

南 華 大 學

資訊管理學系碩士論文

以 Web Services 為基礎之動態商業流程協調

整合架構

**Development of a Web-Services Based
dynamic and collaborative business process
Application and Integration Framework**



研 究 生：黃瓊慧

指 導 教 授：王昌斌

中 華 民 國 九 十 三 年 六 月

南 華 大 學

資訊管理學系

碩 士 學 位 論 文

以 Web Services 技術為基礎之動態商業流程協調整合架構

Development of a Web-Services Based dynamic and
collaborative business process Application and Integration

Framework

研究生：黃 瓊 慧

經考試合格特此證明

口試委員：

陳 永 成
鍾 國 光

指導教授：

王 登 科

資訊管理學系
系主任
吳光國

所 長：

口試日期：中華民國 九十三年 六 月 三十 日

南華大學碩士班研究生
論文指導教授推薦函

資訊管理系碩士班 黃瓊慧君所提之論文
以 Web Services 為基礎之動態商業流程協調整

合架權

係由本人指導撰述，同意提付審查。

指導教授

吳錫

93年 2月 13日

誌 謝

進入南華校園兩年的碩士研究生涯終於在論文完成之際劃下句號。這段時間，承蒙所上老師於學業上的指導及平時之教誨皆使我受益良多，也深深銘記在心。

首先，由衷感謝恩師王昌斌教授兩年多來的不辭辛勞的指導，不僅是生活上之指引，研究上更是提供各項建議和寫作技巧，也給予相當程度的支持鼓勵，讓我能自由發揮自我潛能也不斷提昇問題解決能力。再者，十分感謝口試委員成大製工所陳裕民教授和所上鐘國貴教授對於本論文之建議與指正，讓我進一步釐清研究中更細微之邏輯思緒。

感謝系上助理伊汝平時對於行政事務上之協助，以及書慶學長對於研究上之建議與協助，也感謝所有同學和學弟妹的鼓勵，在平常的討論中讓我受益良多，此外，也感謝祥瑞在研究資源上之支援及平時對於學問上之鼓勵。

最後，我要感謝我的家人尤其是我的父親，讓我在經濟上沒有煩惱更是給予我最大的鼓勵和支持，也讓我有更大的動力和自信能繼續朝自己的目標前進。

黃瓊慧 謹識
于 南華大學資管所
九十三年六月

以 Web Services 為基礎之動態商業流程協調整合架構

學生：黃瓊慧

指導教授：王昌斌 博士

南華大學資訊管理學系碩士班

摘 要

目前電子市集已進入協同商務階段，主要是希望企業合作夥伴間，由產品設計到最後產品交貨甚至績效評估等一連串的營運流程，都能透過網際網路加以整合、同步作業，並依據企業內外部資源作最佳化安排，快速反應市場變化，提高競爭優勢；因此，商業流程整合與自動化勢必成為未來企業進行協同整合時的一大重點。

不過在協同環境下進行跨企業流程整合時通常會面臨下列幾項問題：

1. 缺乏產業標準、市場提供解決方案過多且複雜；2. 異質性分散式系統間跨平台溝通問題；3. 整合成本等問題；目前諸如 Rosetta NET 標準，雖然提供開放、產業通用的流程標準，但較屬於制式化的互動關係，且採用廠商大多為高科技產業。

針對前述問題，本研究擬採用的 Web Services 技術，主要是建構在 XML、SOAP、WSDL 等標準協定上，並採用 HTTP 通訊協定，不僅能解決異質性分散式系統間流程難以整合的問題；有一定程度電腦化的企業都能採用，此外，由於 Web Services 能以低廉代價建構所需的應用程式模組，不因採用不同軟體廠商的整合軟體而額外付出 Adapter 的軟體授權費，能降低企業經營的成本問題。因此本研究即以 Web Services 技術為基礎，提出一中介的流程整合平台架構，運用 Web Services 技術跨平台及訊

息標準化的特性，及 BPEL4WS 流程描述語言支援圖形化流程及流程結構化等性質，將企業間所提供的服務封裝成 Web Services 模組，依特定商業目標和商業協同文件，將一連串服務元件在此平台架構中進行整合與任務協調，達到異質性系統間 B2B 流程整合與及時性商業行為的協同合作。

關鍵字：網路服務、商業流程執行語言、商業流程

Development of a Web-Services Based dynamic and collaborative business process Application and Integration Framework

Student : Qiong-Hui Huang

Advisors : Dr. Chin-Bin Wang .

Institute of Information Management
Nan-Hua University

ABSTRACT

Collaborative Commerce has been the the trend of e-market. Business Process Management and Integration will be the most important part of collaborative enterprise. The major problem is that the heterogeneity of system platform existing among collaborative partners, multifarious solutions in the market and Integration Cost.

In this paper, we construct an dynamic business process integration framework by using Web Services and BPEL4WS. The method of web service is based on XML together with some standard protocol, such as SOAP, WSDL ... We encapsulate the service which provide by collaborative partners as Web Services models. Then, we put series web services models in the framework to proceed to process connect and integrate according to business protocol document and business purpose. Finally, it can achieve the goal of integrating heterogeneity B2B business process and instantaneously business collaboration.

Keywords : Web Services, BPEL4WS, Business Process

目 錄

書名頁	
博碩士論文授權書	
論文指導教授推薦書	
論文口試合格證明	
誌謝	
中文提要	
英文提要	
目錄	
表目錄	
圖目錄	
第一章 緒論	1
第一節 研究背景	1
第二節 研究動機	2
第三節 研究目的	3
第四節 研究範圍與限制	4
第五節 研究流程與方法	4
第二章 文獻與核心技術探討	7
第一節 協同商務	7
壹、協同商務興起與定義	7
貳、協同商務的模式、分類與作業方式	8
參、協同商務運作的困難點	11
第二節 商業流程整合與自動化	12
第三節 Web Services 技術	14
壹、Web Services 運作架構與關鍵技術	14

貳、Web Services 運用於企業的關鍵價值	19
參、小結	22
第四節 商業流程執行語言-BPEL4WS	22
壹、BPEL4WS 簡介	22
貳、BPEL4WS 的特點	23
參、BPEL4WS 的結構與使用方法	26
肆、BPEL4WS 應用現況與法展	27
第五節 統一塑模語言-UML	27
壹、UML 簡介	27
貳、利用 UML 對 BPEL4WS 系統塑模	28
第三章 動態商業流程協調整合架構	31
第一節 系統架構發展步驟	31
第二節 系統需求分析	32
第三節 動態商業流程協調整合架構設計	34
壹、電子化商業流程運作環境架構	34
貳、流程整合架構內部元件定義與互動	38
參、流程建構最基本之內部元件分析	39
肆、動態商業流程協調整合架構設計	43
第四節 小結	44
第四章 以 UML 塑模系統架構	45
第一節 系統架構使用案例分析	45
第二節 系統元件之分析與設計	56
壹、使用者操作管理介面類別	56
貳、整合端類別	57
第三節 小結	60

第五章 跨企業商業流程整合架構設置	61
第一節 系統開發環境與佈署架構	61
壹、開發環境	61
貳、系統架構佈署	63
第二節 系統建構與展示	65
壹、系統建構	65
貳、系統展示	66
第六章 結論與未來研究方向	74
第一節 結論	74
第二節 未來研究方向	74
參考文獻	76

表 目 錄

表 1	協同商務模式	8
表 2	UML 系統塑模觀點使用圖形及適用對象表	30
表 3	Microsoft .NET 與 J2EE 開發環境比較表	62

圖 目 錄

圖 1	研究流程圖	6
圖 2	Web Services 操作角色及行為	14
圖 3	Web Services 技術堆疊圖	16
圖 4	Web Services 對企業影響項目比例圖	20
圖 5	Web Services 服務概念圖	25
圖 6	系統架構五個相關觀點圖	29
圖 7	以 Web Services 為基礎之跨企業流程整合運作圖	35
圖 8	Web Services 為基礎之商業流程協調整合架構圖	37
圖 9	服務元件互動圖	38
圖 10	WSDL 資訊模型圖	40
圖 11	元件內部功能分析圖	42
圖 12	動態商業流程協調整合架構圖	44
圖 13	主要系統架構案例示意圖	45
圖 14	買方使用案例示意	46
圖 15	賣方使用案例示意	47
圖 16	第三方使用案例示意	47
圖 17	服務元件分析使用案例圖	49
圖 18	協同商業協議規則分析使用案例	51
圖 19	商業流程部署使用案例	53
圖 20	商業流程執行使用案例	55
圖 21	商業流程例外處理使用案例	56
圖 22	使用者操作介面泛化類別圖	57
圖 23	服務元件瀏覽登錄與分析類別圖	57
圖 24	協同商業協議規則分析類別圖	58

圖 25	商業流程部署類別圖	58
圖 26	商業流程執行類別圖	59
圖 27	例外補償機制類別圖	59
圖 28	系統開發架構圖	64
圖 29	流程整合系統平台部署圖	65
圖 30	流程整合架構平台主畫面	67
圖 31	登入服務夥伴元件 WSDL 網址畫面	67
圖 32	流程參與元件之 WSDL 分析	68
圖 33	商業文件檔案建檔	69
圖 34	商業文件檔案查詢	69
圖 35	流程編輯器選擇與啟動	70
圖 36	抽象流程 WSDL 文件定義	70
圖 37	CRMService 流程參數定義與佈署	71
圖 38	CheckoutFlow 流程參數定義與佈署	71
圖 39	買方下單狀態	72
圖 40	流程監控畫面及流程狀態	72
圖 41	流程完全執行完之狀態	73
圖 42	信用卡資料之查詢畫面	73

第一章 緒論

第一節、研究背景

由於資訊科技不斷地進步及網際網路的興盛擴張，使得電子化企業間的互動模式也逐漸產生變化；台灣國際電子商務中心在 2002 年 4 月的 EC 研究報告中指出，企業電子化/電子商務四階段，最終目標就是要達成自動化之商業流程整合 (Automated Inter-Business Processes) 與流程分享 (Sharing Processes)，因此，企業除了在經營角度上做變革整合內部資源外，尚須擴及至外部企業合作夥伴間的協同運作，將企業內、上下游供應商及合作夥伴間的流程和資訊整合，建立一透明的共享流程。

目前電子市集已逐漸演變至協同商務階段，亦即企業本身、企業合作夥伴或客戶間為了共同目標，由產品設計到最後產品交貨乃至於行銷策劃等一連串營運流程，都能透過資訊科技加以整合、同步作業，並依據企業內外部資源作最佳化安排，快速反應市場變化，提高競爭優勢；因此，企業間商業流程整合與自動化將成為未來企業進行協同作業時的一大重點。

在協同作業的環境下，企業逐漸希望藉由 IT 技術將商業流程整合起來，以發揮交易合作中種種商業行為(如：下單、存 / 補貨管理等)，並與上、下游供應商系統串接起來，使其在單一網站上呈現出流程各階段的工作進度。

由於企業漸漸認知到商業流程管理的重要性，軟體廠商也越來越強調商業流程管理並提供多種的解決方案，加上 XML、Web Services 等新興技術愈趨成熟，商業流程管理 (Business Process Management, BPM) 將愈受重視，在 ZapThink 一份研究報告中指出，在新興網路服務(Web services)市場當中，以商業流程為主的新技術，可能將取代目前應用程式整合的市場。因此，如何在眾多解決方案及整合工具中做最適選擇，已是現今企業極需面對問題之一。

第二節、研究動機

根據 Gartner Group 研究調查[52]，到了 2005 年，擁有協同商務能力的供應商與客戶，與沒有協同商務能力的企業競爭時，將能贏得超過八成的商機，所以協同商務對於企業未來生存有相當程度的重要性。由於協同商務強調的是依據共同目標，透過資訊與流程的分享整合，達成協同商務的發展；因此流程整合市場也開始蓬勃發展。

不過在協同環境下進行跨企業流程整合時通常會面臨下列幾項問題[36][37]：

1. 缺乏產業標準：目前市場上，諸如 Rosetta NET、ebXML 等標準，雖然提供開放、產業通用的流程標準，但屬於較制式化的互動關係，且跨類別企業間之內部訊息傳遞標準等亦在草擬階段。
2. 市場提供解決方案過多且複雜，導致企業不知如何選擇。
3. 異質性分散式系統間跨平台溝通問題：企業合作夥伴間普遍存在系統平台異質性問題，即使在同一企業內部也可能面臨新舊系統間無法互相溝通和分享資訊等繁瑣問題；目前軟體業界大多採用如：DCOM、RMI、CORBA 等中介技術來支援此類分散式系統間的服務溝通與協調問題，不過由於此類系統整合邏輯多被封裝在分散的 adapter 中，不僅不易調整，亦無法對企業間的商業互動運作進行適當且即時的監控。
4. 協同作業下商業流程高度變化性：由於商務環境瞬息萬變，如何配合企業夥伴即時作改變、快速反應，是一項急需解決的問題。
5. 整合成本問題。

針對前述問題，本研究擬採用近年由軟體業界推出的 Web Services 技術，主要是透過如：XML、SOAP、WSDL 等標準協定，將企業間的服務加以描述封裝，由於採用 HTTP 通訊協定，不僅能解決異質性分散式系統間應用程式難以溝通整合的問題；並適用於一般電腦化企業，此外，由於 Web Services 具有以低廉代價建構所需的應用程式模組特性，不因採用不同軟體廠商的整合軟體而額外付出 Adapter 的軟體授權

費，也能降低企業經營的成本問題。

不過由於 Web Services 缺乏新的商業流程管理規範，使得 Web Services 無法在企業商業往來間的互動上做有效管理和維護商業交易狀態，針對此，2002 年 8 月，BEA Systems、International Business Machines Corporation、Microsoft Corporation 等軟體大廠遂提出「商業流程執行語言」(Business Process Execution Language For Web Services, BPEL4WS)；做為 Web Services 調用、操縱資料、拋出故障或終止一個流程等不同活動，進一步將其連接起來，從而創建出複雜的商業流程。BPEL4WS 為 WSFL 和 XLANG 技術的綜合體，不僅為一套完善的技術規範，目前也以發展出可實際運作的軟體。

因此，本研究擬以 Web Services 技術為基礎，加入 BPEL4WS 商業流程執行語言，並利用統一塑模語言(UML)分析塑模以 BPEL4WS 為商業流程執行語言之企業協同作業夥伴間的互動流程和交易行為，建構一動態商業流程協調整合架構，期能整合異質系統企業間交易互動元件模組，在此架構中進行任務協調與流程整合，以解決上述問題

第三節、研究目的

本研究主要為設計建構一個以 Web Services 技術為基礎的商業流程協調整合架構，主要目的分為下列幾點：

1. 運用 Web Services 技術跨平台、訊息標準化等特性，將企業間所提供的服務封裝成 Web Services 模組；
2. 建構一可行的商業流程協調整合架構，以 BPEL4WS 作為商業流程中 Web Services 元件串接與協同運作之實現語言，並採用 UML 對所設計之 BPEL4WS 系統架構進行塑模，再依特定商業目標和商業協同文件，將一連串互動交涉服務元件，在此架構中進行整合與任務部署；
3. 最後再設計一整合操作平台，方便使用者能在此平台下達成不同流程的任務協調

與整合，達到異質性系統間商業流程整合與及時性商業行為的協同合作。

第四節、研究範圍與限制

本研究的範圍與限制主要分為以下幾點作一討論：

1. 研究領域部分：

本研究前提設定為，企業間已透過 UDDI 註冊機制搜尋到合適之企業合作夥伴，並依雙方合作之共識擬定商業協同協議和簽署協同商業文件，因此，企業間合作夥伴搜尋過程不在本文討論範圍中。此外，本研究探討領域為協同作業環境中商業流程整合問題，擬定範圍在企業間(B2B)的常用商業流程進行整合與協調的探討，期望能藉由本研究所設計之整合架構進行企業間商業流程整合與執行。

2. 流程整合架構之內部功能部份：

為簡化問題與掌握實際資源，本研究在系統展示部份以一簡單訂單流程確認與查詢流程為例以驗證本架構之可行性；關於實體商業環境中商業流程之複雜性與流程種類，則可依據商業目標與協議文件修改流程屬性與訊息等定義，以符合當時交易情節。

3. 研究發展部分：

由於本研究主要針對研究動機中幾個問題點進行分析與架構設計，並以 Web Services 技術作為解決方法之基礎。目前在產業間的整合標準主要有 Rosetta Net、ebXML 等，不過，利用 Web Services 技術與上述標準結合之探討尚在萌芽階段，例如：商業文件規範、訊息傳遞格式等；若能將現有標準技術加入本研究架構中進行進一步整合與探討，將會更加完整。

第五節、研究流程與方法

本研究基於前述研究動機蒐集相關文獻與核心技術探討，了解相關領域資訊與問

題解決方法，再進行相關軟體的使用與學習，並以物件導向方式分析設計系統架構與規範商業流程，同時進行實作與系統測試，最後在檢討系統缺失與探討未來發展方向。

本研究之研究流程主要如下：

1. 確定主題

確定研究範圍、目的等。

2. 文獻蒐集與核心技術探討

基於前述研究動機進行協同作業、商業流程、Web Services 技術及仲介整合平台架構等相關文獻進行蒐集與研讀，再對 UML 塑模語言及 BPEL4WS 等相關軟體進行相關軟體學習與使用。

3. 問題分析與解決

針對相關文獻所蒐集的資料及研究動機中商業流程整合所遭遇之問題進行分析，擬定解決方案。再針對建構商業流程整合協調機制所需的功能性加以分析與整理，進行下一步的架構設計。

