wSDL-input	
wSDLsoap-body	
use	encoded
encodingStyle	http://schemas.xmlsoap.org/soap/encoding/
namespace	http://DefaultNamespace
wSDL-output	
wSDLsoap-body	
use	encoded
encodingStyle	http://schemas.xmlsoap.org/soap/encoding/
namespace	http://cobia.nctu.edu.tw:8080/axis/services/Tanhou

圖 4.20 WSDL的binding元素[23]

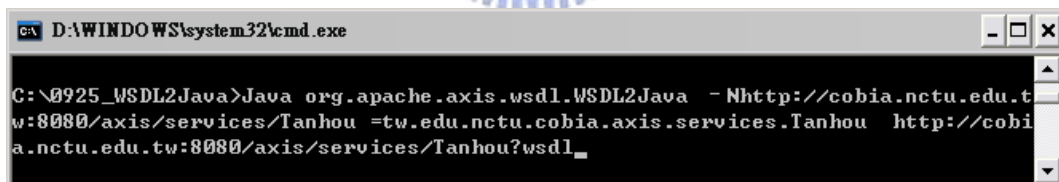
service 元素用於指定服務與 binding 的繫結關係，在子元素 wdl:port 中，指明了服務所在位址是 http://cobia.nctu.edu.tw:8080/axis/services/Tanhou。service 的詳細規格如圖 4.21 所示。

wSDL-service	
name	TanhouService
wSDL-port	
name	Tanhou
binding	impl:TanhouSoapBinding
wSDLsoap-address	
1. http://cobia.nctu.edu.tw:8080/axis/services/Tanhou	

圖 4.21 WSDL的service元素[23]

### 4.3.2 使用 WSDL2Java 產生服務類別程式碼

取得 4.3.1 設計好的 WSDL 文件之後，可以使用 Axis 提供的 org.apache.axis.wsdl.WSDL2Java 工具將 WSDL 文件自動產生相對應服務的 Java 程式碼，其指令與參數如圖 4.22 所示。



```

C:\D:\WINDOWS\system32\cmd.exe
C:\>java org.apache.axis.wsdl.WSDL2Java -Nhttp://cobia.nctu.edu.tw:8080/axis/services/Tanhou =tw.edu.nctu.cobia.axis.services.Tanhou http://cobia.nctu.edu.tw:8080/axis/services/Tanhou?wsdl

```

圖 4.22 org.apache.axis.wsdl.WSDL2Java 工具執行命令

在圖 4.1 的命令之中，-Nhttp://cobia.nctu.edu.tw:8080/axis/services/Tanhou = tw.edu.nctu.cobia.Tanhou 指定將 WSDL 檔的命名空間 http://cobia.nctu.edu.tw:8080/axis/services/Tanhou 對應到 Java 程式碼的套件路徑 tw.edu.nctu.cobia.axis.services.Tanhou。而另一項參數 http://cobia.nctu.edu.tw:8080/axis/services/Tanhou?wsdl 則指定了 WSDL 文件所在位址。Axis 根據 JAX-RPC 規範進行 Java 程式碼與 WSDL 文件的繫結。Axis 針對 WSDL 文件與伺服端的 Java 程式碼的繫結有整套的對應規則如表 4.2 所示。

表 4.2 WSDL與Java程式碼繫結規則[35]

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

根據繫結規則，指令執行完成後有六個 Java 程式檔以及兩個 wsdd 檔被建立在 `\tw\edu\nctu\cobia\axis\services\Tanhou` 的路徑之下，如圖 4.23 所示，分別是 `TanhouSoapBindingSkeleton.java`、`TanhouService.java`、`TanhouServiceLocator.java`、`TanhouSoapBindingImpl.java`、`TanhouServiceSoapBindingStub.java` 以及 `Tanhou.java` 六個 Java 檔與 `deploy.wsdd` 以及 `undeploy.wsdd` 兩個 wsdd 檔。關於六個 Java 程式的關係如圖 4.24 的類別圖所示。而兩個 wsdd 檔的內容與使用方式則在 4.3.3 節說明。

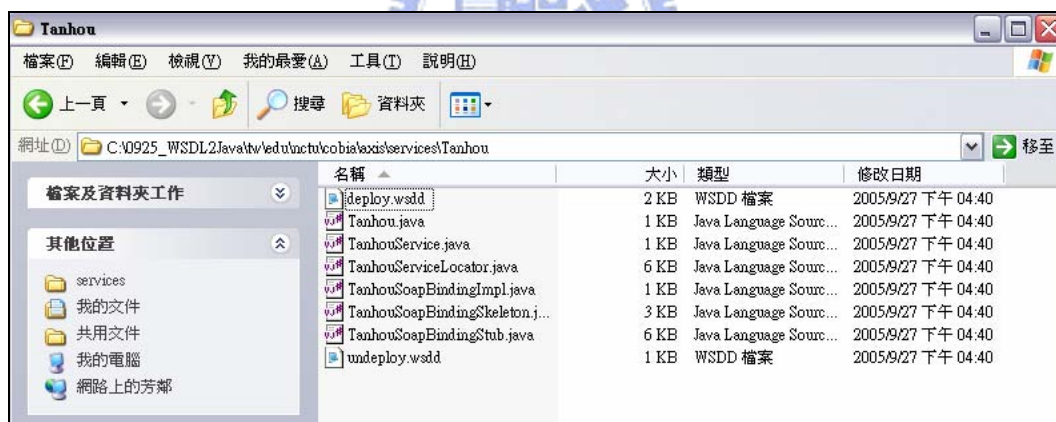


圖 4.23 WSDL2Java 工具執行完成結果

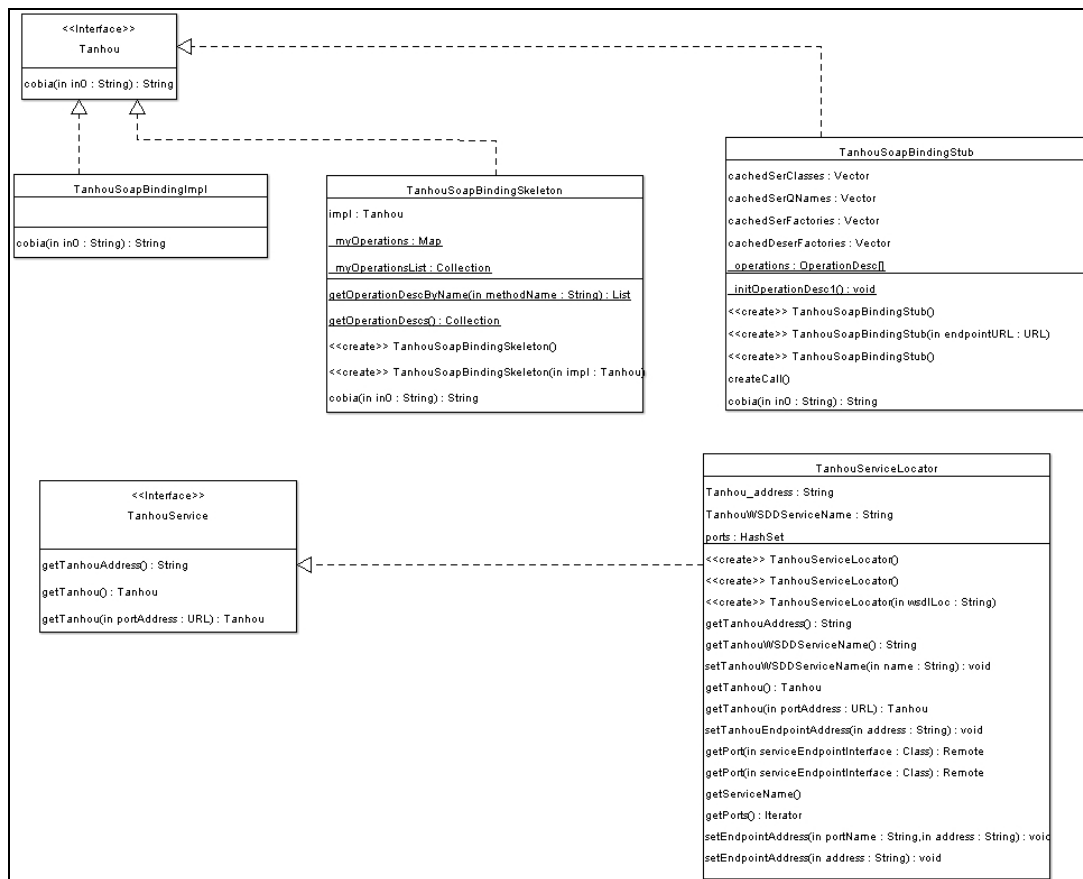


圖 4.24 使用 WSDL2Java 產生的程式碼類別圖

TanhouService 是一個介面，由 WSDL 檔的 service 部分所產生，而 service 的 name 屬性是 TanhouService。TanhouService 透過 getTanhou() 方法取得 Tanhou 物件內關於服務的內容。TanhouServiceLocator 是服務介面 TanhouService 的實作類別，由 WSDL 中的 service 部分所產生。該類別中包含服務位址資訊的存取，因此透過此類別的 getTanhouAddress() 方法能夠取得服務的呼叫位址等資訊，在呼叫的過程中將位址資訊傳遞給 TanhouServiceSoapBindingStub 內部提供的呼叫方法做為參數，完成對遠端服務的呼叫。Tanhou 則是由 WSDL 檔的 portType 部分所產生的介面，portType 的 name 屬性是 Tanhou。Tanhou 根據 WSDL 文件定義了服務方法 cobia()。TanhouServiceSoapBindingStub 由 WSDL 檔的 binding 部分所產生，binding 的 name 屬性值是 TanhouServiceSoapBinding。此類別實作了服務介面 Tanhou，接收到服務位址資訊的參數之後，使用 createCall() 方法完成服務物件的呼叫。而 TanhouSoapBindingImpl 由 WSDL 中的 binding 部份產生，此類別實作服務介面 Tanhou。TanhouSoapBindingImpl 產生時為沒有服務內容的樣板(template)類別，在 4.1.3 節的服務內容程式碼必須加入到此類別檔覆寫 Tanhou 的服務方法 cobia() 內部才能定義服務內容。TanhouServiceSoapBindingSkeleton 實作了服務介面 Tanhou。該類別的功能位置介於 Axis 引擎以及服務內容之間，當 Axis 引擎呼叫服務的時候，TanhouServiceSoapBindingSkeleton 透過呼叫 TanhouSoapBindingImpl 的 cobia() 方法執行服務。

### 4.3.3 在 Axis Server 部署服務伺服器端程式

網路服務部署描述語言(Web Service Deployment Descriptor, WSDD)是Axis支援的部署Web Services方式，稱為一般部署[6]。在4.3.2節自動產生的deploy.wsdd中，className 參數指定使用服務的類別物件，allowMethod參數指定服務類別的服務方法。使用org.apache.axis.client.AdminClient工具執行deploy.wsdd以及其結果，如圖4.25所示。



```
C:\D:\WINDOWS\system32\cmd.exe
C:\0925_WSDL2Java\tw\edu\nctu\cobia\axis\services\Tanhou>java org.apache.axis.client.AdminClient deploy.wsdd
Processing file deploy.wsdd
<Admin>Done processing</Admin>
```

圖 4.25 執行 deploy.wsdd 以及部署成功結果

部署完成後，可以使用與4.2.2節相同的方法使用TCP Monitor封包監聽工具加以驗證服務是否正常運作。





# 第五章 履歷追蹤查詢系統架構- 服務需求端設計

本章說明服務需求端的實作架構，服務需求端的功能有三項：取得網路服務描述文件、解析網路服務描述文件、向服務伺服器端發出服務請求並接收回應。對於取得網路服務描述文件這項動作來說，必須先透過仲介端的查詢來得知網路服務描述文件所在的位址，將在 5.2 節與 5.3 節中予以說明；5.1 節則針對解析網路服務描述文件以及使用 Axis API 向服務伺服器端發出服務請求並接收回應此兩項工作，說明處理的流程架構。

## 5.1 網路服務需求端設計

使用 Axis 使用者端 API 設計服務需求端的開發方法，主要是針對較為簡易單純的服務，可以從網路服務描述語言的文件中直接得知呼叫服務的相關設定值。開發的流程分為四項：首先透過網際網路取得網路服務描述語言文件，關於網路服務描述語言的文件結構已在 4.x 節中說明；經過網路服務描述語言的開發工具檢視此文件後，解析服務伺服器端呼叫的名稱以及相關參數設定；接著利用 org.apache.axis.client.Call 類別所提供的方法，撰寫 Axis 的使用者端 API 的功能進行服務的呼叫與相關設定；最後在程式碼中加入對於服務伺服器端回應結果的處理方法。本方法的開發過程如圖 5.1 所示。

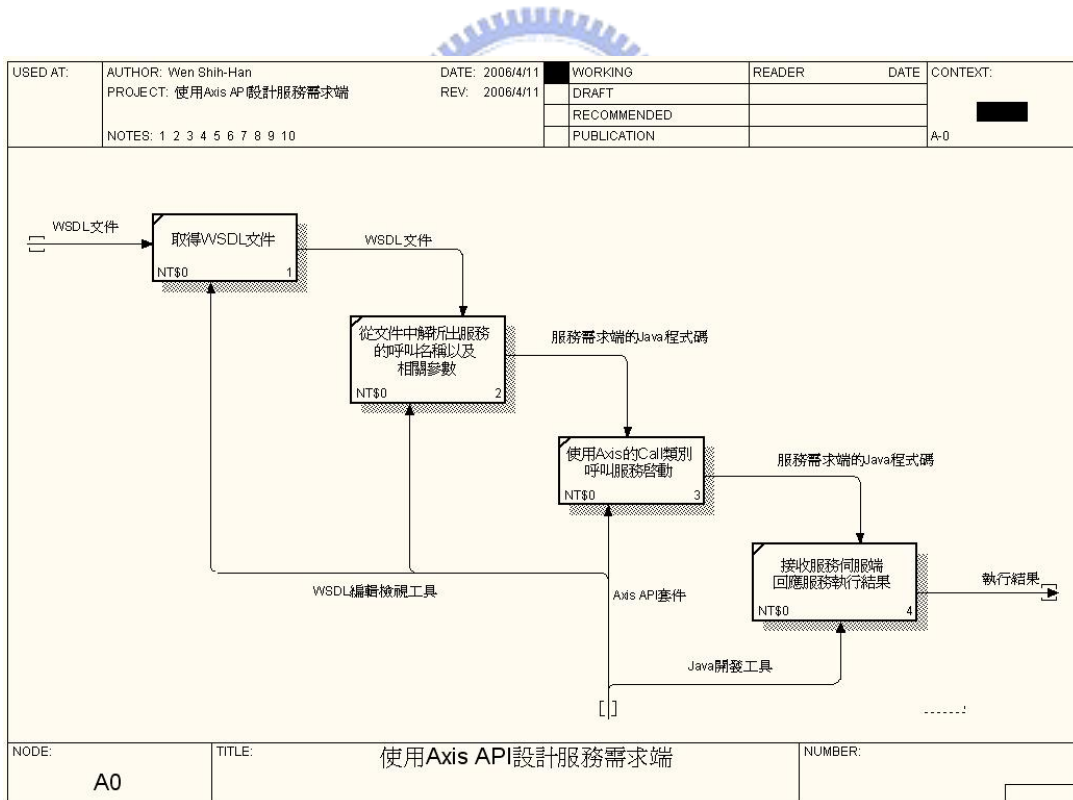


圖 5.1 使用 Axis API 套件設計服務需求端流程

圖 5.2 分別是使用 org.apache.axis.client.Call 類別所提供的方法來啟動服務以及接收服務伺服器端回應結果並處理的流程。啟動服務時主要有五項動作：產生服務物件、設定服務伺服器位址、設定服務存取的方法、設定相關參數以及設定回傳資料型別。在本研究實作的程式中首先使用 createCall()方法產生 call 物件；接著以 setTargetEndpointAddress()方法設定服務伺

服端所在位址為『http://140.113.59.179:8080/axis/services/MyService』；使用 setOperationName() 設定存取服務的方法名稱為『cobial』；設定參數名稱為『arg1』且其參數型態為『IN』；setReturnType()則設定回傳資料型態為『XSD\_STRING』的字串型態；當相關啟動服務的設定完成之後，使用 invoke()方法執行服務需求端的呼叫，並將回應的結果存在變數 ret 中繼續進行後續的回應結果處理程序。

```
try (  
    String endpoint = "http://140.113.59.178:8080/axis/services/MyService";  
    String number = jTextField1.getText().substring(13);  
    Service service = new Service();  
    Call call = (Call) service.createCall();  
    call.setTargetEndpointAddress( new java.net.URL(endpoint) );  
    call.setOperationName("cobial");  
    call.addParameter( "arg1", XMLType.XSD_STRING, ParameterMode.IN);  
    call.setReturnType( org.apache.axis.encoding.XMLType.XSD_STRING );  
  
    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
```

使用 Axis 使用者端的 Call 類別方法呼叫伺服器端服務啟動

接收服務伺服器端回應結果並加以處理

圖 5.2 開發服務需求端使用 Call 類別方法的程式碼



## 5.2 仲介端登錄、發佈與查詢服務 Web-based 的操作

網路服務仲介端的具體任務有三項：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找適當的服務位址。如圖 5.3 UDDI 技術架構所示，註冊機制必須由至少一個 UDDI 工作節點所組成，對內負責管理與營運註冊中心，對外則接受登錄、發佈、查詢等作業；此外還要與其他相容的註冊中心交換彼此的註冊資訊。UDDI 註冊中心是仲介網路服務的實作，藉由可展加註式語言提供服務需求端獲取所需服務的相關資訊。目前與 UDDI 註冊中心的溝通機制有兩種型式：Web-based 介面型態以及 Non-Web API 方式。Web-based 介面型態是將登錄、發佈與查詢等工作的細節操作透過網頁的方式呈現，例如 IBM UDDI Business Registry[33]；而 Non-Web API 方式則是透過 UDDI API 來撰寫登錄、發佈與查詢服務的操作應用程式，例如 UDDI4J。本節將說明 Web 介面型態的操作，而關於 UDDI API 的使用則在 5.3.2 節中說明。而關於登錄、發佈與查詢等工作處理的內容，是以結構化的註冊資訊為基礎，關於結構化的註冊資訊內容將在 5.3.1 節中說明。

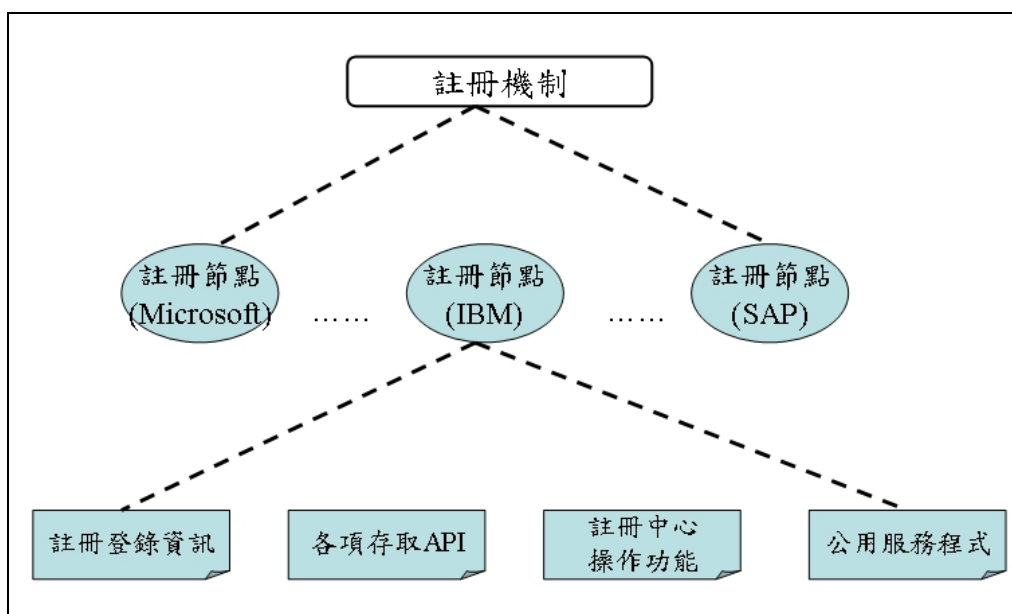


圖 5.3 UDDI 技術架構圖

本研究以 IBM UDDI Business Registry Web Site 作為網路服務仲介端。在網路服務仲介端三項任務中，接收商業實體的登錄註冊是最為單純的工作。其內容就是一組使用者帳號與密碼的註冊以及登錄，如圖 5.4 所示。

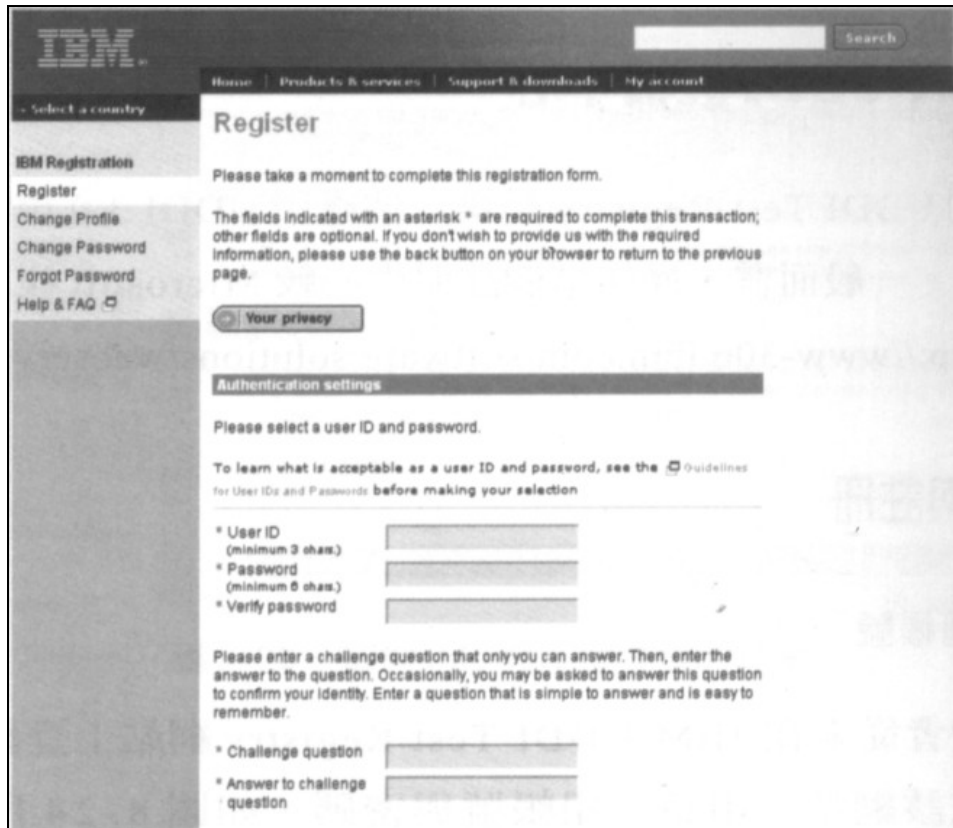


圖 5.4 商業實體的登錄註冊[33]

關於 UDDI Business Test Registry 的服務發佈操作流程如圖 5.5 所示，共有五項工作：新增商業實體、新增服務、新增存取點、新增技術模型實體以及檢視服務。

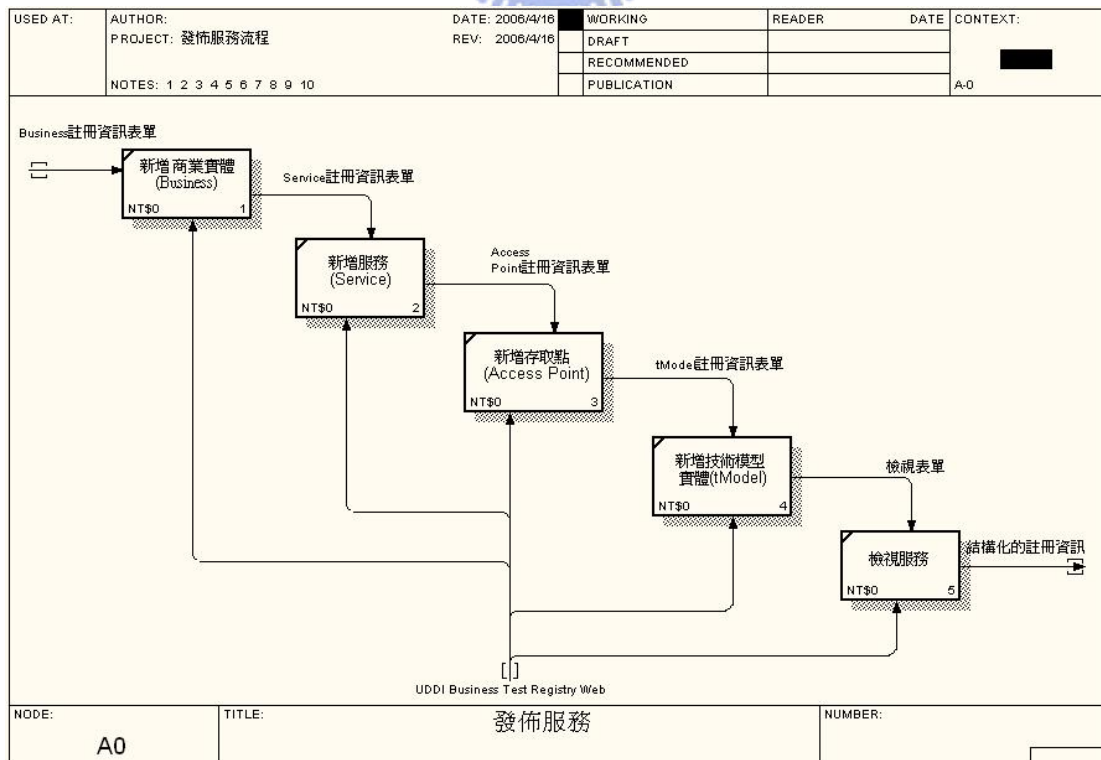
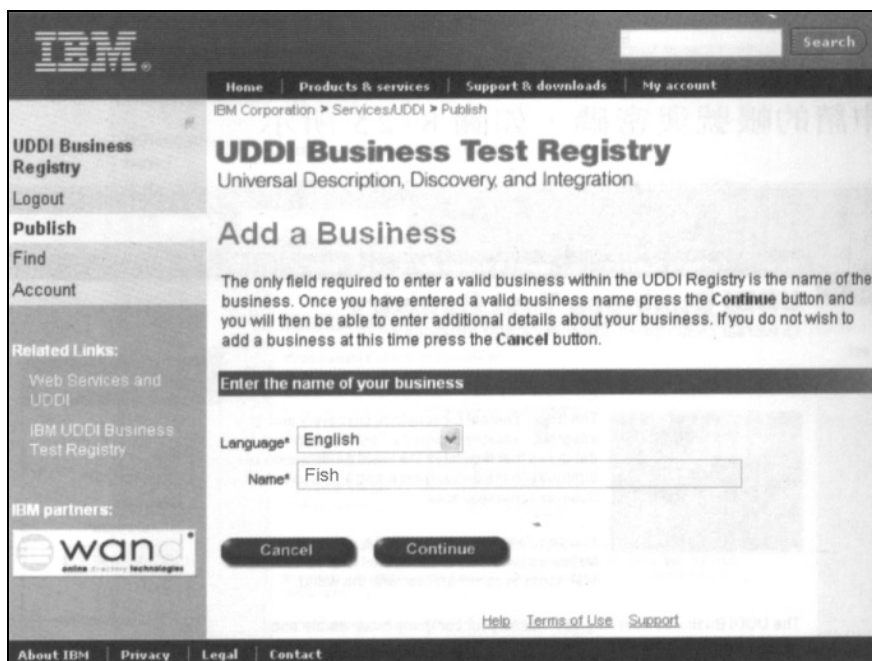