4. 架構建立

針對前述功能和需求建立一以 Web Services 為基礎的商業流程整合架構。以逐步分析方式，從大環境之設計到細部元件定義，以便說明這些元件如何在交易中進行流程的串接和溝通，並以 BPEL4WS 作為調整合架構商業流程串接與協同運作之實現語言。

5. 系統分析與設計

在此部份，我們針對上一步驟之功能需求分析定義，以 UML 進行系統塑模與分析，並於下一階段進行系統建構。

6. 系統建構

針對前述分析結果，以一簡單訂單流程確認與查詢流程為例以驗證本架構之可行性。

7. 系統測試與評估

對於系統進行測試並加以評估。

8. 結論與建議

根據研究結果進行結論的撰寫並提出未來研究方向。

本研究流程圖則如圖 1 所示：

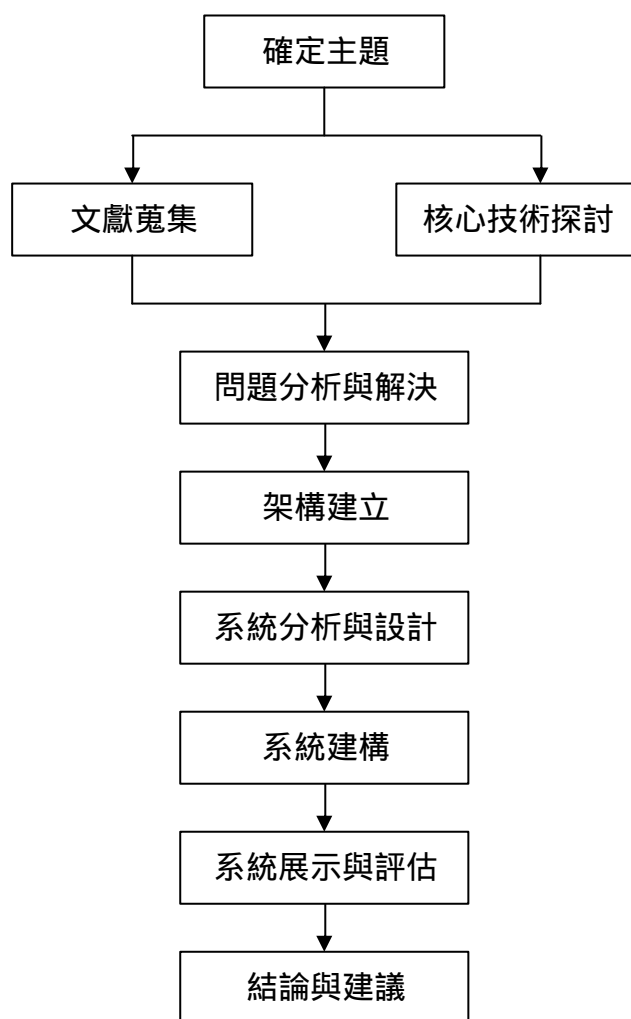


圖 1. 研究流程圖

第二章 文獻與核心技術探討

第一節、協同商務

壹、協同商務興起與定義

由於資訊科技不斷地進步及網際網路的興盛擴張，使得企業間的互動模式也逐漸產生變化；早期的企業資源規劃(Enterprise Resource planning, ERP)，所強調的是企業內部整合，90年代的供應鏈管理(Supply Chain Management, SCM)，則將重心轉移至外部上下游廠商間的互動整合，不過面對市場環境不斷變化，這樣的互動模式似乎已顯不足。因此，在企業合作團隊間，不僅需要和企業內部資源及活動密切結合，更需與外部的供應商或客戶形成傳遞即時且透明的共享資訊流程，才能形成快速反應的電子化企業；也因此協同商務(Collaborative Commerce)的概念也逐漸成形。

根據 Surgency(2001)[34]，為協同商務下的定義為：「協同商務主要應用在虛擬企業間，以合作為導向之商業模式。」；林章鈞(2002)[4]，認為協同商務為：「旨在利用網際網路處理企業與合作夥伴間的各項電子交換，更進一步將雙方的營運流程整合起來，可以是直接將彼此的應用系統串連，達到全面自動化。」；陳世運(2001)認為協同商務是[5]：「強調從產品的設計端、計畫研擬端、生產過程、產品交貨、財務處理、甚至是最後的成效評估端等，都透過電子市集讓交易的夥伴能夠同步作業。」；Michael Harris(2002)[28]所定義的協同商務：「流程管理和整合技術歷史性的一刻。」；針對前述對協同商務的定義，毛立青(2003)[1]以整體性之考量和通盤思考模式定義出協同商務為：「企業(組織)本身、企業(組織)間或企業(組織)與消費者間為了共同目的(如產品開發、生產、供應鏈管理、預測、物流、行銷等)而協同合作，在互信的基礎下，透過運用資訊科技協助完成此目的。所運用之資訊科技必須具備流程管理能力與資訊整合能力。」

貳、協同商務的模式、分類與作業方式

現行最常見的協同商務有三種，如下表 1 所示：

表 1. 協同商務模式

項目	電子市集	供應鏈管理	產品週期管理
合作模式	池塘型合作 Pooling Collaboration	流程型合作 Process Collaboration	互動型合作 Reciprocal Collaboration
應用	電子市集 電子拍賣 電子採購	供應鏈管理	產品週期管理
主要供應商	Commerce One Ariba	I2 Orade	Metaphase(EDS) Matrix One
風險	害群之馬	能力不夠	默契不佳

資料來源：蕭瑞麟、「協同商務為 B2B 合作帶來新境界」

管理雜誌 332 期

1. 交易型模式 (exchange model)：

此模式中企業的主要合作方式是資源的交換，為了取得更有經濟效應的交易，企業往往以拍賣方式競價，以最少價錢取得最好的原料。這種交易型模式又可稱之為池塘式合作 (pooling collaboration)，即「各取所需、分道揚鑣」的原則（除非是有特定的買賣關係）。此合作模式的缺點就是怕組成份子不良。

2. 流程的合作 (Process Collaboration)：

此類合作是將供應商列入供應鏈中之一環，合作模式之特點在追求有彈性的資源整合；透過資訊科技，使每一家廠商都有貢獻出最優秀的服務。此合作模式所擔心的問題是是供應鏈上各公司的服務能力之良窳，若任一家公司出問題，則整個供應鏈可能就因此停擺。更大的風險是供應鏈中前後的廠商是否能建立互信、分享資訊以做到最密切的整合。

3. 互動型合作 (Reciprocal Collaboration)：

此合作模式主要利用資訊系統來管理跨國界的產品開發；這種合作模式最常見

於較複雜的產品開發與設計，例如汽車、航空引擎及積體電路。由於產品本身複雜度高，而且上市時間又緊迫，因而企業須利用資訊系統來連接位於各國的工程師，使其能同時跨地區、跨時域進行產品設計。而此模式所需擔心的則為合作夥伴間的默契問題。

在協同商務的分類方面，Meta Group 將協同作業分為四大功能[56][10]：

1. 設計協同作業：

涵蓋一切非連續性製造產品（Discrete Manufactured Product）以及客製化生產（Configured to Order）產品。這類的協同可以是簡單、樣本的傳送而已，也可以是複雜的、需透過協同工具來追蹤並管理的工作流程。

2. 行銷/銷售協同作業：

指與轉銷商、配銷商等通路夥伴之間的協同作業，著重彼此間的資訊共享，及訂單／價格／品牌管理等流程的共享，並且提供可承諾的資訊。這種形式的協同作業包含建立一個共同品牌的虛擬展示空間，讓從製造商到零售商之間的各通路可以協力支援終端消費者對產品或服務的需求。

3. 採購協同作業：

數家買主結合較大的數量來採購某些產品或服務，以求降低採購成本的協同作業。採購協同作業可以是公開的電子交易市集，也可以是個別企業結合各事業單位對成品和原料的所有需求，一致對外採購；相對地，供應商也可以結合彼此力量提供產品或服務，方便買主一次大量採購，毋須同時向數家供應商下訂單。

4. 規劃/預測協同作業：

背後的驅動力主要是來自 VICS 所推動的「協同規劃、預測與補貨（Collaborative Planning, Forecasting, and Replenishment, CPFR）」模式，CPFR 模式制定了 9 大步驟協助企業與企業之間（尤其是從供應商到零售商之間）如何在規劃、預測與補貨等方面進行合作。

由於協同商務的實現在技術上必須透過企業間的整合(B2Bi),而 B2Bi 根據整合模式基本上可分為四類[10]:

1. 企業間的資料交換

過去企業間最普遍的整合模式就是電子資料交換(EDI),透過增值網路進行企業間的資料交換;由於使用增值網路加上不是即時性的資料交換,在成本及效率上都顯的不彰。近年,Internet 的出現,讓企業從事資料交換的門檻變低,加上 XML 的標準資料格式,將有助於此類整合模式。

2. 企業間應用程式整合

此模式是由企業應用整合(Enterprise Application Integration, EAI)各廠商將產品應用層面擴充所導入的應用觀念,此種模式可以將企業的應用程式向外和其他企業的應用程式做整合,通常將交易夥伴或客戶的應用程式整合在一起,以便形成一個價值鏈。

企業應用整合所使用的方法是以中介軟體(middleware)架構與分散式技術將不同應用程式作安全有效率的整合。主要價值在於有效協助企業達到風險、軟體開發和維護成本的降低,並進一步整合各供應商間的合作關係。

3. 封閉式企業間流程整合

當企業雙方合作時,這些互動的先後關係及組成便形成所謂的「流程」。流程的整合必須依據共同的 Business Process Model 來定義次序、組織結構、事件、執行邏輯以及不同企業間系統內的訊息移動等,流程的整合是策略性的思考,因為它可以加強企業和夥伴的互動關係,展現形成彈性整合後的優勢價值。

在封閉式流程整合中,企業內部本身的事件會形成一流程,並透過流程仲介(Process Broker)整合另一企業內的流程。由於此類整合並非完全透通,所以為一封閉式的流程整合模式。

4. 開放式企業間流程整合

開放式的流程整合具有分享流程的觀念,它之間的流程為「點對點」的夥伴

關係 (Peer Level)，每個企業都可以自行管理內部的流程，但整體上又相互整合。而且企業可以自行決定，那些與流程有關的事件需放在共用流程內與大家分享，那些是僅限於內部的事件而不開放。

由於目前企業競爭環境逐漸邁入一個新紀元，合作企業間強調的是能依據共同目標，透過資訊與流程分享/整合，達成即時性的交易合作，因此重心也將慢慢轉移至開放式企業間流程整合。

參、協同商務運作的困難點

企業間協同運合必須在下面前提下才能支援；在流程方面，必須對交易夥伴的工作流程和採用的商業規則，作清楚的描述；組織人員部份，企業必須要提供無溝通障礙的環境，鼓勵員工直接溝通，並作出決策；而在技術方面企業也必須先建構起適當的資訊基礎架構。

而協同運作的困難點在於[4][10]：

1. 合作交易夥伴間商業模式和工作流程不同。
2. 企業內及企業間異質性系統的問題。
3. 企業間整合的引導技術，必須能處理不同企業間複雜的文件交換、深度整合或是多層次流程管理工作。
4. 市場上所提供的解決方案過於複雜；技術上缺乏共通標準。
5. 內部與外部合作廠商彼此協調與平衡關係的維持。
6. 成本問題。

在面對競爭如此激烈的今天，企業必須導入協同商務這項新的競爭武器，藉由整合企業內外部的資源，例如：供應商、消費者回饋的資訊、及合作夥伴所蒐集的訊息等，進而縮短產品的上市時間、即時的回饋消費者的需求。同時，企業也必須要注意以上導入協同商務的困難點，或找出最適合的解決方案，以順利導入協同商務，進而提升競爭力。

由於推動協同商務所涉及的範圍極其廣，而本研究所擬定範圍主要針對企業合作夥伴間商業流程整合問題；關於流程整合部分將在下一節中會作一詳述。

第二節、商業流程整合與自動化

Medina-Mora et al.(1992)指出[26]在組織中的流程可以區分為三種：

1. 實體流程(Materiel process)：

人員的活動是起源於實際的物體，由於實體的移動、改變而產生活動，傳統的工廠製造流程便是典型的例子，工廠裝配人員的活動是轉移、組合那些實質的零組件。

2. 資訊流程(Information process)：

當「資訊勞動」時代來臨時，實體流程無法應付每天的活動，由於電腦的使用，所有實體工作的界限變得模糊不清且難以辨別，人們的工作變成面對螢幕以 Key in 方式來進行。而學者及資訊科技提供者也發展了複雜的方法來分析資訊的流動，透過資料流分析、資料庫儲存及檢索、交易處理流程、網路溝通等技術使得資訊流程更有效率。

3. 商業流程(Business Process)：

是以市場導向的觀點來看組織間的活動，為的是執行企業所簽訂的合約或滿足特殊消費者的需求，因此商業流程相較資訊流程及實體流程而言，是屬於較高層的概念。本研究欲探討之流程整合範圍即屬商業流程範疇。根據 Davenport 和 Short(1990)[14]對商業流程(business process)所下的定義：「商業流程是指由實體(Entities)、物件(Object)和活動(Activity)三方面所組成，用來達成特定企業目標的一組相關業務活動(Task)。」；Davenport(1990)並認為「商業流程有兩個重要的特徵：(1).流程要有其服務的對象(2).流程可跨越組織疆界。」；Michael Hammer 與 James Champy 對商業流程所下之定義是「企業集合各種原

料，製造顧客所需產品的一連串活動。」；綜上所述，商業流程可解釋為：「一個供應者與流程的負責者資源投入與產生附加價值的過程，此過程中包含許多業務活動的組合，而產出的結果必須以滿足顧客需求為目的。」

商業流程管理，或是工作流程管理，是讓組織裏各部門原本以人力及公文傳遞的過程，利用 IT 技術使其自動化。產品供應商 Ultimus 亞太區副總裁 Keith Wannop 指出，沒有良好的 BPM，企業無法快速而有效完成專案審查或採購申請等流程。由於傳統跨企間應用程式系統都是點對點的整合，再慢慢發展為如：DCOM、CORBA 等中介軟體整合方式，不過當遭遇到系統平台異質性問題時，往往需要投入額外人力進行特定介面轉換的開發和維護，造成效率不彰與資源浪費。

因此，現今熱門討論的整合方式是利用服務導向架構(SOA)來進行整合；所謂服務導向的架構，是在建置軟體系統時，讓資料在多應用軟體和資料來源間，用比較容易分享的方式來進行整合與開發，這和傳統在應用程式直接寫死的資料連接方式不同，服務導向的設計，可以使用較具有彈性的描述語言，如：XML 或網路服務來交換資料。這種鬆散式耦合（loosely couple）設計的好處，是在變更單一軟體模組時，不用去動到所有相連結的軟體系統。此類服務導向的流程軟體可以將資料整合在特定的商業流程當中。負責 ZapThink 研究報告的 Ronald Schmelzer 表示，「當把所有流程集中起來的時候，其實就是在做整合。」

由於在此種架構下，不論是舊的或新的，都希望能夠透過 Web Services 的包裝，成為可以隨取即用的 IT 資產；因此專為 Web Services 所設計的商業流程管理工具也不斷推陳出新。例如：IBM、微軟、BEA 也努力催生商業模型標準，聯合制定的 Web Services 商業流程執行語言（BPEL4WS）。

根據 Aberdeen Group 的報告[53]，2002 全球 PBM 的市場總值達到 25 億美元；Aberdeen 並預估 2004 年之前是 BPM 的高成長期，到 2005 年將擴大為 63 億，商業流程整合(BPM)之重要性有此可見一般。

第三節、Web Services 技術

壹、Web Services 運作架構與關鍵技術

Web Services 是在現今商業環境中一項資訊技術的重大改變，Web Services 原本是要解決分散式應用程式整合的問題，不過在現今商業行為與資訊技術緊密相扣的同時，Web Services 的影響力已逐漸滲透至現有的商業規則與經濟模式中。Web Services 是以服務為導向的架構(SOA)，希望將具有價值的應用程式或企業所提供的服務都能夠透過 Web Services 的封裝，成為可以隨取即用的 IT 資產。Web services 和一般分散式系統不同處在於”標準”和”整合”的概念，這兩個概念則正好符合目前企業應用整合(EAI)和 B2B 的需求[42]。

Web Services 的運行架構主要如下圖所示：

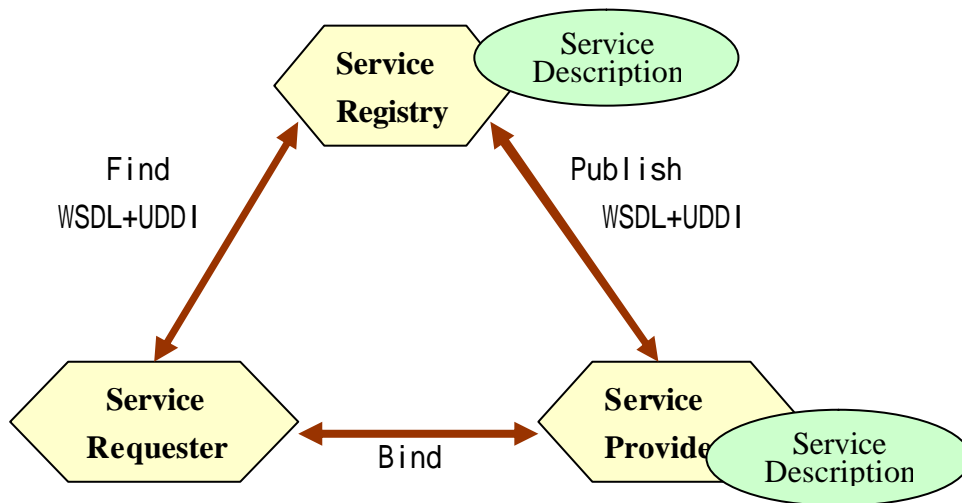


圖 2. Web Services 操作角色及行為

(資料來源：IBM Web Services website)

1. 服務提供者(Service Provider):

服務提供者負責供應商業處理功能讓別人來使用。藉由使用 WSDL，服務提供者可描述所提供的服務中所含的功能、要使用此功能所需要的輸入資料，以及預

期的輸出結果，並提供 URL 連到這支服務程式以便可以真正去呼叫它。最後，服務提供者透過 UDDI 把這些資訊公告出去。

2. 服務需求者(Service Requester):

當服務的需求者希望透過 Web Services 的機制來獲取服務時，服務要求者先送利用 SOAP 訊息傳遞格式化的查尋指令給 UDDI 服務登錄資料庫，之後可以根據查詢到的服務提供者資訊獲得需要的 Web Services，再向服務者建立直接連線，以啟動所需的服務。

3. 服務登錄庫(Service Registry):

這個資料庫的功能就是 UDDI，讓服務提供者能將服務內容公告出來，服務需求者再對服務登錄資料庫提出特定的商業功能或公司名稱來查詢，服務登錄資料庫則將所找到的相關資料回應給服務需求者。

Web Services 主要運作行為依序有：[45]

1. 描述：要讓服務需求者知道 Web Service 提供服務的內涵，以及和其溝通的方式，需要有一種描述 Web Service 的語言。
 2. 發佈：透過某種註冊機制將 Web Service 的描述資訊登錄於某個公開的 Web Service Registry。
 3. 尋找：服務需求者向 Web Service Registry 送出查詢訊息，取得 Web Service 的繫結(Binding)資訊。
 4. 繫結：根據取得之 Web Service 描述資訊，建構一個連結到該 Web Service 的代理程式。
 5. 呼叫：透過 Internet 傳遞適當訊息參數，呼叫該 Web Service。
 6. 整合：整合其他 Web Services 建構出自己的應用程式或是新的 Web Service。
- 當然，Web Services 能在開放的 Internet 動態地發佈、搜尋、執行服務，必定得依存一些技術標準，Web Services 的技術堆疊則如下圖所示：

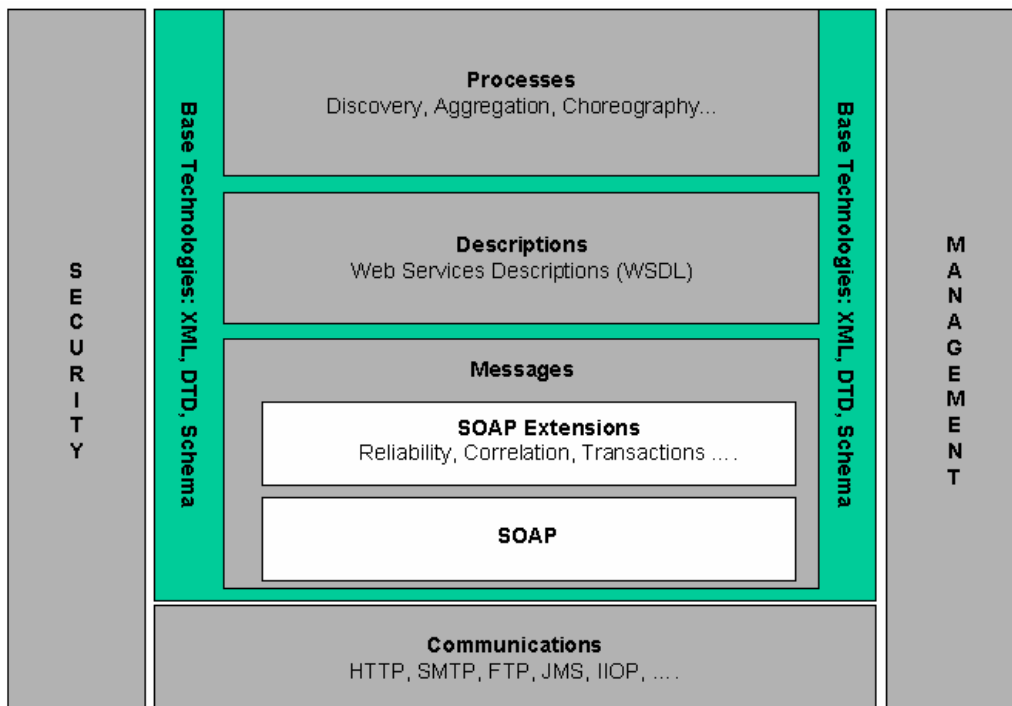


圖 3. Web Services 技術堆疊圖

(資料來源：IBM Web Services website)

最主要的組成技術為：

1. XML(eXtensible Markup Language)

XML 是 SGML 的精簡版，繼承 SGML 自定標籤的優點，並刪除一些 SGML 複雜部份，在功能上補足 HTML 標籤不足，擁有更多的擴充性。XML 並非用來編排內容，而是用來描述資料，使用者可依本身需要義行定義描述資料所需的各種標籤。

由於 XML 具有延展性 文件自我描述性 結構化文件、可描述其他語言等優點；因此 XML 非常適合用來作為不同產業界和特定應用領域發展各自標準的文件和訊息，以利於資訊的交換以及處理和相關衍生性資料的加值服務。

2. SOAP(Simple Object Access Protocol)

SOAP (Simple Object Access Protocol；簡單物件連結協定)，是一個類似 DCOM、CORBA 等分散式物件通訊協定，不同的是它不是只設計用來針對某種物件技術的協定，因此 SOAP 能讓不同物件程式可以跨平台透過 Internet 呼叫。當用戶端要呼叫一遠端物件方法時，這項要求就會被封裝成 SOAP 呼叫傳遞給 Web Service，

Web Service 收到後便會執行其方法並將結果回傳。

SOAP 是利用 XML 的彈性和延伸性來封裝和交換 RPC 的呼叫，以 HTTP 為通訊協定連結的情況下，RPC 呼叫會自然對應 HTTP 請求，且 RPC 回應也會對應 HTTP 的回應 [11]，且所有 SOAP 通訊都使用 80 埠，代表在 Internet 上 SOAP 可以透過任何的 WEB 伺服器溝通，而不會受制於防火牆。此外 SOAP 也可以透過 SSL、S/MIME 等機制加密，提高傳輸時的安全性。

現今有許多的分散式物件協定，但沒有一個可以不被改變就能應用在 Internet 上，SOAP 提供一個架在 HTTP 上，簡單又有彈性的機制傳送需求和回應，讓遠端函數不須有任何改變，程式透過 Internet 存取已不再是一件大事。

3. WSDL(Web Service Description Language)

WSDL 是一種以 XML 格式來描述 Web Service 的語言，也可以說是 Web Service 的 IDL(Interface Description Language)，當服務提供者對外公佈 Web Service 訊息時，必須使用 WSDL 描述提供服務的操作及細節，包括型別(type)、訊息(message)、操作(operation)、埠口類別、連結、埠口及服務等。

WSDL 說明 Web Service 組件中可以被調用的方法名稱、輸入輸出的格式及端點的 URL 位址，讓呼叫它的程式可以依此內容建立呼叫的程式介面。

4. UDDI(Universal Description, Discovery, Integration)

UDDI 是一個服務註冊的架構，透過 UDDI 註冊器能把自己提供的 Web Service 資訊公開發佈，已註冊的企業也能夠被潛在的交易市場或採購商所搜索到。

UDDI 主要將 Web Service 註冊的資訊型態分為三類：

- 白頁(White Pages):紀錄 Web Service 的提供者、網址及連絡方式等資料。
- 黃頁(Yellow Pages):以分類的方式登錄 Web Service。
- 綠頁(Green Pages):登錄 Web Service 的技術細節。

由於應用服務可以透過網路取得，所以企業合作就不會那麼麻煩，也為全球電子商務的快速發展帶來難以估計的價值。

綜觀來說，Web Services 的優點主要如下[44][45]：

1. 可容易穿越防火牆：

SOAP 如同 DCOM 或其他分散式物件通訊協定一樣，能讓讓使用者端與伺服端的 RPCs 溝通。但與其他類似協定不一樣的地方是，它支援防火牆的使用；此外也可以被任何的物件使用，所以它將是兩大物件陣營 COM 和 CORBA 最好的溝通橋樑，讓彼此的物件程式可以跨平台透過網際網路呼叫，因為幾乎所有的防火牆都允許透過埠號 80(HTTP 協定的埠號是 80)來溝通，SOAP 這個分散式物件通訊協定是架在 HTTP 協定之上的，因此，即可穿越防火牆而不受一般物件導向防火牆問題的限制。

2. 非緊密連結：

SOAP 使用 HTTP 來做為它基本的傳輸協定，所以 SOAP 本質上來說算是無狀態的協定；亦即，它是一個鬆耦合(looselycouple) 的通訊協定，除了較有彈性外，也意味著使用者可以隨時更改任何一端的連結，但應用程式仍可以繼續運作。

3. 易於進行跨異質平台的整合：

Web Services 轉換成以信息為基礎(message-based)，高效能的同步技術，及使用 HTTP 或 SMTP 的 Web 通訊協定；重要的是，Web Services 使用 XML 以達到全球通用，XML 可說是 Internet 的通用語言，藉助 SOAP 與 UDDI 等 XML 架構的技術，建立可使用 XML 來提供 Web 形式服務的全新型態軟體，這些 XML Web Service 和軟體元件非常類似，均可程式化並重複使用，不同的是使用者可在 Internet 的任何地方存取他們，並自每個網站擷取資訊與服務，然後以自訂的形式進行合併，並可傳送給任何裝置。由於 XML Web Service 打破了 Internet、應用程式與各類運算裝置的區隔，促進企業間通力合作，提供空前的高度整合與可自訂的解決方案，讓客戶們在任何時間、任何地點與任何裝置上均能處理資訊。

貳、Web Services運用於企業的關鍵價值

由於企業導入新興科技主要是為了對本身所面臨的問題與難題提出解決方案，主要目的不外乎是希望達成新市場開拓、降低成本、提高生產力與增進客戶滿意度等等目標。Web Services 正可解決企業所面臨的難題，也因而成為企業達成目標的利器。

Web Services 為企業帶來的關鍵價值主要如下[54][38]：

1. 降低軟體開發成本

Web Services 釐定了一個嶄新的電腦軟體製造方法，電腦網路軟體程式碼的模組化是 Web Services 架構中最重要的一環。因為採用 Web Services 的模組化技術和工具，所有的軟體元件有如積木般，藉由完整介面定義所有元件可以輕易的組裝和重複使用，隨插即用(plug and play)；即使在龐大的複雜的系統也能由許多元件組成，程式設計時間大幅縮短，軟體開發成本自然降低。

2. 企業應用程式的整合

目前企業應用程式整合市場相當龐大，而每個專案動則花費數百甚至數千萬的預算。目前市場上整合的解決方案很多，但大多是一對一的整合。Web services 則提供另一種全然不同的方法，利用實作標準介面的方式，進行多點的整合。只要將現有的傳統系統可以包裝成 Web Service，便可與企業內的其他系統進行整合與使用。而由於採用全球通用的網路標準，Web Services 可以大幅降低和企業在外面的客戶、廠商及合作夥伴等協同合作的資訊系統結合費用。

Web Services 的平台中立性(Platform independence)，對於應用程式之間溝通的標準，無論定位為服務提供者或服務使用者，都不需要被鎖定於技術平台的使用上。因此，Web Services 可以被應用或開發的組合包含各種硬體、作業系統；開發程式語言、應用程式執行環境，任何連網的裝置皆可以受惠於 Web Services 的系統架構。

所以採用 Web Services 的各項技術，可以使跨平台、跨電腦語言、跨廠牌、跨越企業內的 Intranet 及企業間的 Internet，以及跨新舊系統都成為簡單、廉價的整合工作。

3. B2B 商業模式的開發與整合

一份由 InfoWorld 針對 CTO 所作的調查統計，70 % 的 CTO 認為 Web services 對企業最有影響的是在 B2B 電子商務上。

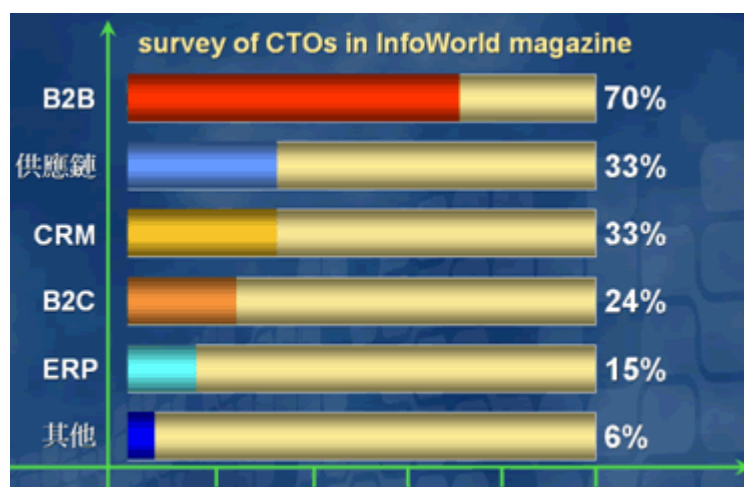


圖 4. Web Services 對企業影響項目比例圖

(資料來源：李清培、「.NET 研究室專欄：Web services 之規劃策略與設計模式－企業觀點」，2003)

由於網際網路的發明，將企業帶進「全球化」的時代，企業間的競爭愈加激烈，傳統的營運模式已無法應付這樣多變的市場環境，在這樣環境下的所要求的資訊系統，不僅僅是企業內部的整合（即 EAI 所要完成的事），更要能夠達到企業間的整合，以便達到企業延伸的目的。

除了製造業以外，傳統各產業之間其實都存著一套共通的電子交換格式 EDI 以進行 B2B 的資料交換，這些交換格式都並須建立產業間特殊的交換平台與資料格式或規範，而 Web services 不需要特定平台，Internet 本身就是一個很好的平台，只需依 XML 的規範，定義各產業的 Schema，即可在共通的標準上進行交易，

利用 Web services 建置的好處在於他的建置成本低，參與的廠商利用 Web services 的標準規範建置溝通介面，只需按照各產業所訂的標準定義描述資料的名詞及內容格式即可。因此，以 Web services 作為 B2B 溝通介面的規劃策略，具有相當大的彈性與經濟效益。

4. 商業流程管理[38]

商業的流程可以定義為很多功能的組合，採用 Web Services 對於企業各運作功能的優點有：

- (1). 功能的選擇有更高的自由度，企業的各功能可以用鬆耦合元件的方式協同作業，藉由清楚定義各元件與系統間溝通介面，讓企業可以在選擇元件部份做更積極的創新，也能以委外的形式來執行部分的功能。因此企業功能的採用可以更為切合需要，也會有更加的成本效益。
- (2). 新舊系統的整合更為方便容易；Web Services 能將既有系統(legacy system)經由轉換而與新系統順利橋接，由於相容性高，可以讓企業過去對於資訊科技的投資順利轉換到新的系統上，大大減低了企業採用 Web Services 的系統轉換成本與風險。
- (3). 各種功能可以動態組合，相當符合企業對於各種特殊服務或專案的需要，因此對於組織的應變彈性亦有正面的幫助。

由此可見，若能利用 Web Services 加上流程管理軟體，可以讓企業內部或外部合作夥伴間建立緊密之作業流程，還可依需要不斷加入應用程式模組，而不影響其他的應用程式。

5. 新商業模式的啟動

當企業因為自身的需求而開發一項 Web Services，亦可以藉由 Web Services 的公開交易機制將這項服務以付費方式、合作或供應商聯盟關係而提供予其他的使用者，變成服務提供者的角色，因此，Web Services 的公開市場，創造了一個元件重複利用的市場機會；而企業也可以利用委外的方式，將一些不是企業核心的業

務，透過網路，由最有效率的專家來提供。因此，Web Services 的市場運作機制也為知識的擴散創造了一個更好的管道，不再只是以往 Web/Html 純粹文字與圖形的分享，而可以促成複雜知識系統的擴散與採用，也啟動了新的商業經營模式機會。

參、小結

綜觀以上所述，Web Services 對企業帶來的效益指日可待，根據 IDC 的預估 [55]，Web Services 將驅動整個軟體、服務、硬體相關產業的進步，並且將於 2004 年創造出 16 億美元的商機，於 2007 年更有機會創造出 340 億美元的商機。

儘管如此，目前許多廠商仍然對 Web Services 抱持觀望的態度，最主要原因應為缺乏商業流程管理的典範，企業在營運互動上一直缺乏對商業流程描述與執行的技術，使 Web Services 無法支援企業往來互動的狀態管理與維護。不過，針對這樣的情形，軟體廠商也紛紛提出一些服務流程語言；例如：IBM 的 WSFL 和 Microsoft 的 XLANG 等，而在 2002 年產生的 BPEL4WS 更是結合上述兩者，為網路服務互動上的商業流程所提出的技術方案，將於下節中作一討論。

第四節、商業流程執行語言 - BPEL4WS

壹、BPEL4WS 簡介

Web Services 商業流程執行語言 (Business Process Execution Language for Web Services, BPEL4WS)是在 2002 年 8 月由 BEA、IBM 與 Microsoft 所共同發表的，定位為整合方面的 Web Services。BPEL4WS 能完成 Web Services 服務的調用、操縱資料、拋出故障或終止一個流程等工作的不同活動，藉由編排或產生這些活動，將其串接起來，進而創建出更複雜的流程。