圖 5.5 發佈服務操作流程



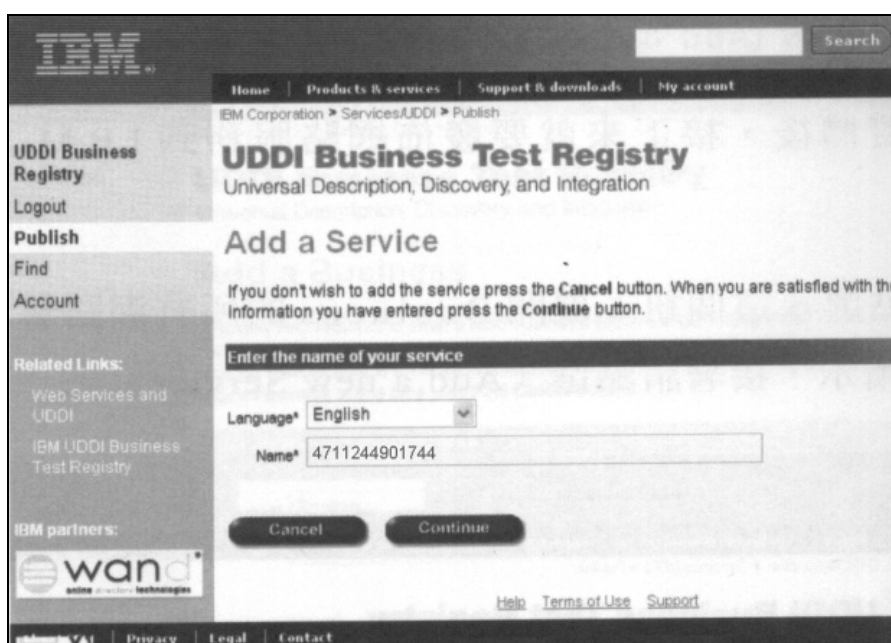
在發佈網路服務到服務仲介端之前，必須先註冊商業實體。商業實體就是服務提供者的名稱，用以告知服務需求端此服務由誰所提供，本研究以「Fish」為商業實體名稱。關於商業實體的新增如圖 5.6 所示，此步驟除了新增商業實體名稱以外，還可建立商業實體聯絡人以及相關描述等資訊。



The screenshot shows the IBM UDDI Business Test Registry interface. The main heading is "UDDI Business Test Registry" with the subtitle "Universal Description, Discovery, and Integration". The page title is "Add a Business". Below the heading, there is a paragraph of instructions: "The only field required to enter a valid business within the UDDI Registry is the name of the business. Once you have entered a valid business name press the **Continue** button and you will then be able to enter additional details about your business. If you do not wish to add a business at this time press the **Cancel** button." Below this, there is a section titled "Enter the name of your business" with a "Language\*" dropdown menu set to "English" and a "Name\*" text input field containing the word "Fish". At the bottom of this section are "Cancel" and "Continue" buttons. The page also features a left sidebar with navigation options like "Logout", "Publish", "Find", and "Account", and a footer with links for "Help", "Terms of Use", and "Support".

圖 5.6 新增商業實體[33]

建立商業實體後接續新增服務，本研究中的服務是以 3.2.3 節所定義的養殖戶唯一識別碼為註冊名稱，本節說明以其中一項服務名稱為「4711244901744」，其操作畫面如圖 5.7 所示。



The screenshot shows the IBM UDDI Business Test Registry interface for adding a service. The main heading is "UDDI Business Test Registry" with the subtitle "Universal Description, Discovery, and Integration". The page title is "Add a Service". Below the heading, there is a paragraph of instructions: "If you don't wish to add the service press the **Cancel** button. When you are satisfied with the information you have entered press the **Continue** button." Below this, there is a section titled "Enter the name of your service" with a "Language\*" dropdown menu set to "English" and a "Name\*" text input field containing the number "4711244901744". At the bottom of this section are "Cancel" and "Continue" buttons. The page also features a left sidebar with navigation options like "Logout", "Publish", "Find", and "Account", and a footer with links for "Help", "Terms of Use", and "Support".

圖 5.7 新增服務[33]

建立服務名稱後必須新增對應的服務存取點，用來記錄服務存取此服務的位址、存取協定以及相關描述等資訊。本節說明的一項服務名稱「4711244901744」其對應的存取點為「140.113.59.178:8080/axis/services/Tanhou」；且使用的存取協定為「HTTP」。其操作畫面如圖 5.8 所示。

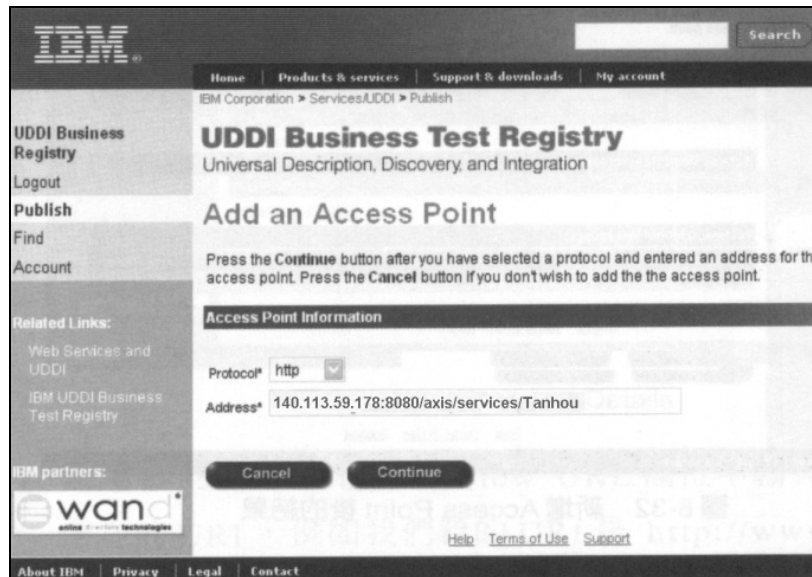


圖 5.8 新增存取點[33]

接著必須發佈網路服務描述語言文件，此資訊包含在技術模型實體之下。如圖 5.9 所示，在「Overview URL」欄位輸入網路服務描述語言文件所在位址，因此本節說明的服務「4711244901744」所對應的「Overview URL」欄位為「http://140.113.59.178:8080/axis/services/Tanhou/Tanhou.wsdl」。完成前述步驟後，透過重新檢視服務來確認該項服務是否成功發佈，若輸入的資訊均已在檢視服務中呈現，表示該項服務已經成功發佈。如圖 5.10 所示，商業實體名為「Fish」，該商業實體提供一項服務名為「471124491744」。

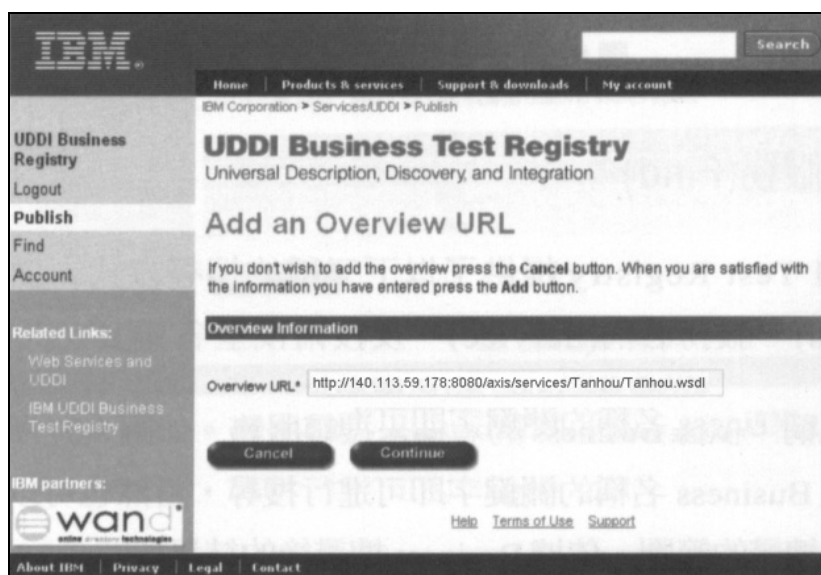


圖 5.9 新增技術模型實體[33]

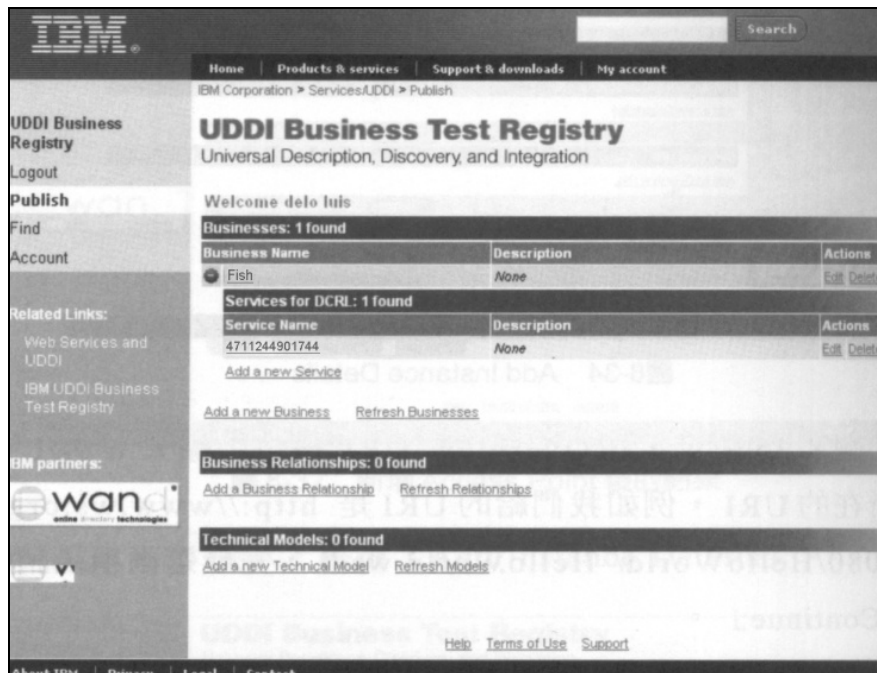


圖 5.10 服務成功發佈[33]

IBM UDDI Registry 提供三種搜尋方式：依照商業實體搜尋、依照服務搜尋以及依照技術模型搜尋。依照商業實體搜尋是指輸入商業實體的名稱來搜尋所提供的服務資訊；而依照服務搜尋是指輸入服務名稱來搜尋相關的服務資訊；依照技術模型搜尋則是指輸入技術模型名稱以搜尋相關的服務資訊。由於本研究使用的發佈服務名稱是養殖戶唯一識別碼，因此使用依照服務搜尋相關的服務資訊。如圖 5.11 所示。