BPEL4WS 主要結合了 WSFL 支援圖形化流程及 XLANG 支援流程結構化構造等特

性，並屏除一些複雜繁瑣的部分，形成一種較為自然的高階商業活動語言。

貳、BPEL4WS 的特點[35]

1. BPEL4WS 並非全新概念

就 BPEL4WS 語言本身的語法結構和編輯概念，對於大多數設計師而言其實是很熟悉的，以下就幾點加以說明：

(1). 與 XML 關聯

BPEL4WS 的語法原完全基於 XML 的規範，若不考慮它的程序語言特性，使用者可以把它理解為普通的 XML 文檔規範；亦即可以把 BPEL4WS 中的所有節點對應到一個虛擬的 DTD 文件中。若不考慮其程序特性，使用者在編寫 BPEL 文件時只是根據這個 DTD 所定義的規範在編寫普通的 XML 文件罷了。

BPEL4WS 內容語法定義主要基於以下幾個 XML 規範：

- WSDL 和 XML Schema 類型：定義提供了 BPEL4WS 流程所用的資料模型。
- XPath：為資料處理提供支援。
- 所有外部資源和流程參與服務元件(partner)以 WSDL 進行描述。

(2). 關於繼承方面：

BPEL4WS 與一般程式設計語言基本特性相同；例如：

- 賦值操作(< container >)；
- 迴圈(< While >)；
- 條件判斷選擇(< switch > 和 < case >)；
- 過程調用(< invoke >)；
- 錯誤捕捉(< catchfault > 和 < catchall >)；
- 錯誤拋出(< throw >)；
- Java, C# 語言中的 try 操作(< scope >)等等。

(3). BPEL4WS 結合了商業處理特點

- 提供對於遠端調用(`< invoke >`)的同步和非同步處理。
- 提供平行(parallel)操作(`< flow >`) :
透過 BPEL4WS 可以同時調用不同地方的 Web Services 進行處理(如：下單、訂貨等)。
- 提供補償操作(`< compensate >`) :
其實任何程序執行都可能有錯誤產生，有些操作錯誤並不會造成影響，但有些錯誤結果則必須進行一些錯誤補償的操作。因此，雖然 BPEL4WS 是最近提出的新規範，不過在本質上卻也讓使用者覺得十分熟悉，使用門檻也降低許多。

2. BPEL4WS 流程實現 Web Services 服務

BPEL4WS 可用來描述兩種不同的情形的：

- (1). 可執行的商業流程(executable business processes)：

塑模參與者在商業互動中的實際行為。

- (2). 不可執行的抽象流程(abstract process)：

抽象流程亦即商業協定(business protocol)，實作上只使用到 WSDL 的 portTypes；換句話說，BPEL4WS 藉由複合一組現存的服務定義出一個新的 Web Services，而服務的所有 portTypes 則組合成流程的邏輯。

而 BPEL4WS 流程實現 Web Services 服務的概念則如下圖 5 所示：

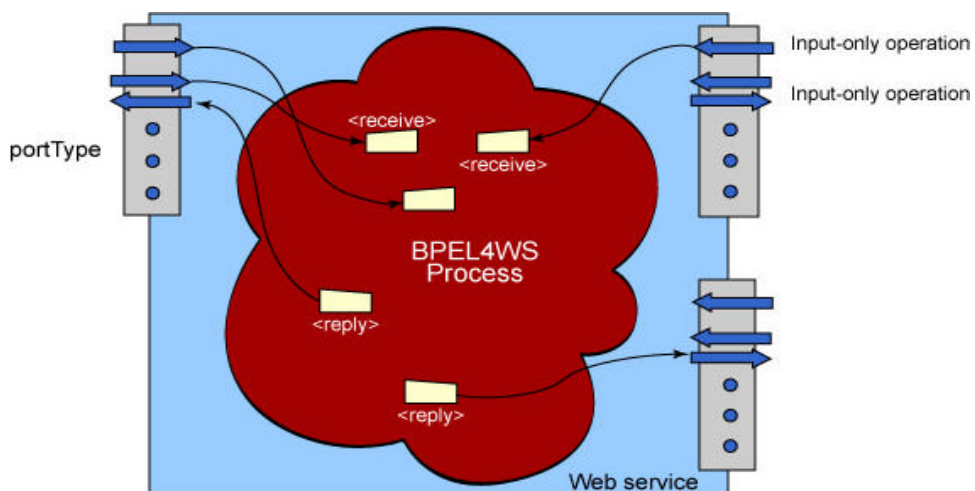


圖 5. Web Services 服務概念圖

(資料來源：IBM BPEL4WS document)

BPEL4WS 流程本身就是一個類似用來表達算法的流程圖，流程中每一步驟稱為一個活動(Activity)。Web Services 的 Port Type 定義就像是服務的入口點一樣，這些入口點藉由消息(message)傳遞接收等動作，進行 BPEL4WS 流程的佈署與執行動作。

3. BPEL4WS 並不可執行

BPEL4WS 雖然定義為一商業流程執行語言，但實際上它並不執行商業流程中的任何細節；亦即它並不涉及商業數據的儲存與處理。BPEL4WS 本質上來說應該為一描述性語言，它只描述什麼時候以什麼順序到哪邊調用哪些需要的服務，並且要怎麼組合這些調用的服務。唯一涉及數據操作的地方只有 container 的使用，不過 container 中的資料只是一些臨時的資料，一旦商業流程處理流程結束，這些資料也跟著消失。因此，所有的操作都由 BPEL4WS 來完成，使用者只會看到本身的用戶介面，而不需擔心這些功能是怎麼實現的。另外，若要實現 BPEL4WS 規範，則必須透過其他引擎來執行運作，例如：IBM 的 BPWS4J 或是 Collaxa BEEL Server 等。

4. BPEL4WS 是真正的分散式系統：

和傳統的分散式系統相較下，利用 BPEL4WS 實現分散式系統具有更高的靈活性，由以下特點可看出：

- (1). 各節點可以為異質性系統。
- (2). 運行時可動態選擇要進行處理的結點機。
- (3). 只要符合 SOAP 協定，可採用各種通信協定進行溝通。

參、BPEL4WS 的結構與使用方法[35]

總體上來說，BPEL4WS 按照主要功能可以畫分為四個最重要的部分：

1. 資料處理

商業流程涉及的狀態由接收和發送消息以及其他有關資料組成，業務流程的狀態維護需要使用狀態變數，稱為 container。另外，資料運算式則將狀態中的資料抽取並作有意義的組合，以控制流程行為。最後狀態在更新需要賦值概念。

2. 基本活動

BPEL4WS 只使用和支援 WSDL 的 input-only 和 input-output (請求-回應 (request-response)) 操作；output-only 和 output-input (要求-回應 (solicit-response)) 操作既不需要也不支援。

此外，BPEL4WS 本身即為一流程圖，類似用來表達演算法的流程圖。流程每一步驟稱為一個活動；主要有以下基本活動：receive、invoke、reply、wait、assign、throw、terminate、empty 等。

3. 結構化活動

藉由使用語言所提供的結構化活動，可以定義一組有步驟的序列 (<sequence>)、使用現在常見的 "case-statement" 來產生分支 (<switch>)、定義迴圈、執行選擇路徑，也可指明執行順序方面的約束。

4. 服務互動

BPEL4WS 把與流程互動的其他服務稱為 "夥伴(partner)"，每個夥伴都以 Web Services 的型態呈現，在互動過程中扮演不同的 "角色(role)"。

服務間兩兩關係被塑造成 "夥伴鏈結(partner link)"，定義關係中每個服務所扮演的角色和指定給個角色所提供的 portType；每個服務則可包含多個 WSDL portType。此外，由於 WSDL 互動模式屬於無狀態式(Stateless)，因此，BPEL4WS 以變數來存取與維持流程運作的狀態資訊。

肆、BPEL4WS 應用現況與發展 [32]

BPEL4WS 已有許多科技廠商如 BEA、IBM、Microsoft、SAP 等表態支持，OASIS

也為 BPEL4WS 成立一技術委員會，公開發表的版本為 Business Process Execution Language Services Version 1.1。預估 2004 年前後，商業流程塑模語言將彙至 BPEL4WS 規格中，並出現可行做法，用來管理商業流程的定義、執行、監督與交流；2005 至 2006 年，主要應用軟體廠商將採用 BPEL4WS 架構重新規劃產品；2007 年左右，應用軟體廠商將可從不同應用程式輸出/輸入商業流程模型。

第五節、統一塑模語言 - UML

壹、UML 簡介

統一塑模語言(Unified Modeling Language,UML)是為了有效的簡化和強化大多數物件導向開發方法所顯露的問題所發展出來的。在 1997 年 11 月被 OMG 全體成員一致通過並作成標準，OMG 並承擔進一步完善 UML 標準化的工作。

UML 主要目標為[12]：

1. 提供即時的視覺化表達模式語言(Modeling Language)，讓使用者可自行發展與交換模型。
2. 提供可擴充且規格化的機制擴充核心概念。
3. 與特定程式語言及發展程序無關。
4. 提供模式話語言正式基礎。
5. 激勵物件導向工具市場成長。
6. 支援如：協作(Collaboration)、結構(Framework)、樣板(Pattern)、元件(Component)等高階發展概念。
7. 已開發軟體整合的最好方法。

UML 提供許多種類的圖形表示方法，來協助使用者分析與設計系統的各個層面，以模型組呈符號的排列，展示系統的某特定部份運作和功能；所包含的圖形包括：使用者案例圖(Use Case Diagram)、循序圖(Sequence Diagram)、協作圖

(Collaboration Diagram)、類別圖(Class Diagram)、狀態圖(State Diagram)和活動圖(Activity Diagram)等等。在系統分析過程中，UML 可以完成的工作包括：視覺化(Visualizing)、詳述(Specifying)、構造(Construction)、文件化(Document)等，而這些工作正是系統開發過程中最重要的工作。因此，也可把 UML 用於複雜的軟體密集型系統。

貳、利用 UML 塑模 BPEL4WS 系統

塑模的目的是要將所設計的系統結構和系統行為作一溝通，且對系統架構進行視覺化和控制。市場上很多專案成功的原因之一就是塑模方法的採用。對 BPEL4WS 系統進行塑模主要原因是為了要更加了解整個系統架構，且塑模語言的採用還能簡化系統並提供元件重複使用的功能。

BPEL4WS 系統模型是對現實商業流程的簡化，提供整個系統運作的藍圖，其中包括了詳細的計畫和高抽象層次考慮系統的總體計畫。雖然 BPEL4WS 使用起來並不複雜，也無複雜資料結構處理，但面對瞬息萬變的商業流程及其商業行為，以及分布在 Internet 上的大量 Web Services 和其介面，對 BPEL4WS 塑模更能幫助使用者理解正在開發的系統。

綜觀來說，對 BPEL4WS 系統塑模主要目的如下[49]：

1. 建造出來的模型(Model)能幫助使用者依照實際情況及所需細化程度，對系統進行視覺化。
2. 建構的模型能詳細說明整個 BPEL4WS 系統的靜態結構和動態行為。
3. 建構的模型要能指引使用者建構 BPEL4WS 系統的模板。
4. 建構的模型要對使用者在開發過程中所作的決策進行文件化動作。

由於 BPEL4WS 本質上是基於物件導向技術和元件技術，利用 UML 可以對要建構的系統進行視覺化、詳細說明、構造和文件化；這也正是對 BPEL4WS 系統塑模的最主要目的。

在對一複雜系統進行塑模時，可利用 5 個相互關聯的觀點(View)來描述整個系統架構，如下圖 6 所示：

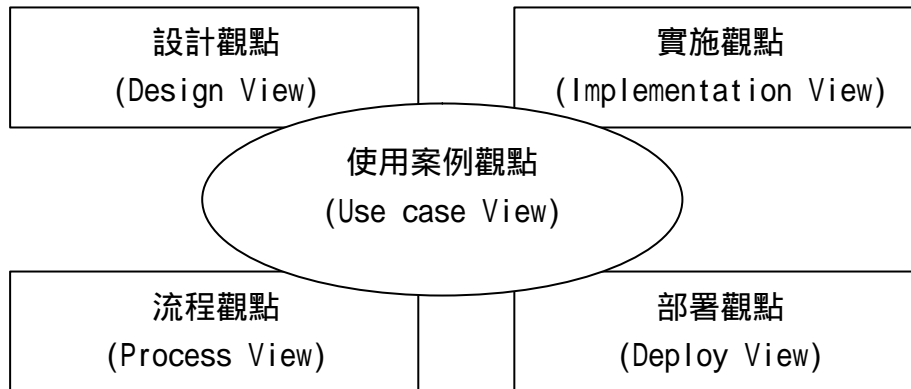


圖 6. 系統架構五個相關觀點

資料來源：廖漢君、「UML：類別圖轉換至物件
關聯模式之研究」，91 年

- 使用案例觀點(Use case View)：

由描述系統行為的使用個案組成，這些系統行為是由使用者、分析師、測試者的觀點來描述，並不實際描述軟體系統的組織。

- 設計觀點(Design View)：

包含類別(class)、介面(interface)和協作(collaboration)。主要來自於描述問題及其解決方法中之辭彙描述。這個觀點主要支援系統的功能需求，表達系統應提供給使用者之服務。

- 流程觀點(Process View)：

包含形成系統並發與同步機制的線程和進程。主這些是來自於系統的平行與同步機制，這個觀點主要表達系統之績效、產出與可擴充性。

- 實現觀點(Implementation View)：

由可以不同方式組裝實際可運作系統之獨立的元件與檔案所組成，這個觀點主要表達系統版本的結構配置管理。

- 部署觀點(Deploy View)：

由構成系統之硬體類型的節點（Nodes）所組成，這個觀點主要表達組成實際系統之零件的分配、傳遞訊息與安裝。

下表2為UML系統塑模觀點使用圖形及適用對象表：

表2. UML系統塑模觀點使用圖形及適用對象表

系統觀點	使用圖示		適用對象
使用個案觀點	使用個案圖、循序圖、合作圖、狀態圖、活動圖		使用者、設計者、測試者
設計觀點	靜態結構	類別圖、物件圖	設計者
	動態行為	互動圖（循序圖、合作圖）、狀態圖、活動圖	設計者
流程觀點	狀態圖、循序圖、合作圖、活動圖		設計者、系統開發者
實施觀點	元件圖、狀態圖、循序圖、合作圖、活動圖		系統開發者、系統整合者
部署觀點	配置圖、狀態圖、循序圖、合作圖、活動圖		系統工程師、系統整合者

(資料來源：資料來源：廖漢君、「UML：類別圖轉換至物件關聯模式之研究」, 91年)

以上五種觀點可單獨使用也可交互作用，在 BPEL4WS 系統中，我們則將整體架構利用不同視圖進行塑模。

第三章 動態商業流程協調整合架構

本章將介紹本研究所提之動態商業流程整合架構，第一節介紹系統發展步驟，接著第二節進行系統需求分析部分，並於第三節進行架構設計；第四節則進一步以 UML 塑模系統架構，以便進行系統建構。

第一節、 系統架構發展步驟

本研究利用物件導向的方式分析與設計開發的系統架構，主要步驟如下所示：

1. 系統需求分析

本研究依據第二章所彙整出在協同作業環境下，商業流程管理所面臨的整合問題為需求分析的主要依據，利用 Web Services 技術對於應用系統間的整合優點，進行商業流程協調整合架構的分析與設計。

2. 動態商業流程整合架構設計

本研究所提之動態商業流程整合架構其動態特性主要定義為流程執行時之即時動態性；亦即當企業合作夥伴間所提供之流程參與服務元件，透過本研究架構進行流程串接部署後，即能透過流程引擎作動態銜接執行，處理非同步性和長時間運行之商業交易，而管理者只要透過操作介面便能做即時性的監控和例外處理。在此我們由大架構到系統架構內部元件逐步進行分析，以便進行系統塑模。

(1). 電子化商業流程運作環境架構

依據前述需求分析，進行整合架構設計，本研究主要以服務導向架構為基礎，設計出以本研究為商業流程協調整合之 Web Services 電子化商業流程運作環境架構，說明電子化企業間如何利用 Web Services 之服務導向架構進行商業互動

(2). 流程整合架構內部元件定義與互動

藉由前述架構之描繪，進而設計出本流程整合架構之雛型和內部元件的定義及元件互動關係。

(3). 流程建構最基本之內部元件細部分析

此步驟則將流程運轉最基本之內部元件，進行細部分析，以便說明這些元件如何在交易中進行流程的串接和溝通。

3. 以 UML 塑模系統架構

在此步驟，我們採用 UML 對所設計之 BPEL4WS 系統架構進行塑模，以標準的圖形和語言描述研究之系統架構，方便讀者理解並作為後續研究發展時的陳述與溝通。針對上一步驟的系統架構進行功能的規劃與設計，將這些系統外部所能見之功能進行使用案例規劃，以外部功能表現系統內部類別應有的整體輪廓，再將產生的使用案例進行內部功能的元件分析，找出系統中可能的類別之後，在繪製出類別互動循序圖，透過循序式的方法對系統進行視覺化、詳述、結構性和文件化等動作。

4. 系統展示與評估

最後我們將所設計之流程整合架構以一簡單訂單確認查詢示例進行系統實作，以驗證其可行性，並進行評估。

第二節、系統需求分析

根據第二章文獻探及核心技術探討，本研究匯整出在協同商務環境中，企業合作夥伴間為了共同目標，希望能將一連串的營運流程都能透過 IT 加以整合、同步作業，因此協同作業下商業流程管理的重要性也亦趨重要。

基於第一章研究動機中跨企業流程整合時會面臨的其他幾個問題點，本研究針對其問題點為考量進行需求分析與設計並尋求問題解決之方式：

1. 異質性系統間應用程式溝通問題：

在流程整合過程中，往往需要連結至不同企業夥伴間營運系統，不過在企業環境中往往因為不同產業特性或是安全性等等考量，各自採用不同的作業環境或

系統平台，因此在資料傳輸及應用程式間的溝通都是相當複雜與困難的問題。

本研究所採用商業流程執行語言是以 Web Services 技術為基礎之 BPEL4WS(Business Process Execution Language For Web Services,BPEL4WS) , 在 Web Services 技術中最基礎的呼叫協定為 SOAP , SOAP 是利用 XML 的彈性和延伸性來封裝和交換 RPC 的呼叫,並以 HTTP 為通訊協定連結,具有可擴充性跨平台與穿越防火牆等等優點,因此,利用 SOAP 呼叫夥伴間以 WSDL 所封裝描述之 Web Services 交易服務模組,將能解決此類異質性系統間溝通的問題。

2. 協同作業下商業流程高度變化性：

在高度競爭的商業環境中，商業模式(Business model)也會因雙方的夥伴關係而隨時調整，如何快速且即時的調整企業本身營運流程或是配合合作夥伴作一更改，都是企業急需面對的問題。

針對此，本研究利用 Web Services 技術能重新組合的特點，讓資訊系統得以機動性配合任務的調整，對各項以 Web services 方式提供的服務進行不同形式的連結和協同運作，同時也能快速的加以部署。

3. 產業標準及解決方案問題：

企業間流程整合部分，市場提供解決方案過多且複雜，目前已有幾個國際標準組織進行相關流程標準的訂定，唯有透過 public process 的訂定才能將企業間的共同行為標準化，目前流程導向式的整合架構有 Rosetta NET、ebXML 等標準，雖然提供開放、產業通用的流程標準，不過這些標準所著重的產業與解決 B2B 電子商務資料交換問題所採用的技術及範疇，彼此間有相當的差異，也屬於較制式化的互動關係，導致企業選擇的困惑。

Web Services 技術則是架構在如：XML、SOAP、WSDL 等標準協定上，因此若能藉由 Web Services 技術標準化的特性，提供一流程整合架構，透過標準訊息的傳遞與簡單的單一操作介面，讓企業間交易服務模組能在此架構下進行連結與協調整合，也能提供企業作一選擇。

4. 整合成本問題：

Web Services 能以低廉代價建構所需的應用程式模組，且不因採用不同軟體廠商的整合軟體而額外付出 Adapter 的軟體授權費，也能降低企業經營的成本問題。

綜上所述，本研究的目的是建構一個以 Web Services 技術為基礎的商業流程協調整合機制，將企業間所提供的服務封裝成 Web Services 模組，運用 Web Services 技術跨平台及訊息標準化的特性，及 BPEL4WS 流程描述語言支援圖形化流程及流程結構化等性質，將一連串互動交涉服務元件，依特定商業目標和商業協同文件，在此架構下進行整合與任務協調，主要是應用在以網路服務為媒介的商業環境中。

第三節、 動態商業流程整合架構設計

企業間商業流程整合面臨的問題即企業互動環境中，充斥許多異質系統，不但溝通困難，要達到整合與協同的目的更是一大挑戰。本研究目的是利用 Web Services 技術，並根據前一節對商業流程整合系統功能需求及架構討論結果，建構一動態商業流程協調整合仲介架構，應用在以 Internet 為溝通媒介之企業間整合(B2B)環境中。

壹、 電子化商業流程運作環境架構

下圖 7 為以 Web Services 電子化企業商業流程整合運作圖，並於後作一說明：

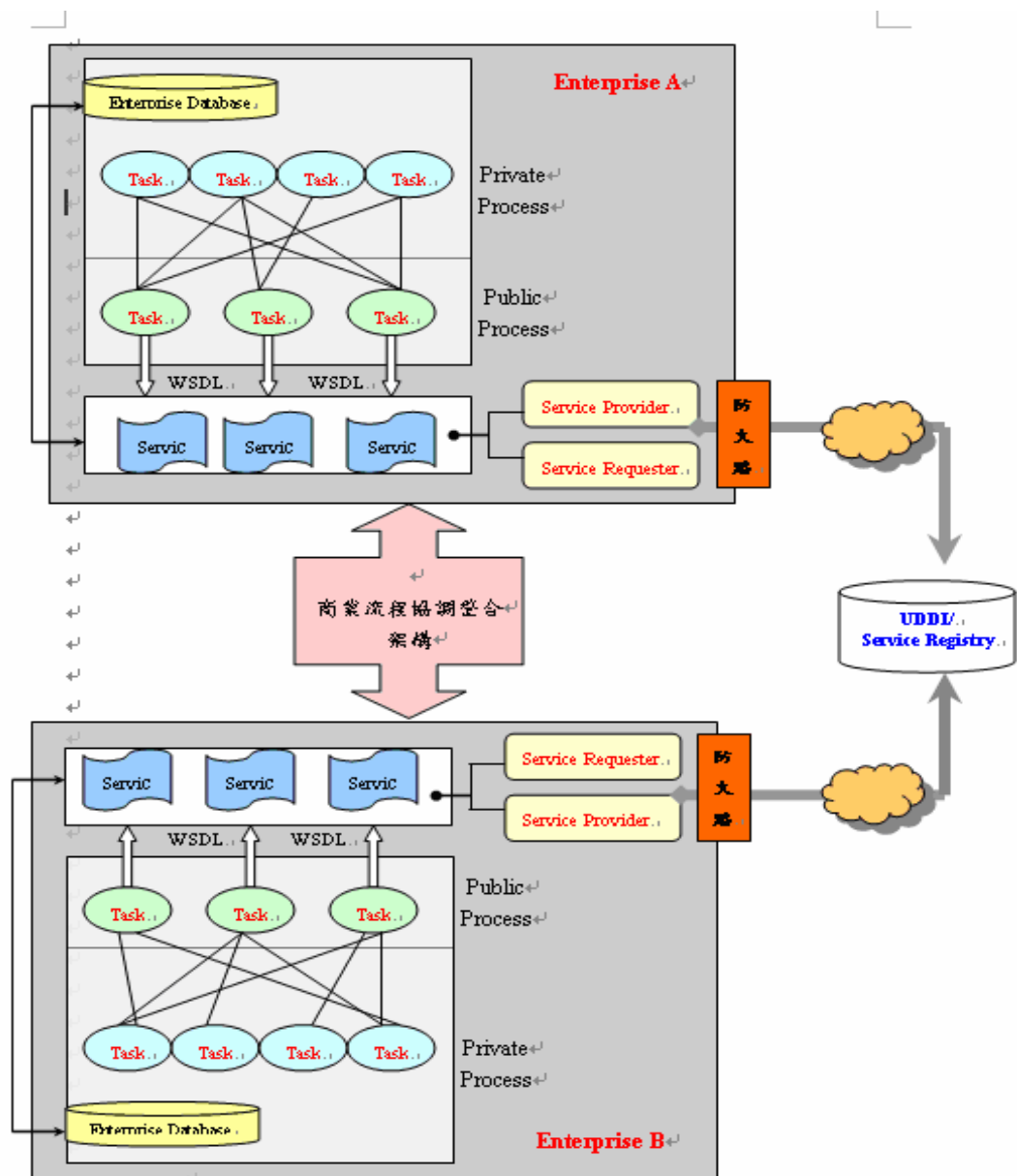


圖 7. 以 Web Services 為基礎之跨企業流程整合運作圖

(資料來源: 本研究)

本研究之跨企業流程整合運作概念主要以服務導向架構模式為主, 再透過商業流程協調整合架構, 作企業間商業交易的協同合作平台, 達成共同之商業目標。

1. 本研究將商業流程分為內部流程 (Private Processes) 和外部流程 (Public Processes) ; 內部流程為其企業內部各部門之運作, 而外部流程則為和企業合作夥伴間的互動流程。由於企業大多不願意商業夥伴知道其內部所有的決策和資料

管理，此外透過把內部流程和外部流程分開，當要改變內部流程時並不會影響到外部和合作夥伴間的商業協議。

2. 流程中基本單位為 task。task 主要為企業中個人或一個小組在一時程裡執行一項任務的具體結果，每一個 task 都會有其前後觸發事件，如：前置作業或是後續動作等等，並會產生執行結果。在本研究中，我們將公開流程的 Task 依照其商業功能以 WSDL 封裝為不同的 Web Services 功能元件模組，在本研究流程整合架構中我們將其稱為 partner 元件。跨企業商業流程整合可視為 partner 元件間的交互運作關係，亦即整合之際必須讓不同企業夥伴間所屬的 partner 元件模組能夠彼此互動，依照彼此間的企業協議及協同文件等觸發後續動作並串接成流程，互傳訊息及資料，達成共同的商業目標。
3. 企業可同時扮演 Services Provider 和 Services Requester 的腳色，並由公開 Services 註冊機制(UDDI)中，尋找適合之企業合作夥伴，建立協同作業協動，並開始執行協同作業。
4. 本研究所提之商業流程協調整合架構則讓企業間能在此架構平台下進行商業流程之整合。

為了能將企業間所提供的 Services 元件動態地串接起來，本研究擬定以 Web Services 的業務流程執行語言(Business Process Execution Language For Web Services, BPEL4WS)，作為指定業務流程及相關 Services 元件的調用、操縱資料、拋出故障或終止一個流程等工作的不同活動。

主要系統架構示意圖如下圖 8 所示：

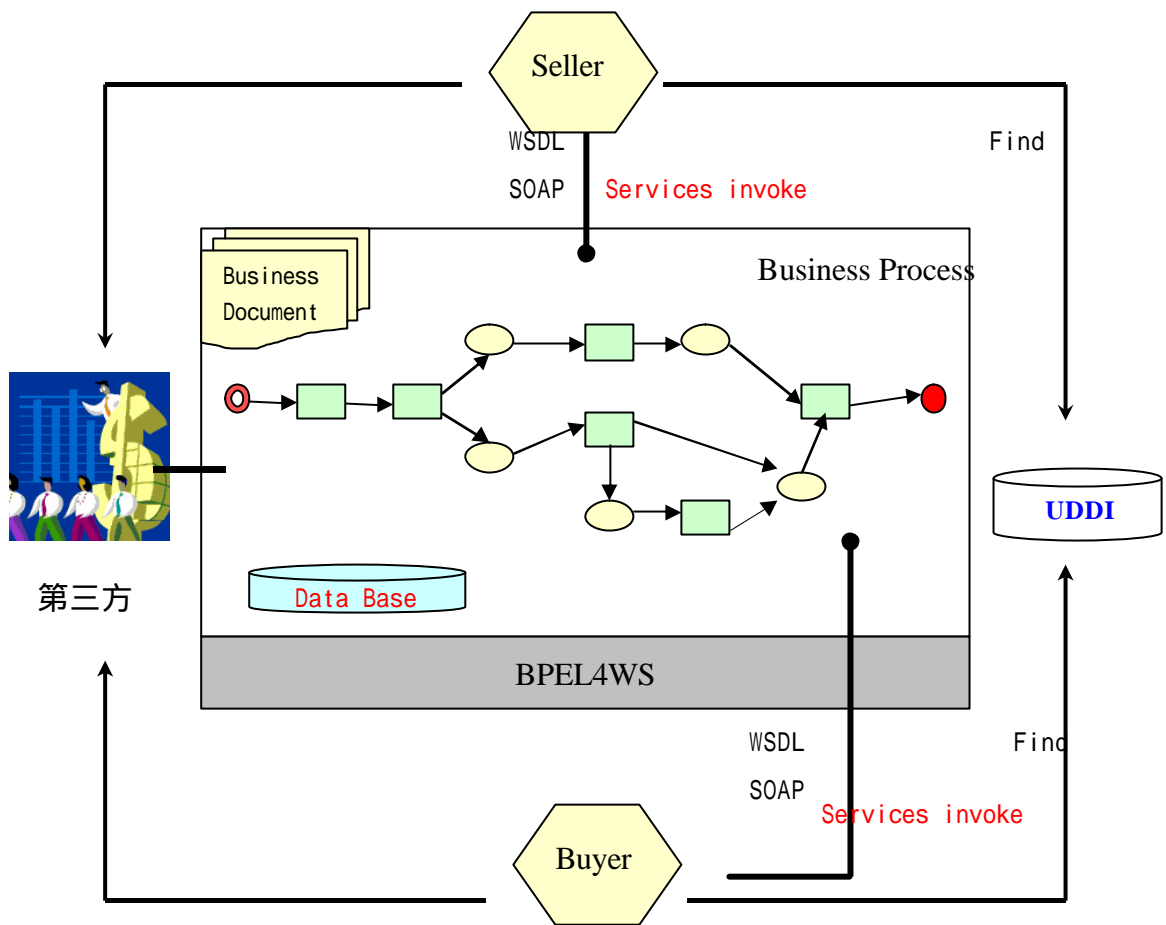


圖 8. Web Services 為基礎之商業流程協調整合架構示意圖

(資料來源:本研究)

本研究系統架構中，為使讀者更明瞭，將企業間合作關係簡化為買賣雙方和第三方單位。買方提出商業交易服務請求；賣方則為企業合作夥伴或協力廠商等，提供買方相關交易需求；第三方則為銀行信貸服務等單位，負責付款機制、信用核對及借貸等等動作。

我們依據買賣雙方及第三方所提供之服務元件的介面定義，並參照買賣雙方所協議之商業協定及文件，例如文件中所提及之資料的相關行為，例外情況處理等。作為本研究之 BPEL4WS 流程觸發及調用部屬之參考，進行流程部署與調用，並開始執行商業交易之流程。

貳、流程整合架構內部元件定義與互動

在 Web Services 元件內外部互動及定義方面，則如圖 9 所示：

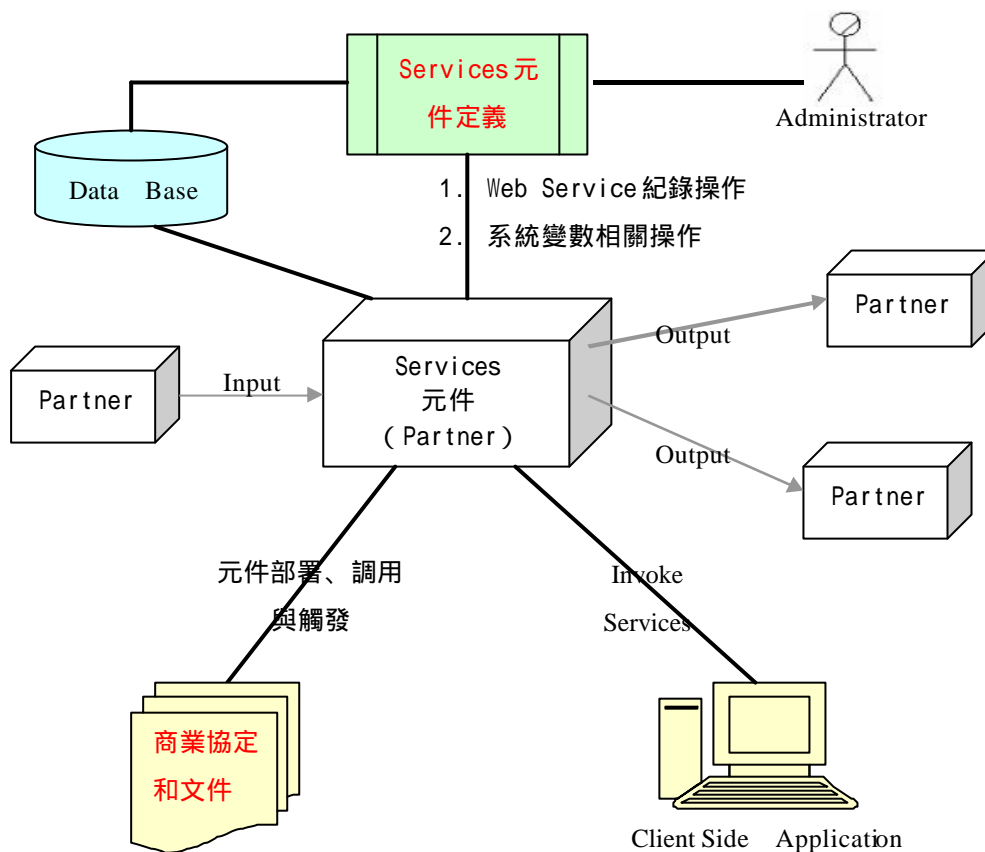


圖 9. 服務元件互動圖

(資料來源:本研究)

在 BPEL4WS 系統架構中，Services 元件的前後任務觸發元件都稱為 partner。而與 Services 元件互動的其他因素則有：

- Services 元件定義分析

由於企業提供服務時會將其服務利用 WSDL 作一完整定義，(1).功能性的描述(如：服務實作定義和服務介面定義)和(2).非功能性的描述等(如：服務類型、服務規格使用、訊息語法等等細節)，在此部分我們先針對其對於合作夥伴所提供之服務，進行服務元件分析動作，辨識出該服務擁有的屬性與行為等，並

進行儲存。

- 商業協議和文件(business protocol and document)

為協同合作企業間所制定之商業協議而產生之商業文件，主要包括：

- 資料相關行為；如：訂單中單項產品數量、訂單總價或交付截止日期等。
- 例外處理與補償；如：協力廠商若無法提供支援時的替代方案等異常條件及其處理的方法。
- 長期運作的商業交易所需的跨企業夥伴服務協調等。