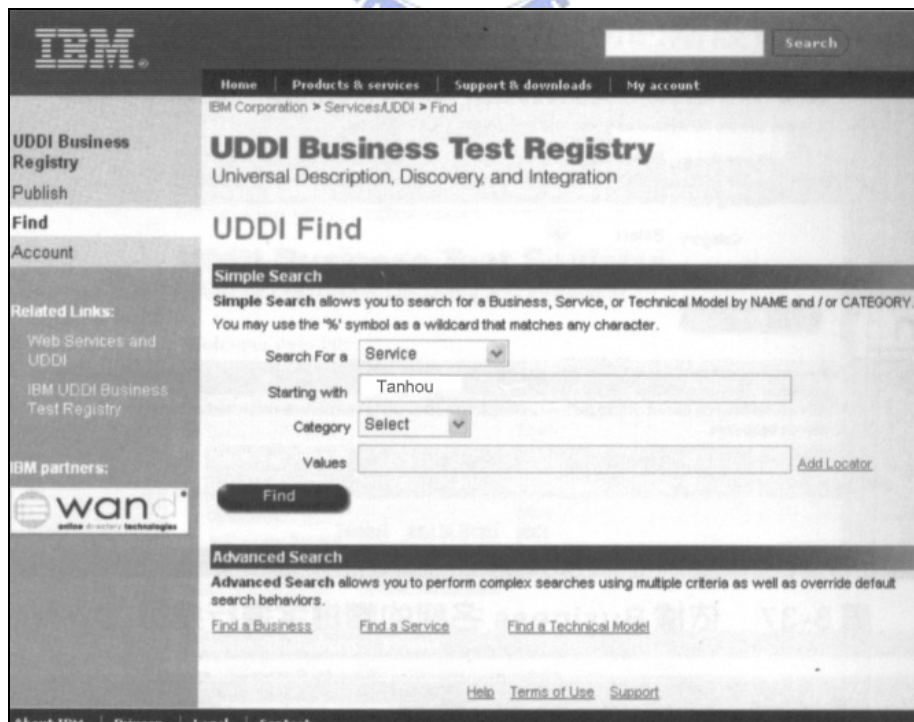


圖 5.11 依照服務搜尋[33]



## 5.3 以 Non-Web API 設計服務需求端與仲介端的溝通

一般使用者操作 5.2 節 Web-based 的介面使用 UDDI 註冊中心，可以達到註冊、發佈與搜尋等基礎功能，而且不需要自行開發程式。但是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。本研究在註冊以及發佈的功能上只需要透過 Web-based 介面操作即可，但是搜尋功能卻需要與 5.1 節的服務需求端應用程式整合，將 UDDI 搜尋的結果讓服務需求端應用程式可以自動呼叫服務伺服器端。5.3.1 節說明本研究的仲介端可展加註語言文件規格；瞭解規格後於 5.3.2 節使用 UDDI4J 這套 Non-Web API 解析仲介端可展加註語言文件，並與服務需求端應用程式整合。

### 5.3.1 仲介端可展加註語言(XML)文件規格

UDDI 規範的服務註冊相關資訊均使用可展加註語言格式來界定，使得服務仲介端能夠遵循標準程序，處理網路服務中所負責的任務。在可展加註語言文件中把發佈資訊分為三個類型：白皮書、黃皮書以及綠皮書。白皮書是關於商業實體基本資訊，例如商業實體名稱、聯絡資訊等等；而黃皮書則可視為依類別區分的電話簿索引，目的是讓服務需求端的使用者按照分類尋找已經登錄發佈的商業實體；綠皮書說明如何與服務伺服器端進行溝通的技術資訊細節，例如網路服務技術規格的參考資訊、網路服務取得的參考格式等。

在上述三個類型中所登錄發佈的資訊是使用可展加註語言綱要定義，共包含五種資料結構：businessEntity、businessService、bindingTemplate、tModel、publisherAssertion。本研究的仲介端可展加註語言文件與這五種資料結構的對應如圖 5.12 所示，以下將逐一說明。

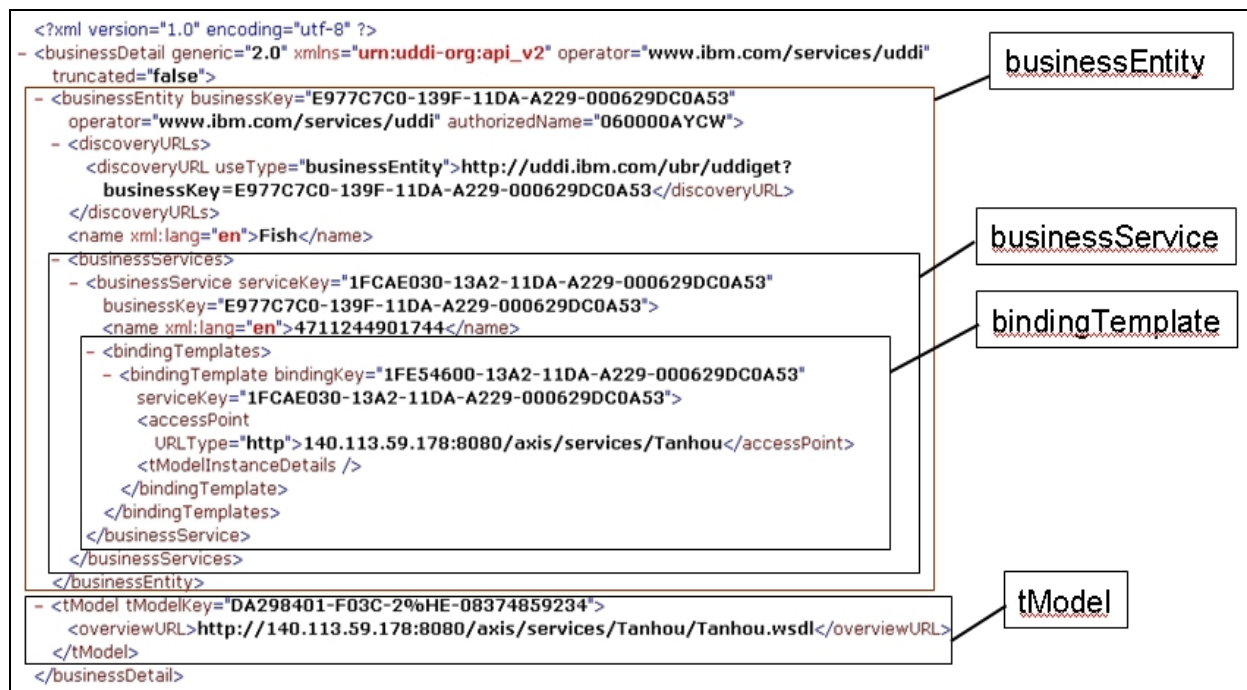


圖 5.12 可展加註語言文件的資料結構對應



businessEntity 的目的是讓可能的服務需求者能夠取得該公司正確的相關資訊，並對其所提供的網路服務定址。其資料結構包含白皮書與黃皮書的功能。本研究在 businessEntity 根元素下紀錄了 businessKey、discoveryURLs、name、businessServices 等元素，相關涵義如表 5.1 所示。

表 5.1 businessEntity 使用元素說明

元素名稱	元素值	說明
businessKey	E977C7C0-139F-11DA-A229-000629DC0A53	服務仲介端系統自動產生的商業實體識別碼
discoveryURLs	http://uddi.ibm.com/ubr/uddiget?businessKey=E977C7C0-139F-11DA-A229-000629DC0A53	說明如何執行 UDDI 查詢功能的檔案或連結
name	Fish	本研究針對水產品所進行的商業實體命名
businessServices	(本研究內含一項 businessService)	服務的集合名稱，表示 Fish 這個商業實體所提供各種服務的代表

businessService 是 businessServices 內的子集合，包含一系列有關商業流程的網路服務，並且與服務的描述組合在一起。businessService 與 bindingTemplate 組合成為綠皮書的內容，關於 businessService 所包含的元素相關涵義在表 5.2 說明。

表 5.2 businessService 使用元素說明

元素名稱	元素值	說明
serviceKey	1FCAE030-13A2-11DA-A229-000629DC0A53	服務仲介端系統自動產生的服務識別碼
businessKey	E977C7C0-139F-11DA-A229-000629DC0A53	對應 businessEntity 的商業實體識別碼
name	4711244901744	登錄提供服務的名稱，本研究以養殖廠商的識別碼作為服務名稱
bindingTemplates	(包含一項 bindingTemplate)	記錄本項服務對應的相關技術鏈結資訊，表示服務與技術鏈節的包含關係

bindingTemplate 的資料結構用來對應其上層的 businessService 節點，例如技術鏈結與通訊所需要的技術細節、網路服務的存取位址等。本研究關於 bindingTemplate 所使用的元素相關涵義在表 5.3 說明。

表 5.3 bindingTemplate 使用元素說明

元素名稱	元素值	說明
bindingKey	1FE54600-13A2-11DA-A229-000629DC0A53	服務仲介端系統自動產生的鏈結識別碼
serviceKey	1FCAE030-13A2-11DA-A229-000629DC0A53	對應 serviceKey 的服務識別碼
accessPoint	140.113.59.178:8080/axis/services/Tanhou	服務鏈結所描述的服務存取位址，本研究是使用 URL 的方式以及 HTTP 協定
tModelInstanceDetails	(空元素)	用來描述技術規範，包含服務鏈結存取點的描述

tModel 的資料結構用來描述技術細節。由於存取網路服務的存取位址已經在 bindingTemplate 中定義，但仍然缺少部份更具體的資訊，例如回應的格式、參數型態與順序、安全機制等。關於 tModel 使用的元素涵義如表 5.4 所示。

表 5.4 tModel 使用元素說明

元素名稱	元素值	說明
tModelKey	DA298401-F03C-2%HE-08374859234	服務仲介端系統自動產生的 tModel 識別碼
overviewURL	http://140.113.59.178:8080/axis/services/Tanhou/Tanhou.wsdl	透過超鏈結可以得到此 tModel 具體技術規範

關於 publisherAssertion 的資料結構是用來說明不同商業實體之間的相關性。但是本研究僅使用單一商業實體「Fish」，在將各個服務建置於「Fish」之下，因此並未使用 publisherAssertion 元素。

### 5.3.2 使用 UDDI4J 解析仲介端可展加註語言文件

針對網路服務仲介端三項任務：登錄註冊、網路服務的發佈以及服務查詢，本研究在開發服務需求端應用程式方面的需求概念如圖 5.13 所示，主要是在服務需求端使用服務名稱為關鍵字，向服務仲介端查詢相關的服務細節。因此在開發服務需求端應用程式上，也只需要實作查詢這項功能；而登錄註冊與發佈兩項功能則透過 5.3.1 節的 Web-based 介面執行。

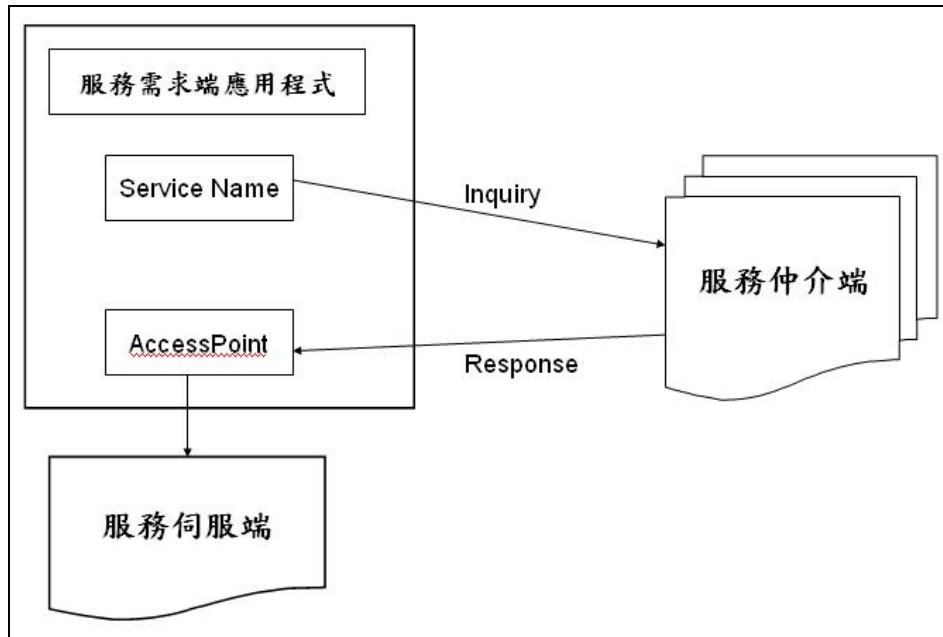


圖 5.13 服務需求端應用程式與服務仲介端互動關係

本研究使用 UDDI4J 這項 API 工具來實作圖 5.14 中服務需求端應用程式與服務仲介端溝通的過程。UDDI4J 是開放原始碼的函式庫，提供如同 5.3.1 節 Web-based 介面的註冊、發佈與查詢功能。表 5.5 是 UDDI4J 主要的 API 結構與功能對照。

表 5.5 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.request	發送請求訊息到服務仲介端主機
org.uddi4j.response	從服務仲介端主機回傳執行結果
org.uddi4j.transport	傳送附屬資訊