商業流程運作時則根據此部分的協議和法則決定服務元件的前後觸發動作，主要為服務串接的依據。

- Client 端所提供的服務應用程式

流程運作時，根據服務提供者所提供之服務元件的描述，調用所需之服務應用程式加以執行。

- Data Base

儲存 Services 元件的內部定義及商業協議文件，方便取得流程運作時 Services 元件該處理的資料內容與處理的結果。

參、流程建構最基本之內部元件分析

在本研究系統架構中，商業流程處理的最基本元素為 partners 亦即 Services 元件，而 Services 元件則是透過 WSDL 將其作一描述和包裝，WSDL 所屬之資訊模型主要用來定義 Services 元件的服務介面和服務實作兩方面，資訊模型圖如下圖 10 所示：

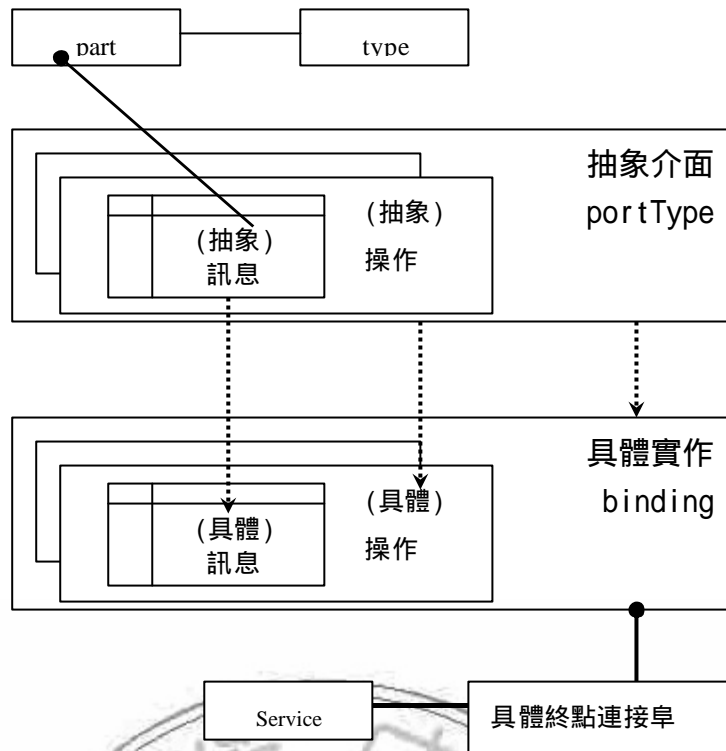


圖 10. WSDL 資訊模型圖

(資料來源：Steven Graham, Simeon Simeonov, Toufic Boubez, Doug Davis, Glen Daniels, Yuichi Nakamura, Ryo Neyama、以 Java 建置 Web Services、)

- PortType

PortType 元素主要用來描述 Web Services 介面，如：

```
<!--Port type definitions-->
<portType name= " PriceCheckPortType " >
  <operation name = " checkPrice " >
    <input message = " PriceCheckRequest " />
    <output message = " PriceCheckResponse " />
  </operation>
```

< /portType >

WSDL 文件可以包含零或多個 PortType 定義；一般而言，大多數的 WSDL 文件都會包含一個 PortType。這樣的定義將不同的 Web Services 介面定義分割成不同的文件，也提供很好的重覆使用方法。PortType 元素中的 operation 元素，則主要用來操作定義 Web Service 上的方法，包括方法的名稱和輸入參數及方法的輸出或傳回型別等。

- message

WSDL 文件中，也可包含零或多個 message 元件，每個 message 元件可以用來當成操作內的輸入訊息、輸出訊息或錯誤訊息。

- types

Types 元素主要是用來定義一個或多個結構描述，讓使用者能有彈性地來描述完全不同的型別系統。

- binding

Binding 元素主要是要告訴服務要求者，如何以通訊協定特定的方法格式化訊息。每個 portType 可以有一或多個 binding 元素關聯。對於一個 portType，binding 元素可以描述如何使用一個訊息/傳輸協定來叫用操作，例如：SOAP 是透過 HTTP 來傳送訊息等。

- port

Port 的唯一目的就是指定終點主持 Web Service 的網路位址，亦即，port 元素將一個通訊協定特定的位址與個別的 binding 元素做關聯性連結。

- service

Services 元素的目的是包含一組相關的 port 元素。

因此，流程服務元件模組圖 11 下：

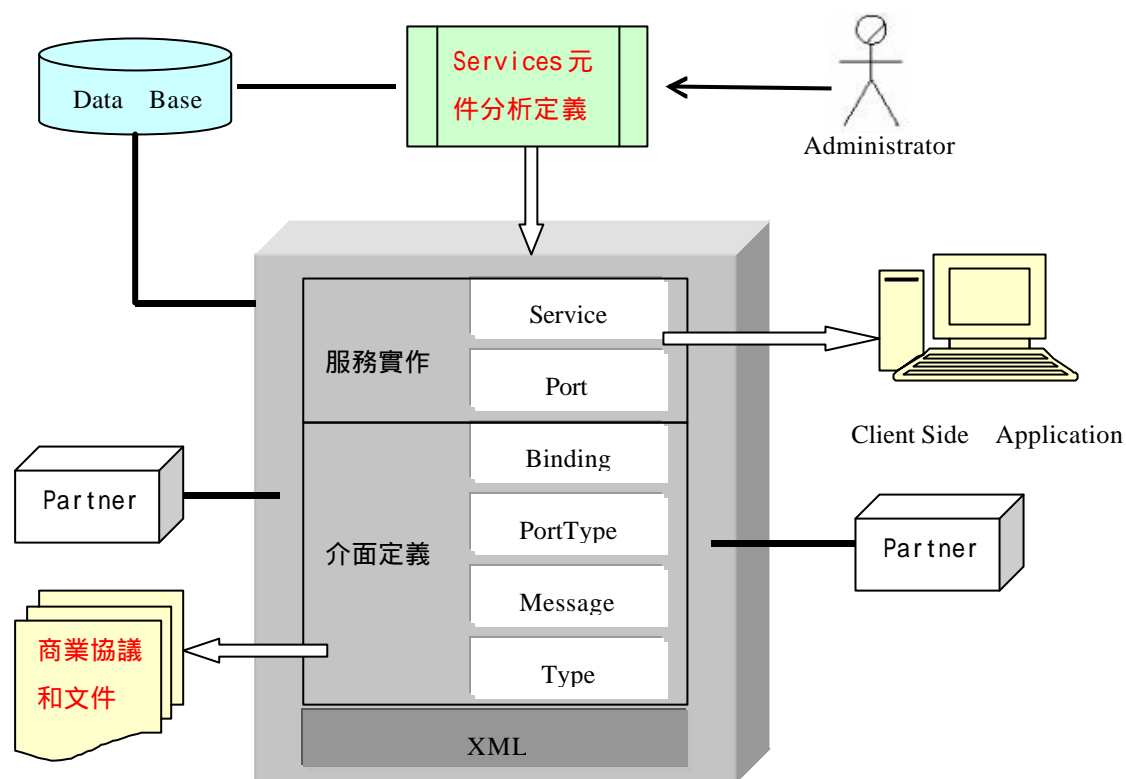


圖 11. 元件內部功能分析圖

(資料來源：本研究)

本研究架構主要使用 BPEL4WS 作為流程執語言，在此語言描述下，每個企業間的商業流程都透過 Web Services 介面與其他企業流程連結、作用。

介面定義中的 Type 定義用於 Web Service 裡所有資料型別的集合，由各種訊息部分元素來參考；message 主要是定義一組由方法簽章或是操作參考的參數；portType 為 Web Service 的抽象介面定義，每個子元素 operation 則定義一個抽象方法簽章；binding 包含在 portType 中，主要定義如何將特定資料格式與通訊協定的結合以具體方式呈現。

在服務元件實作部份，透過 port 指向終點服務的網路位址，並與介面定義中 binding 作一關聯連結，啟動合作伙伴所提供之 Services 元件。由於 Web Services

主要的基本技術就是 XML，換句話說，一個 Web Services 就是指一個應用程式能透過 XML 來描述、查詢及利用 URI 來辨識，並經由介面和連結方式的完整定義，藉著 XML 形式的訊息以合乎網際網路的協定來直接驅動，因此，所有的流程服務元件都是建構在 XML 之上。

肆、動態商業流程協調整合架構設計

根據前述對於流程架構和基本元件之分析設計，在本研究架構中，主要期望能藉由單一使用介面讓流程操作者能在此平台架構下進行流程整合動作。我們將流程整合動作以階段性方式呈現；開始流程運作時，我們先將流程參與元件置入本研究平台中，藉由元件分析動作，得知其實作及介面定義等元素，接著再置入此流程之參考商業協議文件，進行分析與儲存動作，並根據元件分析和協議文件進行流程佈署和串接，最後再將流程置入工作引擎中進行流程實際執行的動作。透過流程引擎，流程可以自動執行長時間非同步交易，管理者可以不須時時佇立在電腦螢幕前面，只需藉由視覺化之流程監控，便可得知目前執行狀況與及時之異常處理；此外，藉由錯誤補償程式之定義，當流程發生異常狀況時，也會自動回傳錯誤訊息並進行補償動作與結果回報。

本研究之動態商業流程協調整合架構圖則如下圖 12 所示：

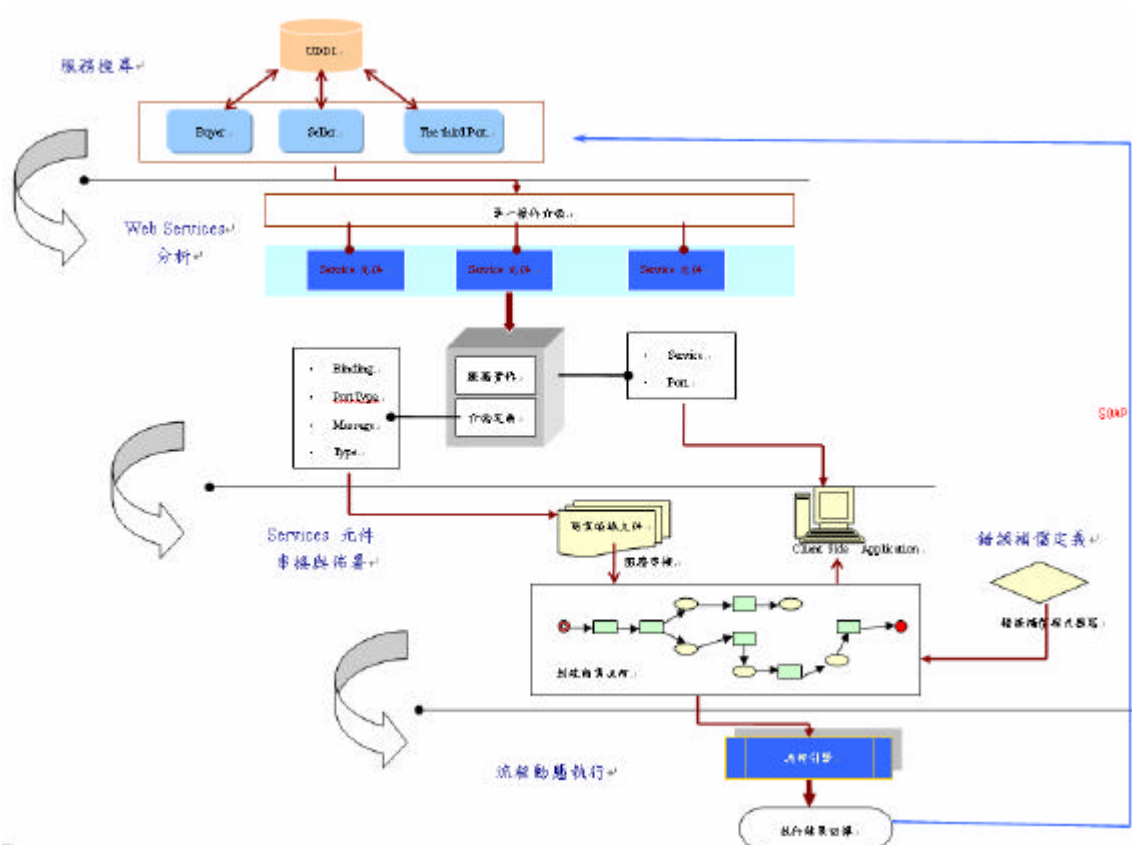


圖 12. 動態商業流程協調整合架構圖

第四節、小結

本章系統分析部份，針對第一章研究動機中所提及之商業流程進行整合時所會面臨到的問題點，和本研究所擬定使用的技術支援進行問題的解決和系統功能需求的分析，希望能針對每一問題點提供較好的解決方法，再進而針對前述之分析結果進行系統架構和系統功能設計。

在系統架構和系統功能設計部分，主要由整體大架構慢慢抽絲剝繭以推導到基本元件的內部定義，藉由上而下之功能分析，以便建構較完整之系統結構和釐清系統元件功能需求，方便進行下一章節的系統塑模。

第四章 以 UML 塑模系統架構

此章節中，主要對第三章所提出之流程整合架構進行塑模動作，以外部能見之功能進行使用案例規劃，以外部功能觀點表現系統內部類別應有之輪廓，並依不同事件流程繪製成使用案例圖供後續元件分析使用。經由上述步驟，我們針對每一使用案例進行類別發掘與建構，以其找出最適之類別模型，以利於後續系統建構。

第一節、系統架構使用案例分析

依據第三節中之系統架構，主要區分為買方、賣方和第三方，而本研究架構可視為整合端；在此，我們提供一整合平台讓企業能做連結也能架構在企業 IT 架構中。

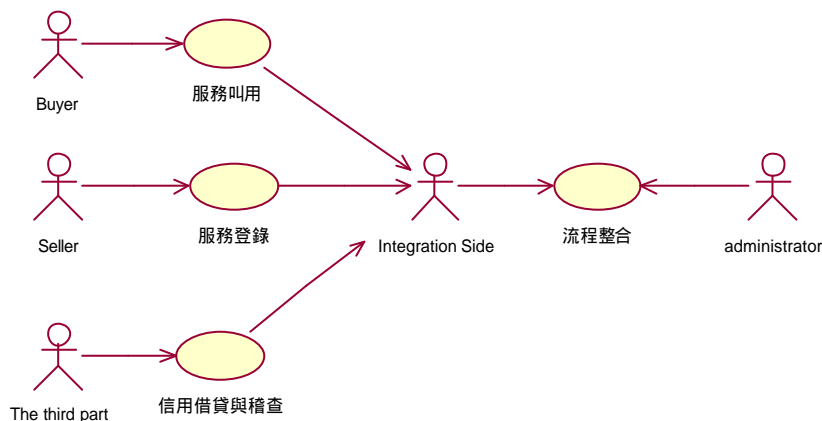


圖 13. 主要系統架構案例示意圖

(一)、買方

買方在本研究架構中的定義為商業協同合作中之服務需求者。在進行商業合作前，買方可先到 UDDI 註冊機制中搜尋符合本身需求之賣方，並進一步向欲合作對象提出合作請求，若合作方答應此項請求，便可進行下一步之協同協商並進行商業合作。倘若賣方不同意此項要求，便可在尋求其他服務提供者的合作。在雙方合作期間，

尚需第三方(銀行信貸機構)之參與，進行交易中信用借貸或財務往來之橋接。

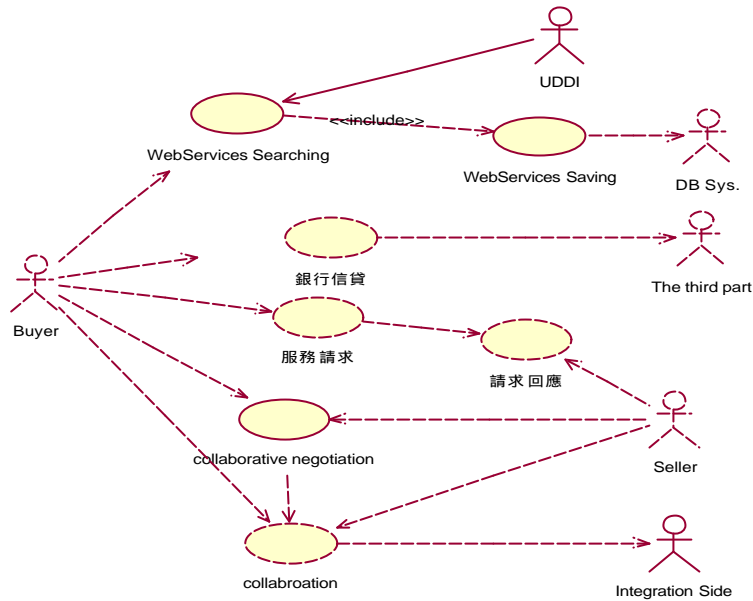


圖 14. 買方使用案例示意

(二)、賣方

賣方在此定義為協同合作中之服務提供者。當買方提出合作請求時，賣方可先至第三方銀行信貸機構查詢買方財務信用等事宜，在經過內部流程評估後決定是否與其進行商業合作。若願意接受其合作請求，便可進一步進行協商動作並開始雙方合作關係；若評估後認為合作請求不可行，便駁回買方服務請求。在合作期間，同樣也需要第三方銀行信貸機構之參與。

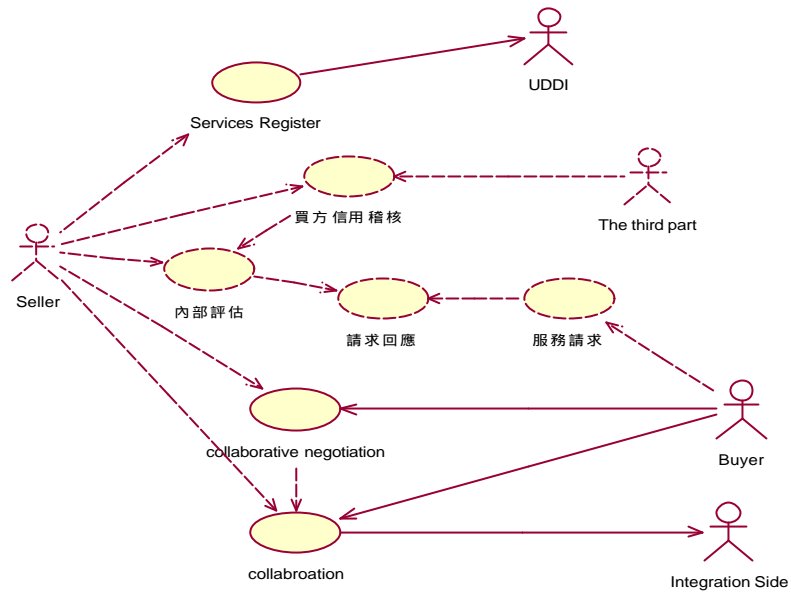


圖 15. 賣方使用案例示意

(三)、第三方

第三方在此定義為銀行等信貸機構，最為買雙方信用稽核、信用貸款或付款等等機制。

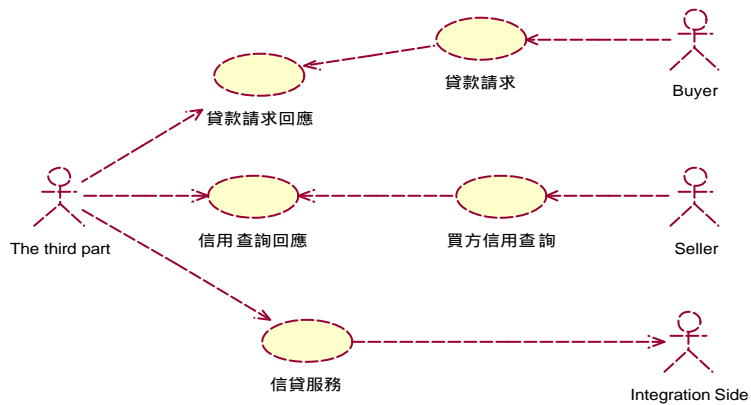


圖 16. 第三方使用案例示意

(四)、整合端

整合端也就是本研究之主要流程整合架構，當買賣雙方決定進行商業合作時，便可將流程所需之服務在此進行流程串接與部署等整合工作。在此部份之系統運行主要分為以下幾項機制：

1. 服務元件瀏覽登錄與分析機制

上一章節中，我們已對流程整合架構中之內部基本元件作好內部功能定義，在此機制中，我們設想企業已從 UDDI 伺服器中找到企業合作夥伴，此時便可將合作流程參與的服務元件，在此機制中進行服務元件分析動作，找出服務元件之介面定義，如：服務操作方法(屬性、參數等)、服務鏈結型態等等，並將分析結果儲存於資料庫中，便於之後流程進行時之服務調用與連結處理。

- Web Services Explorer

此使用案例主要作為對遠端 Web Services 的即時查詢與服務的登錄，合作雙方將協同合作所需之服務元件在此進行查詢與登錄後，並可藉由所輸入之 Web Services 的 WSDL 網址進行內容的解析，辨識該服務所擁有的屬性與行為等。

- Service Interface Operation Analyzing

此案例主要用來分析 Web Services 元件的介面操作的 Type、Message、PortType 等形態，得知操作 Web Services 的方法。

- Services Implement Analyzing

此案例為分析 Web Services 元件之實際調用部分，主要工作為辨識該服務之 port 所指向的終點服務網路位址，並與介面定義中 binding 作一關聯連結，啟動合作伙伴所提供之 Services 元件。

- Web Services Record Saving

此案例是針對前述對 Web Services 所作之分析記錄加以登錄儲存，以供後續回溯叫用時使用。

- Web Services Record Querying

針對整合端所登錄之服務元件進行細節資訊的查詢。

- Web Services Record Deleting

針對整合端所登錄之服務元件進行細節資訊的刪除動作。

- Web Services Record Modifying

針對整合端所登錄之服務元件進行細節資訊的編修動作。

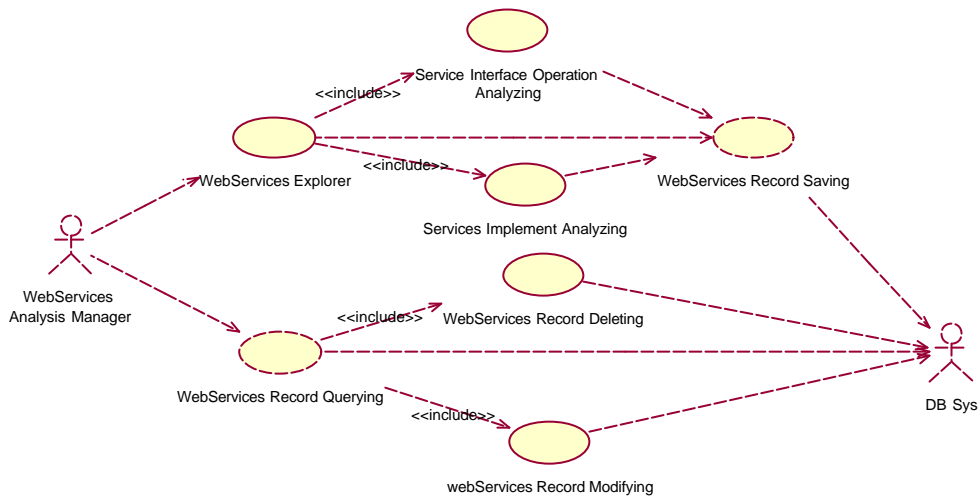


圖 17. 服務元件分析使用案例圖

2. 協同商業協議文件分析機制

此一機制主要針對合作雙方所制定之協同商業協議文件進行分析與儲存，包括跨企業業務中的所有重要行為部分，例如：訂單數量與交貨日期等資料相關行為、補償機制處理與買賣雙方間工作單元的調用方法、交易進行時之訊息傳遞定義與夥伴腳色定義等等相關資訊，以供流程串接與佈署時之參考依據。

- Data-Related Action analyzing

此使用案例主要為分析商業協議中的資料相關行為文件，例如：供應鏈協定中訂單的單項產品數量、訂單總價和訂單流程中之交付截止日期等達成商業目的所需之使用條件和超時構造，以及合作過程中交易角色之定義等。

- Exceptional conductions judging

此使用案例為定義商業交易中異常情況，如：流程異常時或夥伴請求撤銷等等例外情況之判定，以供流程執行時之判斷和例外處理程式之定義。

- Compensation Handling

對於上述例外處理使用案例所提供之補償機制處理，和補償程式之定義。

- Business Partner implement outcome invoke

商業流程長期運行(long term running)中，通常需要多個合作夥伴間內部工作單元的執行結果，如：內部流程執行成功或失敗的傳回值，本使用案例則為分析跨企業合作夥伴間工作單元的調用方法。

- Protocol Record Saving

此案例則針對前述之商業協同協議所制定之法規分析進行儲存動作，以便後續流程追溯時使用。

- Protocol Record Querying

針對協議之資料分析結果進行細節資訊的查詢動作。

- Protocol Record Deleting

針對協議之資料分析結果進行細節資訊的刪除動作。

- Protocol Record Modifying

針對協議之資料分析結果進行細節資訊的修改動作。

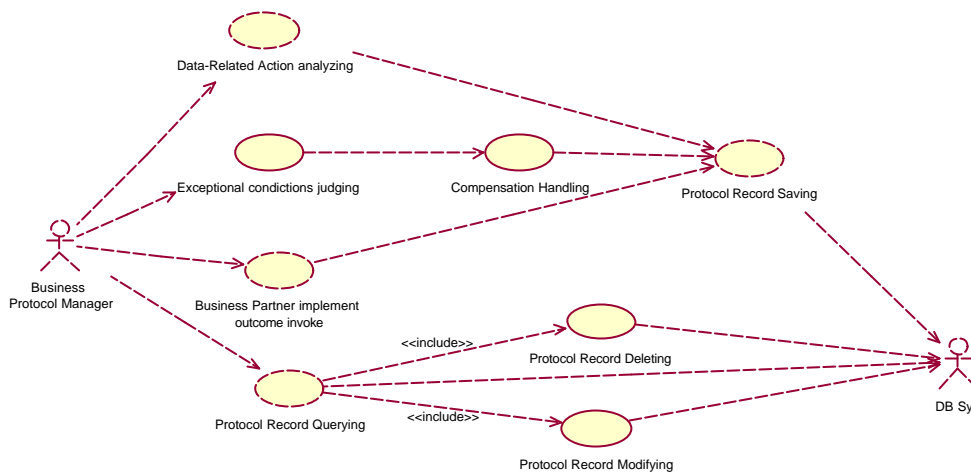


圖 18. 協同商業協議文件分析使用案例

3. 商業流程部署機制

此機制主要為商業合作交易中之抽象流程的佈署與分析。透過抽象流程的描述，BPEL4WS 流程可以定義商業協定中的腳色，如：買賣雙方兩個不同角色。在合作過程中，合作夥伴間都有自己的抽象流程，而抽象流程主要為處理和商業協議有關的資料。因此在此機制中，根據前一機制對商業協議文件之分析，把和商業協定有關的資料視為消息屬性的方式，做為合作夥伴間公開流程的服務鏈結和協定訊息溝通，而有關內部流程方面則使用不確定的值來隱藏流程中行為的私有部份。

- WSDL Document Definition

此使用案例主要為定義流程佈署之 WSDL 文件，亦即對抽象流程部分進行串接佈署動作，根據前一機制中的商業協議文件內容，定義合作角色所要遵循的消息交換協定。

- Message Defining

通常消息資料由應用程式資料和協定相關資料所組成。消息屬性則是被用於嵌入在商業協議和抽象業務流程中的可見應用程式資料中的屬性，此使用案例主

要用來定義流程中的消息屬性，屬性的定義主要用來創建(create)流程進行時的唯一名稱並使其和 XML Schema 類型產生關聯。此外也包括消息相關性的定義和流程資料的處理。

- PortType Defining

此使用案例主要用來指定不同服務間所提供的 portType，以供後續服務鏈結時的銜接調用。

- ServicesLink Defining

此使用案例用來描述兩個服務間的關係，並定義服務所扮演的腳色關係和指定每個角色所提供的 portType。ServicesLink 類型定義可以是獨立在一個 Web Services 的 WSDL 文件中的一個單獨的組成要素，也可以以放在定義 portType 的 WSDL 文件中。

- Business Process Deploying

此使用案例是將可執行流程部份依照 Web Services 元件分析機制中的 Web Services 介面操作和服務鏈結分析結果，並依照商業協議中的商業規則進行流程的邏輯佈署動作。

- Process-related Elements Defining

此使用案例為商業流程進行時相關流程物件和屬性的定義。主要包括流程屬性定義，如：活動或元素的刪除、Flow 的鏈結、鏈結目標等；還有作用域(Scope)內發生的錯誤、補償機制的處理、觸發相關活動的訊息指定、指定截止日期或設定的持續時間過後會觸發的相關活動執行、夥伴服務與商業流程的交互、商業流程狀態的消息儲存方法等等。

- Basic Business Processes Deploying

此使用案例為商業流程進行時之基本活動的佈署。基本活動主要為建構流程基本活動，如：接收服務、調用服務、更新容器(container)內容和發出故障信號或是終止服務實例等等。

- Structure Business Processes Deploying

此使用案例為流程進行時之結構化活動的佈署。結構化活動主要是規定了一組活動(active)的發生順序,描述商業流程是如何透過把執行的基本活動結構化而創建的,這些結構化的動作表達涉及商業協議的流程實例間的控制形式、資料流程、故障或外部事件的處理及消息交換的協調等。

- Business Process Execution Language Description

此使用案例主要是將上述流程部署相關動作匯集而成完整的 BPEL 描述語言,並將其放至流程工作引擎中執行。

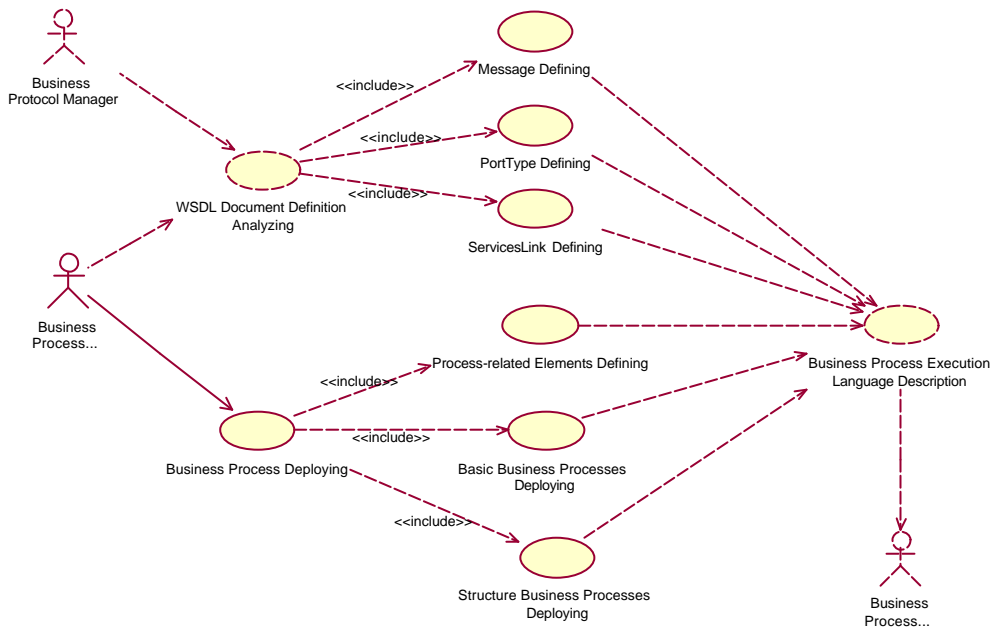


圖 19. 商業流程部署使用案例

4. 商業流程執行機制

此一機制主要為商業流程中可執行的流程實際執行過程。經由前一機制對抽象流程的具體描述和流程的完整定義,我們即可將以 BPEL4WS 所描述的商業流程透過商業流程執行引擎進行實際流程運行動作,而流程的參予者也能透過視覺化圖形介面監控流程執行狀態,進行例外情況的及時處理。

- Process Logic Analyzing

此依使用案例主要將前面流程部署機制中所定義好的抽象流程放置到商業流程執行引擎中作執行動作。

- Business Process Implementing Engine

此使用案例為商業流程執行工作引擎，為 BPEL4WS 語言執行環境，透過流程工作引擎能將先前描述之流程活動進行實例創建，並進行實際流程串接、服務調用和錯誤補償等工作。

- Process Status Monitoring

此一使用案例為流程狀態的監控，透過此，流程參予者能隨時監控流程執行狀態，若有例外請求或是錯誤產生時，便能及時作處理動作。

- Messaging

透過訊息的傳遞進行動作的判斷、銜接與服務調用等，因此，此使用案例專司流程執行時訊息繞境的動作。

- Dynamic Binding

此使用案例則為服務間的動態繫結部份，透過服務元件的完整界面定義和鏈結定義，便可進行服務間的動態繫結。

- Fault Handling

此使用案例為流程發生錯誤時，便將錯誤訊息傳至例外處理機制中作一錯誤處理和補償動作。

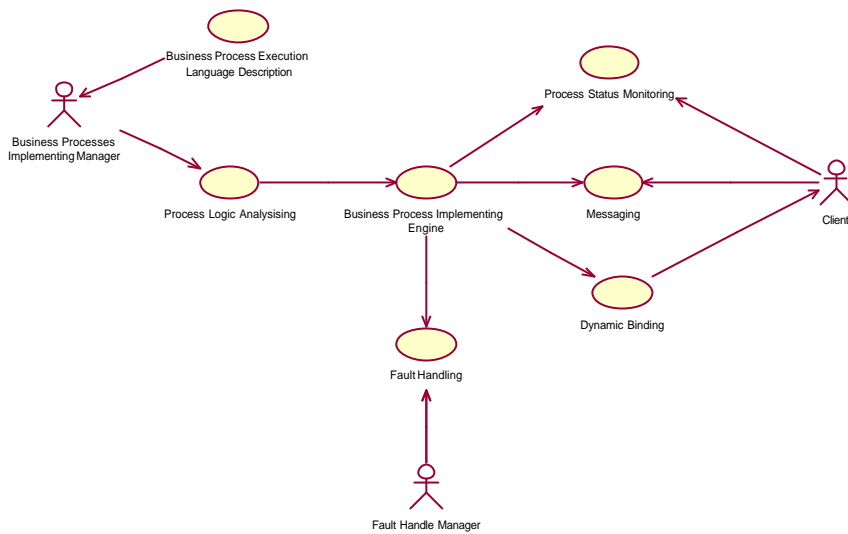


圖 20. 商業流程執行使用案例

5. 商業流程錯誤處理機制

錯誤處理機制主要是用來處理流程中不正常的運作情形或是流程參予者的例外請求等。

- Fault Handling

此一使用案例主要為流程執行時的錯誤偵測動作，當有錯誤情況產生的時候，便要判斷是要立刻終止流程或是進行修復並進行補償動作。

- Process Activities Ending

此使用案例負責錯誤發生時流程活動的停止動作。

- Fault Program Defining

此一使用案例主要是定義了一組制定故障處理活動的方法，如：Catch 活動能攔截某種故障情況，以供 Fault Handling 參考和處理。

- Compensation Handling

當流程有故障或錯誤發生時，可能會造成一些無法避免的損失，因此，此使用案例主要負責藉由錯誤補償的定義，進行補償程式的調用，以進行補償動作。

- Compensation program Defining

主要動作為補償活動的包裝，以供補償處理時的調用。

- Compensation Program Invoking

此一使用案例為補償程式的調用動作。

- Scope

在流程活動的執行時所需的 content，主要是由 Scope 提供，Scope 主要提供故障處理程式、補償處理程式、資料容器和資料相關集等。

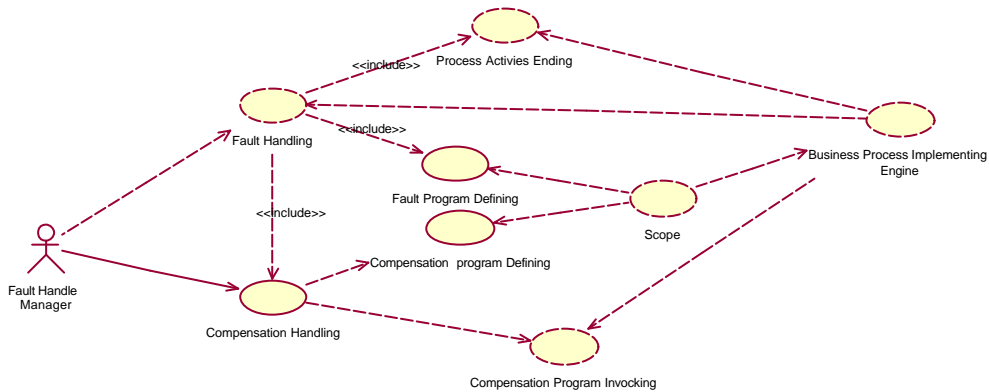


圖 21. 商業流程例外處理使用案例

第二節、系統元件之分析與設計

在前一節對流程整合端幾個機制的使用案例分析中，我們再針對個別使用案例進行類別的發掘和分類，對各類機制中的元件進行分析與設計。由於在整合端機制中某類別具有相同的操作屬性，因此我們先進行類別泛化動作將使用者管理操作類別定義一個通用的介面，相關類別再依其處理目標不同字型擴充新的屬性和方法。