圖 5.14 使用 UDDI4J 取得服務需求端所需資訊。主要分為四個部份：設定 InquiryURL 進入點、傳送 ServiceName、建立 Service 鏈結節點以及取得 AccessPoint 服務位址。InquiryURL 可視為服務仲介端解析服務的進入點，必須一開始就完成位址設定，由於本研究在 IBM UDDI Registry 註冊發佈，因此 InquiryURL 必須設定 IBM 提供服務仲介的進入點位址；而傳送 ServiceName 則是指服務需求端應用程式將服務名稱送出，用來取得服務仲介端可展加註語言文件中關於 ServiceInfo 相關節點的資訊；建立 Service 鏈結節點指的是利用前一步驟透過 ServiceName 所取得的服務識別碼(ServiceKey)，建立鏈結的節點；最後利用此鏈結節點，取得鏈結資訊 AccessPoint 位址，再將此結果回傳給服務需求端應用程式進行後續處理。

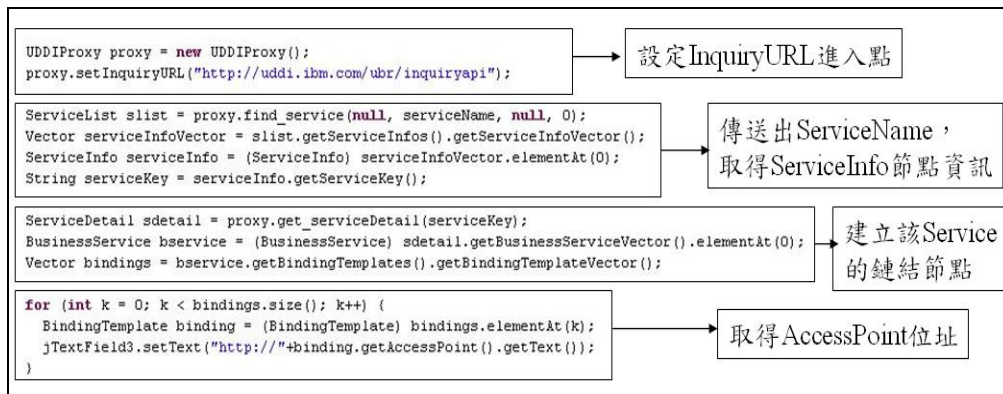


圖 5.14 使用 UDDI4J 套件取得服務需求端所需資訊的程式說明





## 5.4 射頻識別讀寫功能的設計

本節說明射頻識別系統的讀寫功能設計。在服務需求端的軟體操作時，必須配合射頻識別讀寫器硬體的讀寫，作為識別碼的資料來源，有了識別碼的資料才能繼續進行 5.1、5.2 與 5.3 節的後續處理動作。本節共分三小節：5.4.1 節說明射頻識別的硬體架構、5.4.2 節說明射頻識別的軟體設計架構、5.4.3 節則是射頻識別系統的操作。

### 5.4.1 射頻識別硬體架構

本研究所使用美商Intermec公司的IF5 射頻識別讀寫器是一套Linux OS 2.6 為核心的嵌入式系統主機[38]，此射頻識別裝置架構如圖 5.15。主機內建一套BRI指令集讓使用者能夠簡化操作流程，透過BRI指令集可以完成天線啟動與關閉、射頻識別讀取與寫入、編碼轉換等功能。而此BRI指令集需要運用Socket技術與射頻識別讀寫器主機互相傳送溝通，有關BRI與Socket技術的使用將在 5.4.2 節中說明。IF5 主機運作分為訊號輸入與訊號輸出兩部分。在訊號輸入端使用銅軸纜線與感應天線互相連接，感應天線會持續發出射頻電波在大約 2 公尺的有效範圍內，當射頻識別標籤進入到此電波作用區中，射頻電波會提供動力給予識別標籤，讓識別標籤啟動並且傳送內含的資料，天線會將接收到的資料透過銅軸電纜傳輸到IF5 主機。而對IF5 主機訊號輸出端而言，使用乙太網路輸出，使得產銷履歷入口等網路服務應用程式可以在網際網路上取得射頻識別標籤內含的資訊。圖 5.16 以 IDEF0 流程圖說明射頻識別裝置的訊號讀取流程。

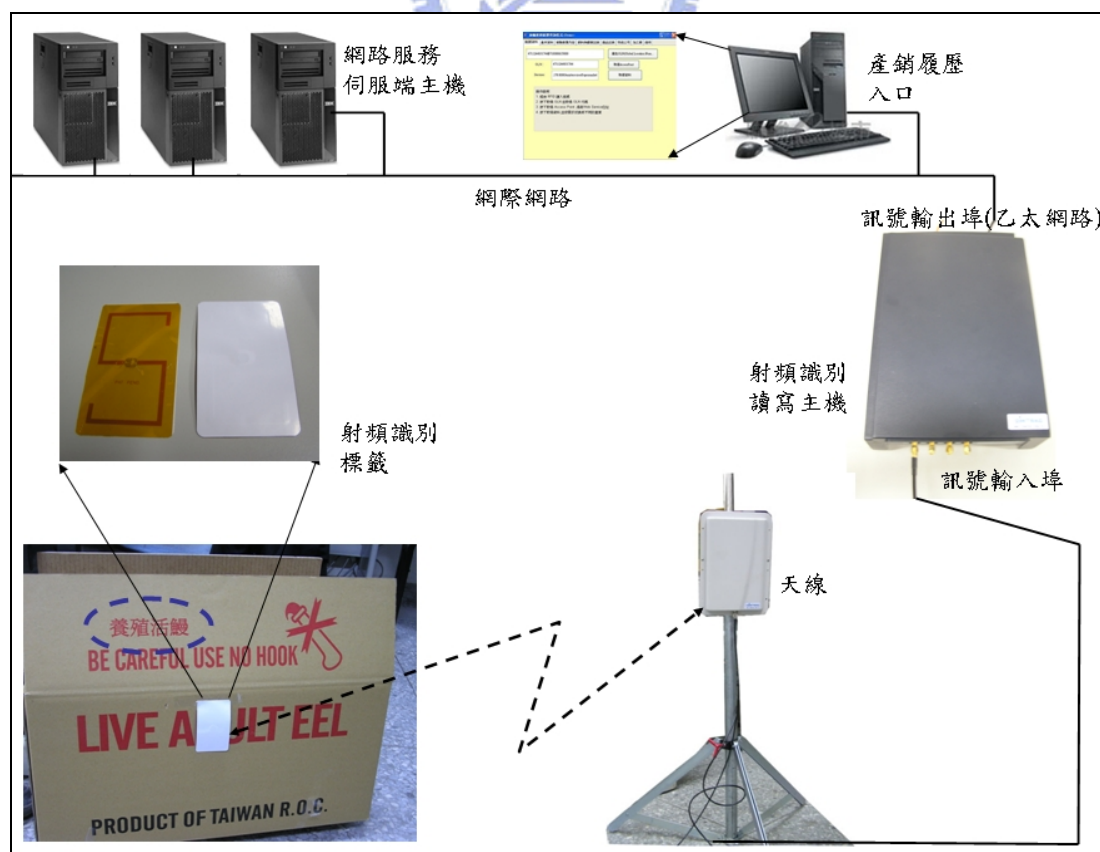


圖 5.15 Intermec IF5 射頻識別裝置架構圖

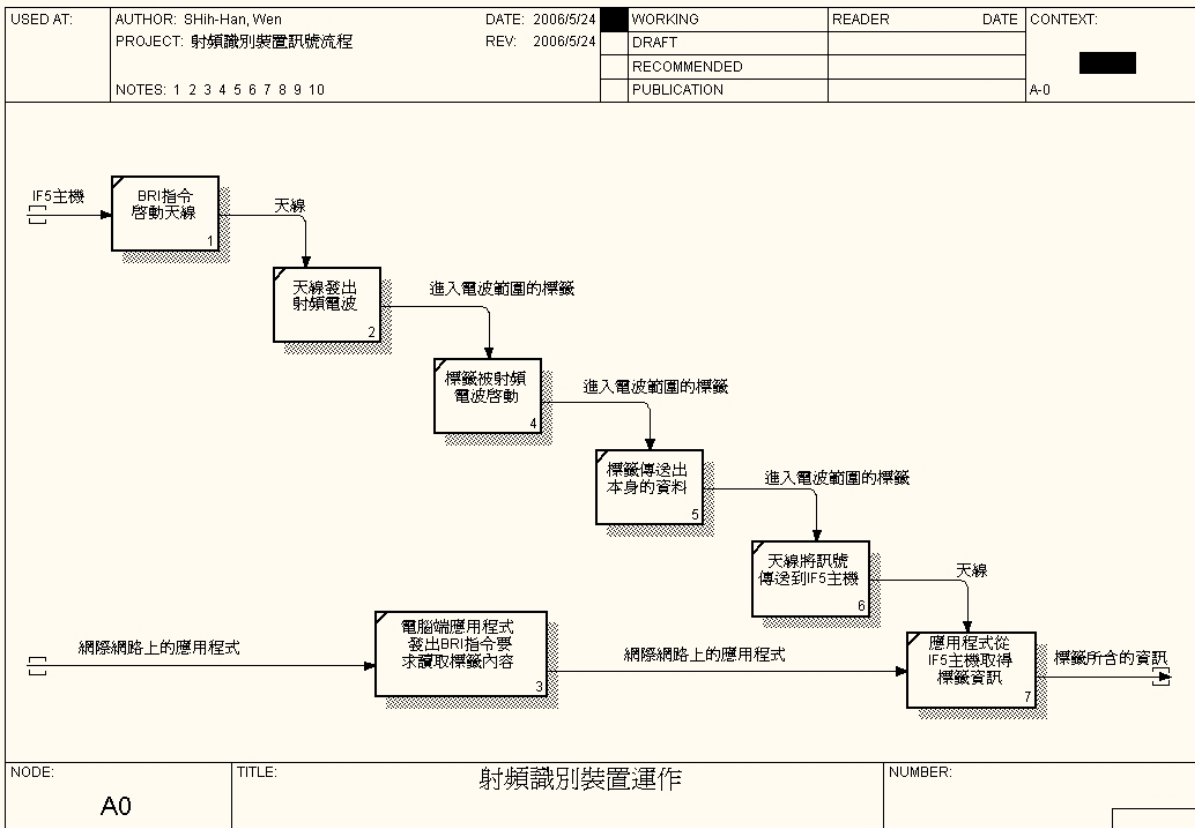


圖 5.16 射頻識別裝置的訊號讀取流程

對於射頻識別標籤而言可以分為天線與晶片兩部份。天線的功能是感應射頻電波，以磁生電的方式啟動晶片部分，使得標籤具有讀寫的功能；而晶片的部份是用來記錄資料的記憶體區塊，分為唯讀區塊的 UID 識別碼以及可讀寫的資料區塊。標籤與記憶內容格式如圖 5.17 所示。

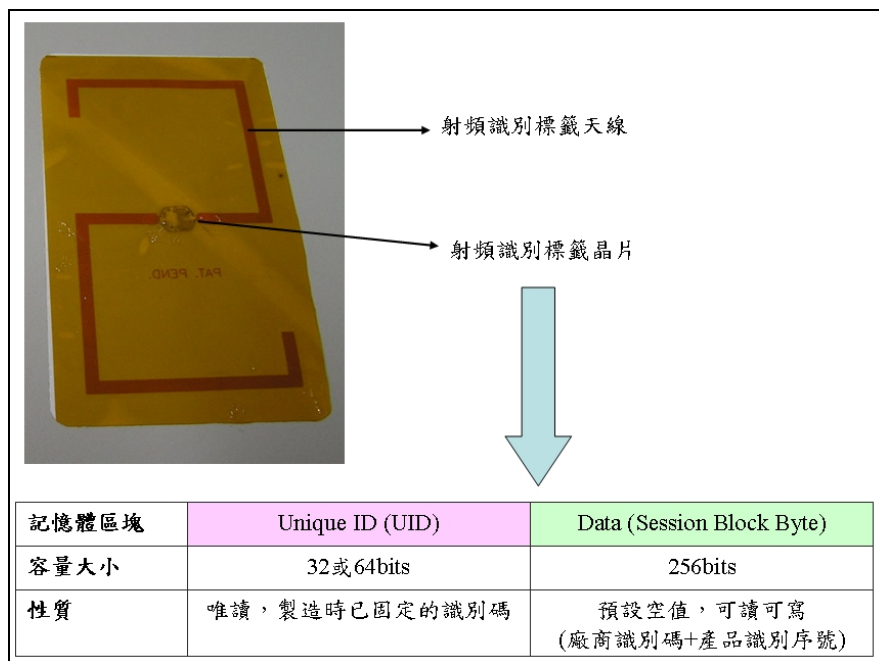


圖 5.17 射頻識別標籤規格

## 5.4.2 射頻識別軟體架構

透過網際網路傳輸資料的方式中，Socket 的觀念讓網路連線可視為另一種資料流，在建立 Socket 連線後，將寫入 Socket 的資料傳送至遠端的射頻識別讀寫器，而遠端的射頻識別讀寫器也將執行結果寫入 Socket 傳送回來。簡易的 Socket 技術級廣泛運用到網路的各層面上。

在射頻識別系統中，射頻識別讀寫器可視為 Socket 遠端，而服務需求端軟體可視為 Socket 本地端，其架構如圖 5.16 所示。對遠端而言，ServerSocket 類別為遠端所建立的 Socket，遠端以 ServerSocket() 建構方法在預設埠號 2189 的通訊埠建立一個 ServerSocket 物件，並以 ServerSocket 的 accept() 方法傾聽來自本地端的連線。而對本地端而言，在連線狀態的過程中使用 Socket() 建構方法產生新的 Socket 物件，然後告知遠端的網路位址以及通訊埠號 2189，便可建立與本地端資料傳輸的連線。在本地端的 Socket 物件達成與 ServerSocket 的連線後才會建立連結兩端的 Socket 物件，並透過兩端 Socket 物件提供的輸出入資料流，使得兩端的主機都可以同時接收與傳送命令，連結兩端點的 Socket 物件將會持續效用到伺服端或客戶端發出 close() 方法即結束。

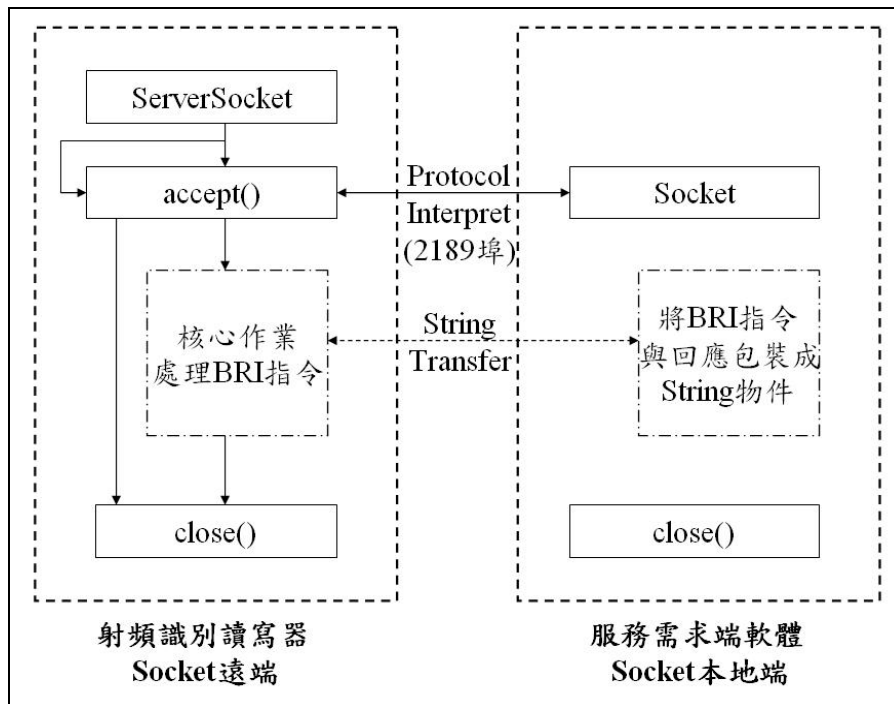


圖 5.16 射頻識別讀寫功能溝通架構

在表 5.6 中，將列出服務需求端軟體傳送 BRI 指令集操作射頻識別讀寫器的原理所對應之 Java 程式碼。主要分成三部份：設定連線、設定 I/O、BRI 指令傳出與接收。設定 Socket 連線必須以連線位址與埠號來初始化；I/O 的設定產生 DataInputStream 與 DataOutputStream 來作為 BRI 指令傳出與接收結果的物件；而 BRI 指令傳出與接收是使用 BRI 指令 readString() 發出射頻識別讀取命令，透過 Socket 傳送至射頻識別讀寫器主機並執行處理後，會將讀取值傳回給服務需求端軟體，此時用 Str1 物件接收回應結果。

表 5.6 射頻識別讀寫器的原理所對應之 Java 程式碼

原理說明	Java 程式碼
設定連線	Socket socket = new Socket("140.113.59.175", 2189);
設定 I/O	DataInputStream datain = new DataInputStream (socket.getInputStream()); DataOutputStream dataout = new DataOutputStream(socket.getOutputStream());
將 BRI 指令傳出 並且接收回應結果	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);

### 5.4.3 射頻識別系統的操作

在 5.4.1 節與 5.4.2 節已分別說明射頻識別系統的軟硬體架構，本節將說明其操作流程。射頻識別系統的操作過程可分為兩部份：參數設定與 BRI 指令的執行。關於參數設定部份，在正常的開機程序如圖 5.17 之後，Intermec IF5 主機載入 Linux2.6 作業系統，同時提供 BRI 指令集的服務。此時可以透過系統內建的操作網頁，進行使用環境相關參數的設定，如圖 5.18 所示。

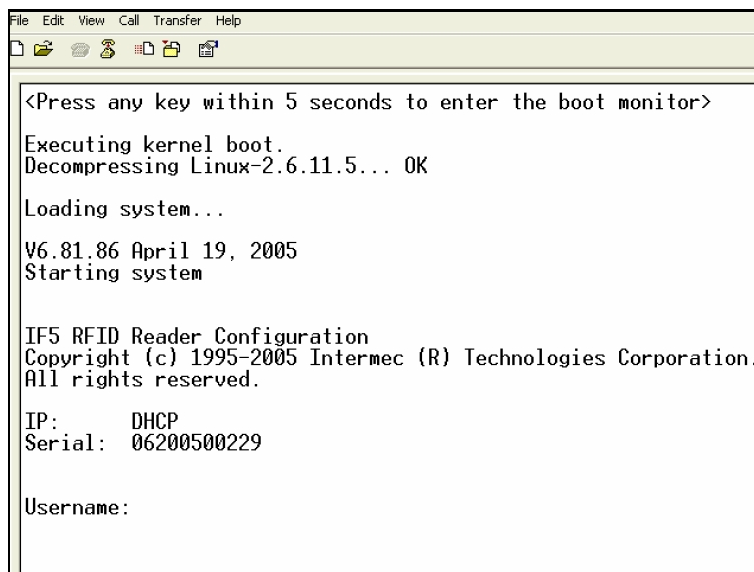


圖 5.17 IF5 開機啟動畫面

操作參數的設定部份可分為硬體版本、讀寫報告與讀寫使用狀態以及天線使用狀態三



項。硬體版本設定射頻識別標籤的種類、通訊介面的版本；而讀寫報告與讀寫使用狀態則設定了讀寫報告的使用狀態以及讀寫不成功時重新嘗試的次數限制；天線使用狀態則是設定目前使用天線的數目與編號，同時可以針對已連接的天線決定是否要使用。

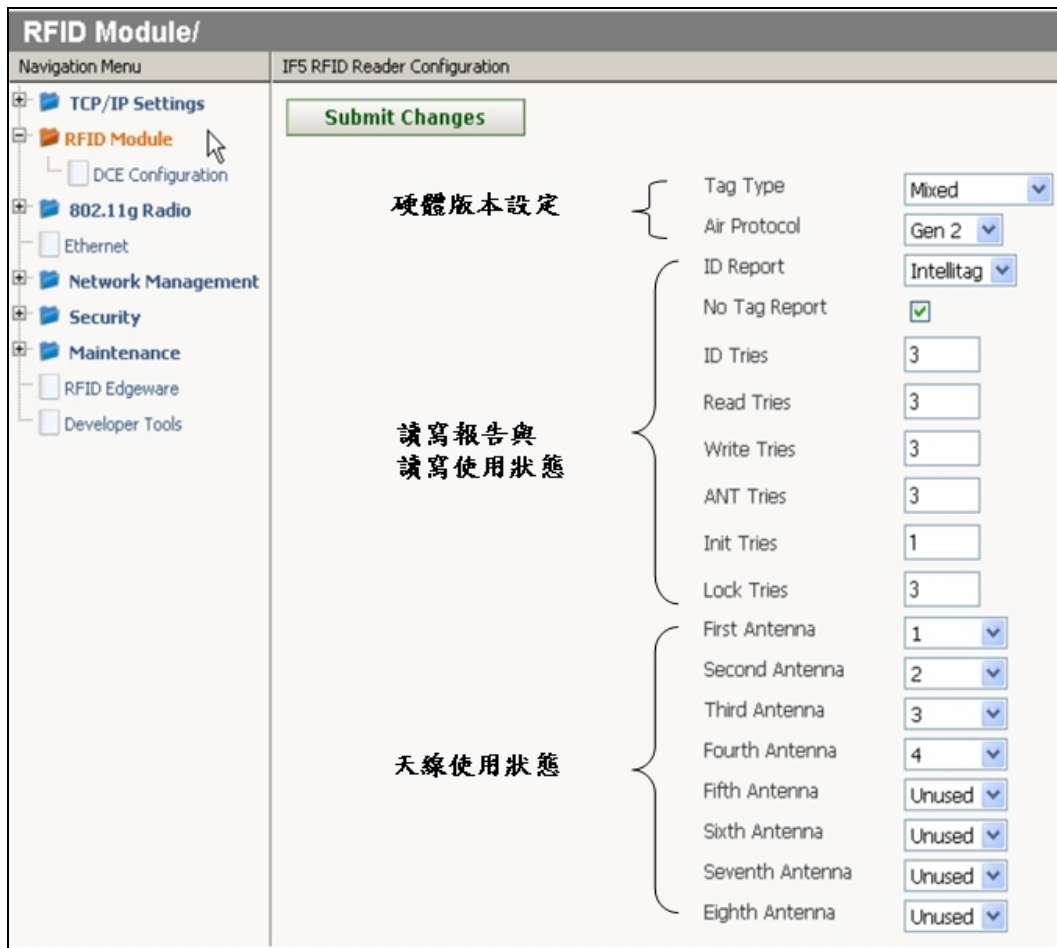


圖 5.18 操作參數的設定畫面[38]

而對於 BRI 指令的執行而言，可以使用單行指令列或是段落式腳本(Script)的方式執行。圖 5.19 為單行指令列的方式執行「讀取射頻識別標籤」此動作，在單行指令列中打入「READ」，即可從 IF5 主機取得標籤的資料區塊內容「HE004487BCC000000」。而段落式腳本的優點是類似一般的程式語言，可使用變數以及迴圈等基本運算功能，可以執行過程較為複雜的讀寫功能，如圖 5.20 所示。



圖 5.19 單行指令列執行BRI[38]

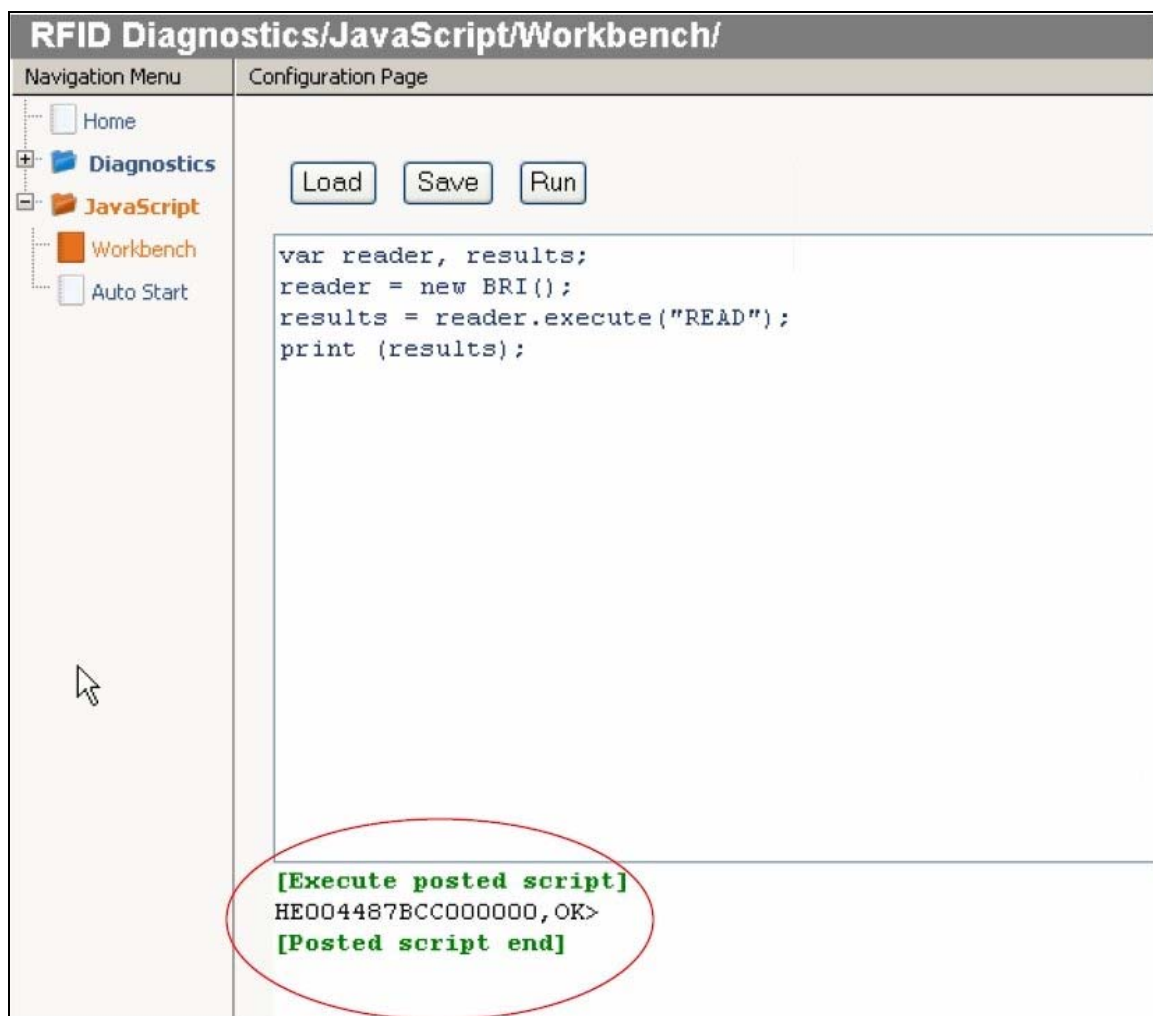


圖 5.20 段落式腳本執行BRI[38]

## 5.5 服務需求端整合入口的操作

本節說明產銷履歷入口的操作步驟。產銷履歷入口就是網路服務技術架構中的服務需求端程式與射頻識別系統的整合介面，此入口程式的操作主要透過輸入產品識別批號，再透過網路服務技術架構追溯相關的資訊。本研究使用 EAN.UCC 標準下的供應商識別碼/全球位址碼(Global Location Number, GLN)、全球商品碼(Global Trade Item Number, GTIN)以及配送包裝識別碼(Serial Shipping Container Code, SSCC)作為養殖水產供應鏈的唯一識別編碼，藉此克服不同生產業者間的識別與通訊障礙。

圖 5.21 是產銷履歷入口讀取射頻識別或是一般條碼作為輸入資料，然後運用網路服務技術的各項服務提供者與仲介者，完成一項產品的履歷資料的追溯與整合。關於射頻識別系統的讀取流程已在 5.4.1 節中說明，當產銷履歷入口透過射頻識別系統取得標籤的資料後，會將標籤內容轉換為超級市場的 GLN 碼以及 GTIN 碼，再依序進行圖中的八項流程。第一步驟：會將此資料送到 IBM UDDI Business Registry 仲介主機以取得超級市場 GLN 碼對應的網路服務描述語言文件所在位址；第二步驟：向超級市場網路服務伺服器取得該產品在超級市場的相關處理資料，同時在此相關資料中取得上游加工廠的 GLN 碼與 GTIN 碼；第三步驟：向 IBM UDDI Business Registry 仲介主機取得與加工廠 GLN 碼對應的網路服務描述語言文件所在位址；第四步驟：向加工廠網路服務伺服器取得該產品在加工廠的相關加工資料，同時也獲取上游的物流公司 GLN 碼與 GTIN 或 SSCC 碼；第五步驟：IBM UDDI Business Registry 仲介主機以取得物流公司 GLN 碼或 SSCC 碼對應的網路服務描述語言文件所在位址；第六步驟：物流公司網路服務伺服器取得該產品在物流公司的相關處理資料，同時也獲取上游的養殖場 GLN 碼與 GTIN 碼；第七步驟：向 IBM UDDI Business Registry 仲介主機取得與養殖場 GLN 碼對應的網路服務描述語言文件所在位址；第八步驟：向養殖場網路服務伺服器取得該產品在養殖場的相關生產資料。

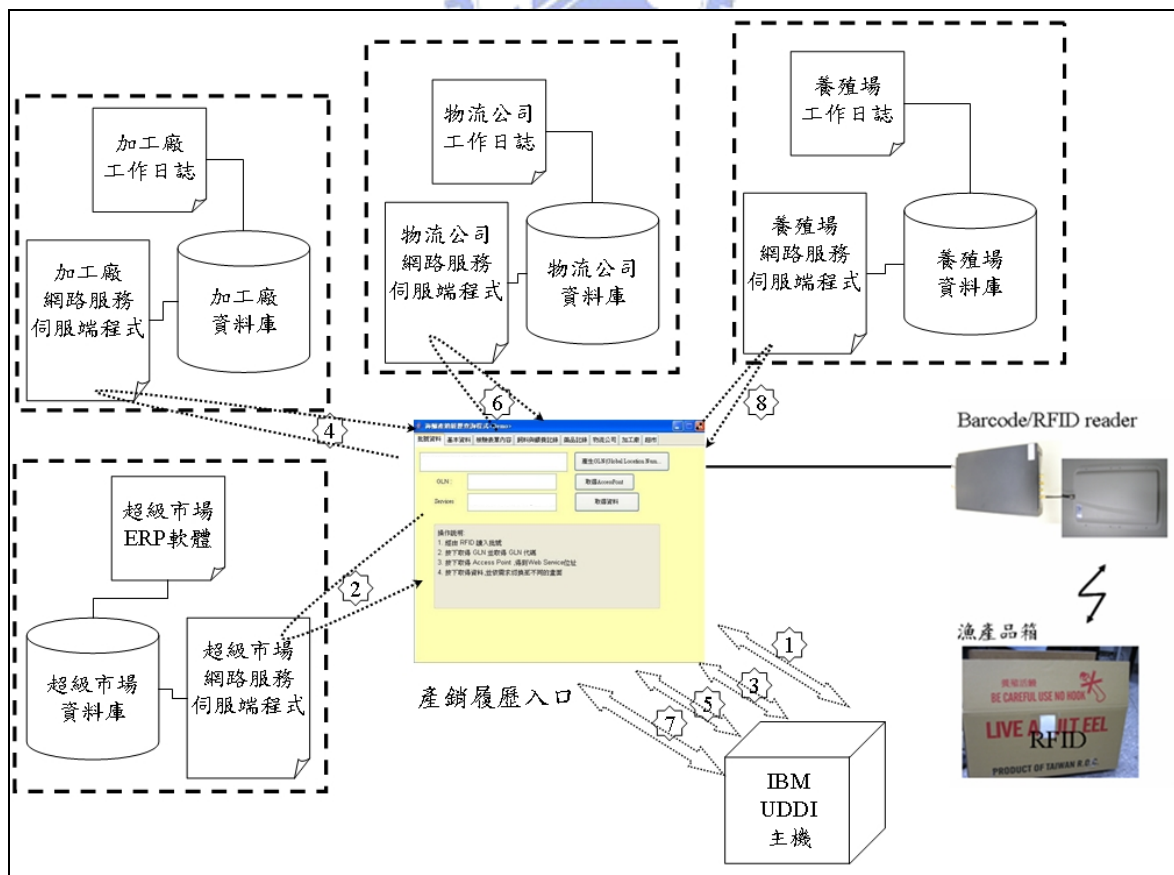


圖 5.21 產銷履歷入口與網路服務整合執行流程

圖 5.22(a)為產銷履歷入口的初始主畫面，第一個文字方塊可以透過射頻識別感應、條碼機或是人工鍵盤輸入的方式輸入產品唯一識別碼，如圖 5.22(b)所示。透過第一個按鈕「產生 GLN(Global Location Number)」可以對產品相關識別碼進行解析，將解析的結果顯示在第二個文字方塊如圖 5.22(c)所示；而第二個按鈕「取得 AccessPoint」則會利用文字方塊內的代碼，向 IBM UDDI Business Registry 取得服務所在位址，並將該服務位址顯示在第三個文字方塊，如圖 5.22(d)所示；第三個按鈕「取得資料」則會向服務位址請求資料查詢的服務。經過服務回傳的資訊會依照其在供應鏈中的位階類別，顯示在相對應的頁籤中。有關於取得資料在頁籤中的顯示在圖 5.23 中說明。

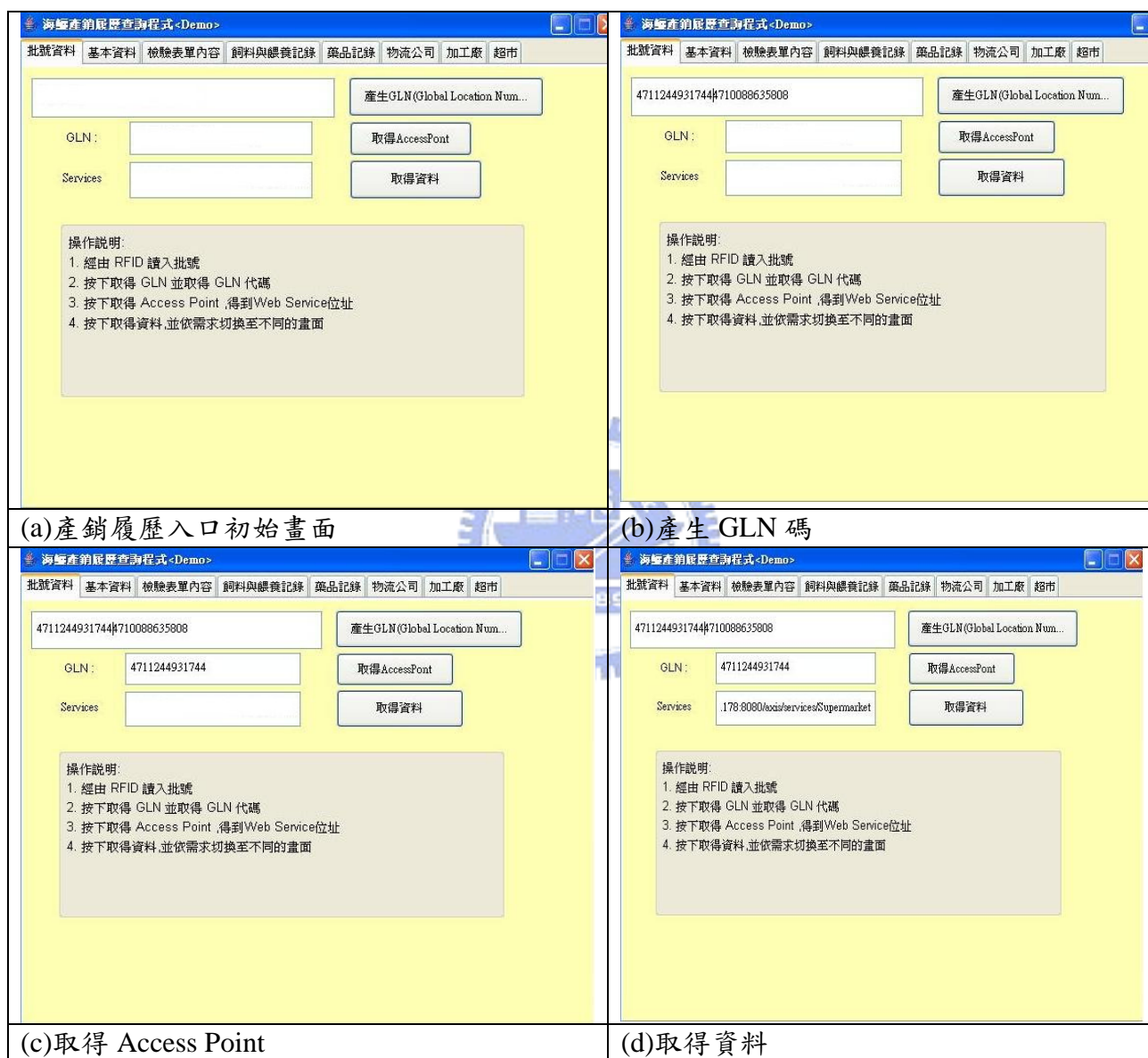


圖 5.22 產銷履歷入口操作說明

圖 5.23(a)是向養殖場取得養殖場名稱、承銷商、檢驗表單單號、捕撈日期等基本資料服務。圖 5.23 (b)是向養殖場取得檢驗中心、檢驗項目、檢驗值等檢驗表單內容服務。圖 5.23 (c)是向超級市場取得進貨批號、進貨業者、進貨時間、進貨數量、銷售批號等商品資訊服務。圖 5.23 (d)是向物流公司取得進貨批號、出發時間、運輸工具、抵達時間等物流資訊服務。圖 5.23 (e)是向加工廠取得進貨批號、進貨時間、加工動作、包裝方式、出貨批號等加工資訊服務。圖 5.23(f)是向養殖場取得投藥日期、藥品品項、投放量、藥品主要成分與添加物等藥品



投放紀錄服務。圖 5.23(g)是向養殖場取得餵食日期、飼料品項、投放量、飼料主要成分與添加物等飼料與餵養紀錄服務。



圖 5.23 產銷履歷服務畫面

## 第六章 結論與未來方向

本章對於本論文所提出的網路服務與射頻識別技術整合架構與應用作一總結。6.1 節簡述本研究的產銷履歷追蹤與追溯架構。6.2 節則探討未來可能加以改進與發展的方向。

### 6.1 結論

本研究主要的目的在於供應鏈中建立一個資訊整合交換的系統架構，這個架構是以產業流程以及產業知識為基礎，進而搭配各項資訊技術而得。因此不論未來的資訊技術如何演進，都可以利用這個架構，建立新世代的產銷履歷資訊整合系統。在本節中將探討技術、產業與科技對於系統的影響。

在本研究中可分為兩大架構：工作日誌追蹤系統與產銷履歷追溯系統。工作日誌追蹤系統利用 IDEF0 與 IDEF1X 的技術進行流程分析與系統規劃，有效縮減開發工作的時程，並且確保資料庫架構設計的正確性。而對於產銷履歷追溯系統而言，透過網路服務技術的特性，改善了目前單純採用條碼或射頻技術所搭配的集中式軟體架構缺失，其優點在於軟體簡易開發與佈署、分散式的資料庫架構以及開放的資訊共享；另一方面，射頻識別技術在本研究中可以有效改善條碼記錄空間不足的缺失以及讀寫速率的速度。

產銷履歷資訊系統從設計規劃到執行，其可行性是著眼於四項考量因素：投資報酬率、技術支援、產業供應鏈配合度以及作業流程分析。對投資報酬率而言，傳統採用集中式架構的產銷履歷系統的軟硬體由於必須負荷極大量的網路連線與資訊儲存功能，因此軟硬體成本將高達數百萬。但是藉由本研究所提出的分散式架構整合系統，其所需的成本僅需要普通的桌上型電腦以及網路連線費用而已，預計一年之內可以完全回收。這樣的投資報酬率不論在何種產業均是可以接受的範圍，因此推行上可以較為順利。對於技術支援而言，由於從西元 2000 年開始可展加註語言技術的蓬勃發展，目前已有相當大量的相關套件問世，而各資訊大廠也將此規格整合到各自的軟體系統中。因此根植於可展加註語言的網路服務技術，目前在開發與標準規劃上都已非難事，因此目前可說是網路服務技術實作導入的最佳時機。而對於產業供應鏈配合度而言，在 1.2 節已說明目前供應鏈業者對於現行集中式架構的疑慮，而在目前現行的集中式架構下，只把資訊整合的對象集中在養殖業者而非供應鏈的整個環節，無法達成供應鏈資訊的整合交換。而對於作業流程分析而言，本系統規劃之初，已經對於台灣的養殖魚產品供應鏈有完善的分析，因此當技術與相關理論成熟，能夠快速的找到資訊科技在傳統產業中所能夠展現的效益。

本研究成果具有可移植性，在其他產業中均能夠以同樣的架構進行供應鏈資訊的整合應用，可謂知識經濟的最佳運用。在本研究過程中發現，在農業界應用資訊技術必須特別考量的幾項因素：跨平台能力、標準整合、操作簡單、維修容易。對跨平台能力而言，由於各農業單位電子化程度不一，因此造成各式作業平台並存的狀況。對於政府整合農業供應鏈的立場，跨平台是首要考量，這也是本研究選用 Java 以及多套開放式原始碼進行系統實作的主因。對於標準整合而言，整合的標準是各個產業供應鏈推行資料交換的共同問題。從歷史資料觀之，技術的普及速度與將來標準的整合困難度成反比。而農業界的資訊網路相關技術普及速度相當緩慢，造成各單位皆有專用的資料標準，更遑論農產品整個供應鏈的資料格式差異。[13]因此需要有系統主動加以整合各個單位具有差異的資料格式，才符合目前情況所需。對於操作簡單而言，由於農業界的資訊人才與其他產業尚有一段差距，因此程式開發時力求操

作簡單，唯有簡單的操作介面才能夠讓此系統的推廣較為容易。而對於維修容易而言，由於農產品供應鏈的組成成員並不一定會工業界有群聚的現象，因此在系統建構時必須考慮維修的便利性，以本研究網路服務技術為核心架構下，不論是服務伺服器端或是服務需求端的軟體，均能夠很容易地透過網際網路更新發佈。





## 6.2 未來發展方向與建議

以本研究為基礎，未來的發展方向可分為兩部份，第一個部份是網路服務技術的安全性議題，第二部份是服務導向架構的業務流程管理開發技術以期能達成網路服務的協同合作。以下分別針對這兩個方向加以說明。

在可展加註語言興盛之後，最為人詬病的當數資訊安全議題。過去程式可以透過位元碼(binary code)的方式包裝驅動遠端的應用程式，但採用可展加註語言的架構之下，舉凡發佈、註冊、呼叫與執行結果的傳遞都是可任意讀取的文字訊息，而網路服務架構的商業應用也因此受到侷限。由於資訊安全對於電子商務發展的重要性，目前 W3C 與 OASIS 組織已針對資訊安全及加解密制定相關規範，如圖 6.1 所示。由於網路服務是以既有的通訊協定與可展加註語言為基礎，因此安全議題實際上取決於可展加註語言基礎規範在資訊安全方面的制定。

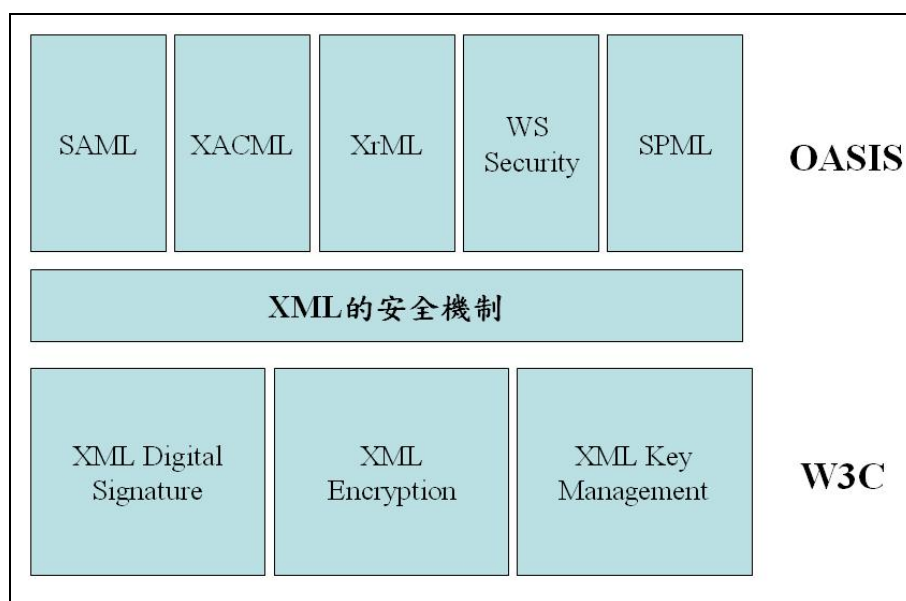


圖 6.1 可展加註語言環境下的安全機制[17]

而對於服務導向架構的業務流程開發架構而言，由於目前網路服務通常僅針對個別的服務定義其內容架構，包括註冊、發佈、鏈結等。但是在商業化的應用目標之下，單一的網路服務功能非常有限，因此常需要將許多個網路服務予以整合串接，這就是網路服務的協同運作。欲達成整合協同的功能，網路服務架構必須加入管理作業流程的機制，把作業流程以可展加註語言的形式描述在各個服務之間如何互動的規範，使得開發管理大量的網路服務變得更為簡易。由於目前網路服務的商業化有廣泛的應用，為了達成這些應用，若不採行業務流程管理的機制，網路服務的實用性將變得支離破碎。未來研究可以就這兩項議題，達成更深入的研究與整合，使得網路服務在商業運用上有更好的效果。



## 參考文獻

- [1] 工研院系統與航太技術發展中心，RFID 木棧板應用測試報告- 含水率分析，經濟部商業司，2004。
- [2] 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。
- [3] 冉繁華，陳詩璋，黃謝田，「水產養殖產銷履歷資訊系統之推動與建立」，漁業推廣，20-27 頁，2005。
- [4] 冉繁華、張正明、莊慶達、陳詩璋，水產養殖廠導入 HACCP 相關標準規範擬定研究，中華民國養殖漁業生產區發展協會，2003。
- [5] 冉繁華、陳詩璋，優良水產養殖場安全管理制度之建立，台灣漁業經濟發展協會，2005。
- [6] 吳信輝，「Web Services 技術介紹(九)」，資訊技術雜誌，第 2106 期，30-36 頁，2005。
- [7] 李伊婷，「台灣花卉批發市場交易資訊流的標準化設計」，國立交通大學工業工程研究所碩士論文，2005。
- [8] 李伊婷，梁高榮，「花卉批發市場交易資料庫的標準化」，機械工業雜誌，七月，170-183 頁，2005。
- [9] 周忠信，杜孟儒，郭振堯，羅國書，「利用 RFID 與 BPEL 技術建構及時企業」，機械工業雜誌，十二月，142-149 頁，2005。
- [10] 周耀傑，「歐盟預定於 2005 年導入水產品全程追溯體系」，國際漁業資訊，第 131 期，2004。
- [11] 施威銘研究室，SQL Server2000 設計實務，旗標出版股份有限公司，2000。
- [12] 施威銘研究室，SQL Server2000 管理實務，旗標出版股份有限公司，2000。
- [13] 施重豪，「Java 與 XML 技術導向之交易資訊處理系統」，國立交通大學工業工程研究所碩士論文，2002。
- [14] 胡其湘，「漁產品安全管理新趨勢—產品可追蹤性介紹」，農政與農情，第 143 期，2005。
- [15] 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。
- [16] 張永佳，漁產品產銷履歷資訊建置與整合研究報告，漁業署，2005 年。
- [17] 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。
- [18] 梁高榮，洪欣儀，「電腦整合製造裡 IDEF 技術的整合：IEDF0 與 IDEF1X」，工業工程學刊，Vol. 15, No. 1, pp. 83-94, 1998。
- [19] 梁高榮，「XML 與 RDF 技術介紹」，機械工業雜誌，七月，230-248 頁，2001。
- [20] 梁高榮，農產品交易工程學，國立交通大學出版社，1999。
- [21] 陳永興，「建構 RFID 監控技術應用在物流中心之風險分析」，高雄第一科技大學運輸與倉儲營運所碩士論文，2004。
- [22] 傅家驥，「水產業未來的趨勢：產品可追蹤性」，國際漁業資訊，第 131 期，2004。
- [23] 溫師翰，梁高榮，王清要，「利用網路服務技術建構產銷履歷系統」，機械工業，二月，110-130 頁，2006。
- [24] 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。
- [25] 謝博文、梁高榮，「利用物訊圖/IDEF1X 技術/二維條碼作資訊系統再工程」，亞太工業工程暨中國工業工程學會論文集，508-513 頁，1994。
- [26] 顧君暉，「RFID 與 EPC Network」，機械工業雜誌，十二月，47-55 頁，2005。
- [27] Arduser, L. and Brown, D. R., HACCP & Sanitation in Restaurants and Food Service Operations: a Practical Guide Based on the FDA Food Code with Companion, Atlantic Pub. Group, 2005.
- [28] Bernstein, P. A., "Synthesizing Third Normal Form Relations from Functional Dependencies,"

- ACM Trans. on Database Systems, Vol. 1, No. 4, pp. 277-298, 1976.
- [29] FIPS 183, Integration Definition for Function Modeling (IDEF0), National Institute of Standards and Technology, 1993.
  - [30] FIPS 184, Integration Definition for Function Modeling (IDEF1X), National Institute of Standards and Technology, 1993.
  - [31] Garfinkel, S. L., "Adopting Fair Information Practices to Low Cost RFID Systems," Ubiquitous Computing Workshop, 2002.
  - [32] <http://jtds.sourceforge.net/>, jTDS Project Website.
  - [33] <http://uddi.ibm.com/ubr/>, IBM UDDI Business Registry Web Site.
  - [34] <http://xml.fujitsu.com/jp/tech/uddi/> Fujitsu UDDI Website.
  - [35] <http://ws.apache.org/axis/>, Axis Website.
  - [36] <http://www.ca.com>, Computer Associates Company Website.
  - [37] <http://www.epcglobalinc.org/>, EPCGlobal Website.
  - [38] <http://www.intermec.tw/>, Intermec Technology.
  - [39] <http://www.w3.org/TR/soap/>, W3C SOAP Specification Website.
  - [40] [http://www.winpesim.de/petrinet/e/hpsim\\_e.htm](http://www.winpesim.de/petrinet/e/hpsim_e.htm), HPSim Program Free Download Website.
  - [41] <http://www-128.ibm.com/developerworks/webservices/library/ws-bpelcol1/>, IBM BPEL4WS.
  - [42] ISO 22000: Food Safety Management Systems - Requirements for Any Organization in the Food Chain, ISO, 2005.
  - [43] Kroenke, D. M., Database Processing: Fundamentals, Design & Implementation, Prentice Hall, 2004.
  - [44] Kulkarni, N., Kumar, S., Mani, K., and Padmanabhuni, S., "Web Services: E-Commerce Partner Integration," IT Pro, pp. 23-28, March-April 2005.
  - [45] Monson-Haefel, R., J2EE Web Services, Addison Wesley, 2004.
  - [46] Murata, T., "Petri Nets: Properties, Analysis and Applications," Proceedings of the IEEE, Vol. 77, No. 4, pp. 541-580, 1989.
  - [47] Pasley, J., "How BPEL and SOA Are Changing Web Services Development," IEEE Internet Computing, pp. 60-67, May-June 2005.
  - [48] Sara, M. and Wallace, C., HACCP: A Practical Approach, Aspen Publishers, 1998.
  - [49] Werner, C., Buschmann, C., and Fischer, S., "WSDL-Driven SOAP Compression," 18 International Journal of Web Services Research, 2(1), pp. 18-35, Jan-March 2005.