壹、使用者操作管理介面類別

使用者操作管理介面主要提供商業流程參予者，亦即企業合作夥伴，透過身分認

證後對於流程狀況監控和流程處理所面對的畫面。在整合端機制中的每一過程都藉由單一介面方便始用者做額外事件處理和掌握長期運行交易流程的進度。

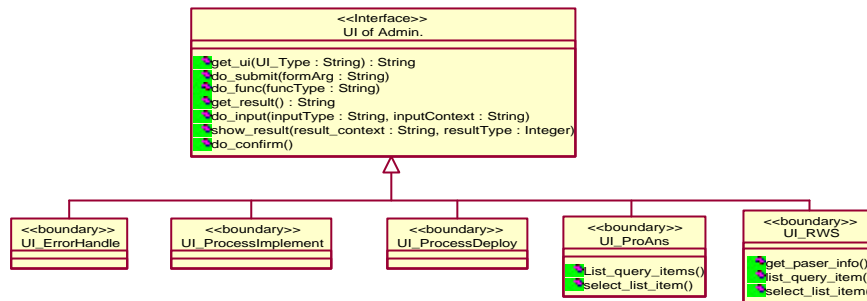


圖 22. 使用者操作介面泛化類別圖

二、整合端類別圖

根據各類別特性與在使案例的角色，產生以下類別圖以表達類別間的關係。

1. 服務元件瀏覽登錄與分析類別

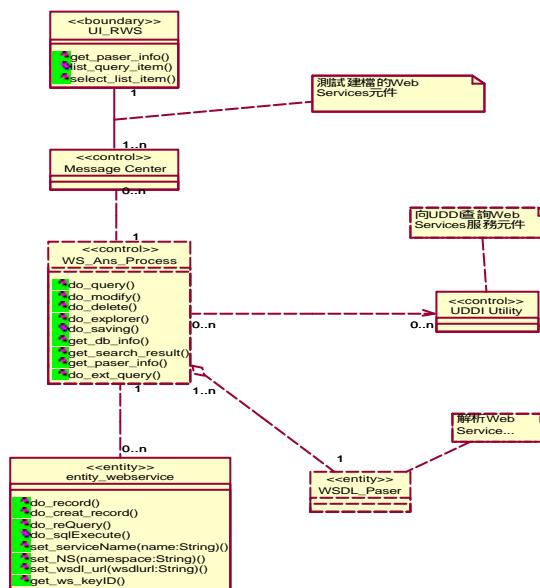


圖 23. 服務元件瀏覽登錄與分析類別圖

2. 協同商業協議規則分析類別

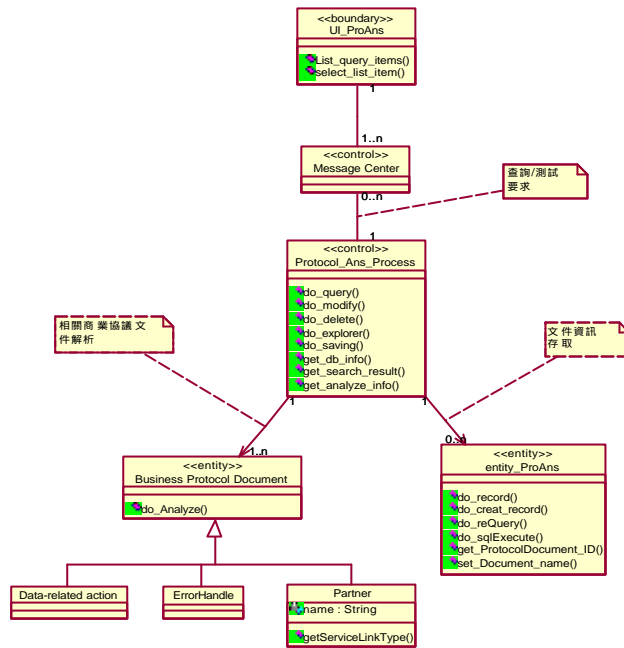


圖 24. 協同商業協議規則分析類別圖

3. 商業流程部署類別 [32]

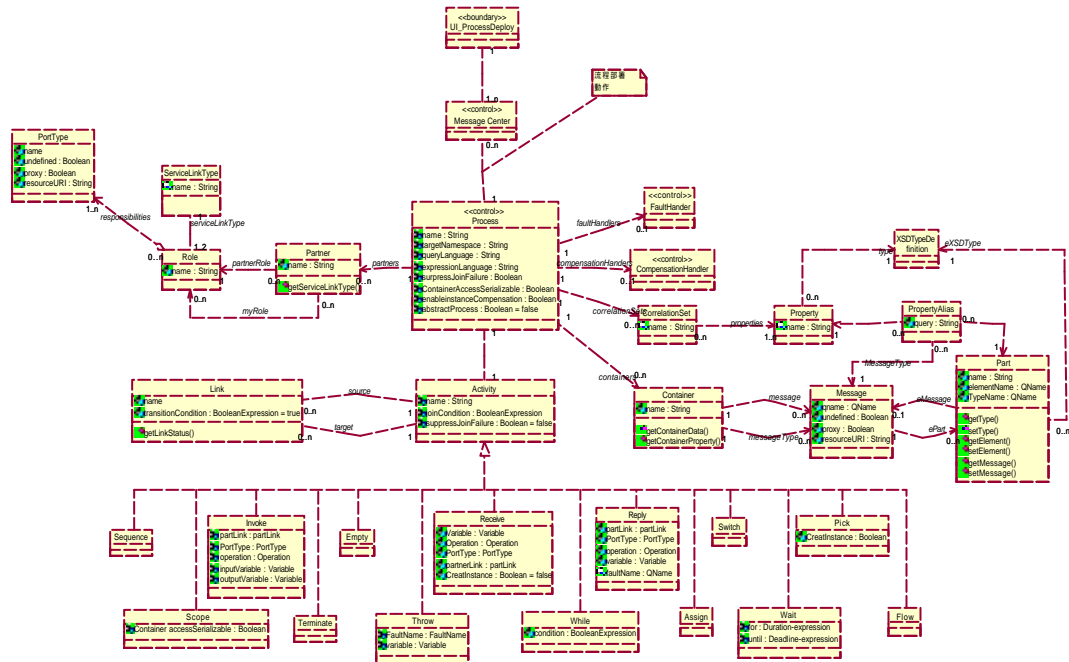


圖 25. 商業流程部署類別圖

4. 商業流程執行類別

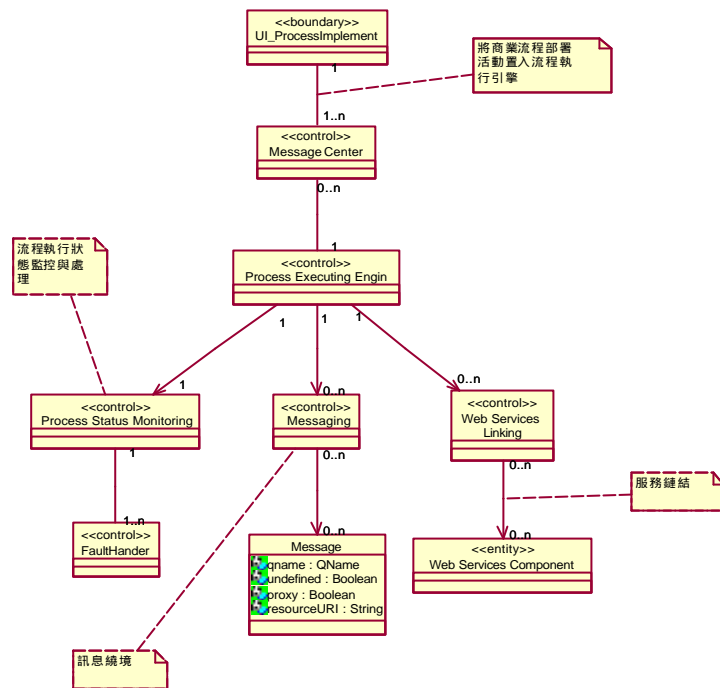


圖 26. 商業流程執行類別圖

5. 例外補償機制類別

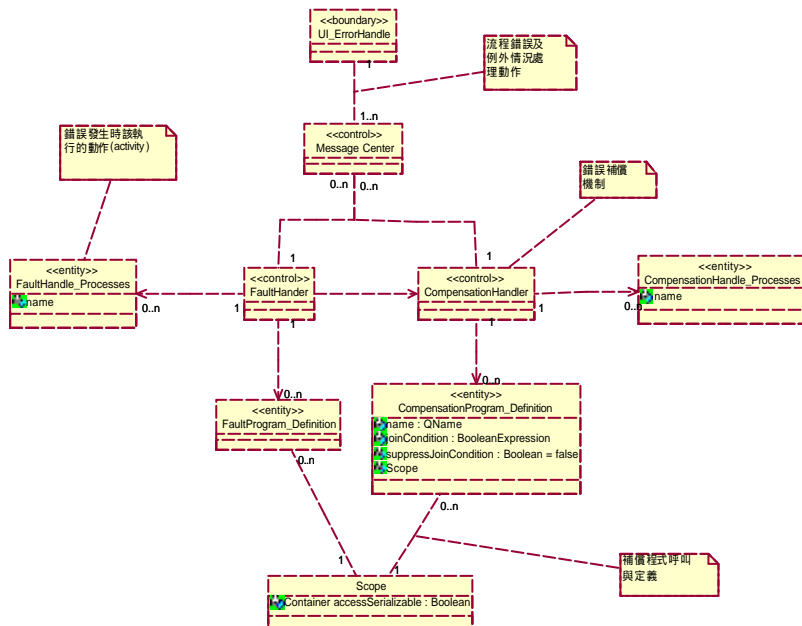


圖 27. 例外補償機制類別圖

第三節、小結

在此章節我們主要是利用塑模方式將流程整合系統以使用案例表現系統外部所能見之功能，依不同事件流程繪製成使用案例圖，再針對整合端各別機制之使用案例進行系統類別的發掘與分類，以便於進行第五章之系統建置。



第五章 跨企業商業流程整合架構建置

第一節、系統開發環境與佈署架構

壹、開發環境

本研究系統架構主要利用 Rational Rose 分析工具進行類別元件開發與設計；此外，為了讓商業流程能在跨企業夥伴間異質性系統模式下進行整合動作，我們藉由 Web Services 平台中立、可輕易穿越防火牆、標準化訊息與鬆耦合等特性，克服分散式物件技術的缺點；並採用 BPEL4WS 所具有的流程整合能力，進行企業夥伴服務模組間的流程串接動作，提供一個自動化平台架構，讓企業合作夥伴進行協同作業時能連結到平台上進行商業交易，減少商業活動進行時資料交換和溝通時間。

在系統環境部份，本研究之開發環境須有實現 Web Services 之能力，目前軟體業界所提供的環境主要有 J2EE 和 .NET，兩者比較如下所示：

1. Microsoft .NET[50]

根據 Microsoft 公開對 .NET 的定義，.NET 是為了要實現分散式運算的潮流。Microsoft 相信目前潮流正朝分散式運算轉型，.NET 的目的就是為了要加速下一代分散式運算的發展。

簡單地說，.NET 就是 Microsoft 為 XML Web Service 所提供的平台。XML Web Service 可讓多個應用程式透過 Internet 彼此通訊並共用資料，不論其作業系統或程式語言為何。Microsoft 認為，要實現分散式計算最快的方法就是是藉由以下三大方法：

- (1). Web Service 普及化 ；
- (2). 能夠歸納整合 Web 服務；
- (3). 能提供簡單且吸引使用者的操作環境。

NET 的目的即是推動這三項方法，使分散式計算更普及。

2. J2EE[51]

J2EE(Java 2 Enterprise edition) , 由昇陽與 IBM 等廠商協同業界共同擬定而成的技術規範, 是以企業間運算為導向的 JAVA 開發環境。J2EE 架構定義各類不同元件, 如 Web Component、EJB Component 等, 而各類元件具有可用性(, 讓已開發完成的元件, 或者是經由市面採購而得的元件, 能進一步組裝成不同的系統。在 J2EE 架構下, 開發人員可依循規範基礎, 進而開發企業級應用; 而不同 J2EE 供應商, 亦會支援不同 J2EE 版本內所擬定的標準, 以確保不同 J2EE 平台與產品之間的相容性。換言之, 植基 J2EE 架構的應用系統, 基本上可部署在不同的應用伺服器之上, 無需或者只須要進行少量的代碼修改, 即能大幅提高應用系統的可移植性(Portability)。

J2EE 的核心規範是 Enterprise Java Beans (EJBs)。EJB 依照特性的不同, 目前共分為三種, 分別是 Session Bean、Entity Bean 和 Message Driven Bean。其中 Session Bean 與 Entity Bean 算是 EJB 的始祖, 這兩種 EJB 規格在 EJB 1.x 版本推出時就已經存在, 而 Message Driven Bean 則是出現在 EJB 2.0 的規格中。

下表則為 J2EE 與 .NET 兩種開發環境差異比較:

表 3.Microsoft.NET 與 J2EE 開發環境比較表

	J2EE	.NET
技術型態	標準(Standard)	產品(Product)
解決方案參與廠商	包括 SUN 等 30 餘家廠商	Microsoft
Interpreter	JRE	CLR
前端動態網頁技術	JSP	ASP.NET
中介層元件	EJB	.NET Managed Components
SOAP、WSDL、UDDI	有實作能力	有實作能力
可實現平台種類	多個	單一

(資料來源: Byron Taylor Estes and Oriel Maxime, "J2EE vs .Net", COMPUTER WORLD)

經由上述發現，J2EE 標準具有較多參與廠商，且可實現平台種類也較多，對於企業間異質性環境來說具有較佳的解決方案，因此，本研究擬定以 J2EE 標準為系統開發時之系統軟體工具選擇之考量；在執行平台方面，則選擇 BEA 之 WebLogic 8.1 為系統執行平台，以實現 Web Services 元件的動態流程整合。

本研究之開發軟體為下：

1. 商業流程描述語言：Business Process Execution Language for Web Services Version 1.0；用來作為流程整合之部署與調用之描述。
2. 流程引擎：Collaxa - BPEL Orchestration Server；由於 BPEL4WS 為一商業流程描述語言，本身並無執行能力，因此需要流程引擎(Flow Engine)推動其流程執行。在流程引擎部分，本研究使用內建視覺化管理主控台之 Collaxa 公司所推出之 BPEL Orchestration Server，為實現流程整合系統之流程引擎。
3. Java 開發環境：Sun JDK 1.4.2_04；Sun 公司所推出的 Java 2 Platform, Standard Edition 主要為 Java 語言實現的開發環境。
4. Web Services 容器：Apache Axis axis-1_1rc2；用來掛載 Web Services 服務。
5. Java servlet 驅動器：Resin-2.1.13；用來裝載 Java servlet 的平台。
6. 資料庫：My SQL 5.10；作為開發程式碼的資料庫系統。
7. 網頁編寫與伺服：Macromedia Dreamweaver 4、Microsoft IIS 5。

貳、系統架構佈署

由前面章節對於系統架構之分析與設計，本研究主要利用企業入口網站(Portal Site)方式，設計一簡單的使用者操作介面，藉由單一介面讓使用者能較容易且方面利用本系統進行流程整合動作。

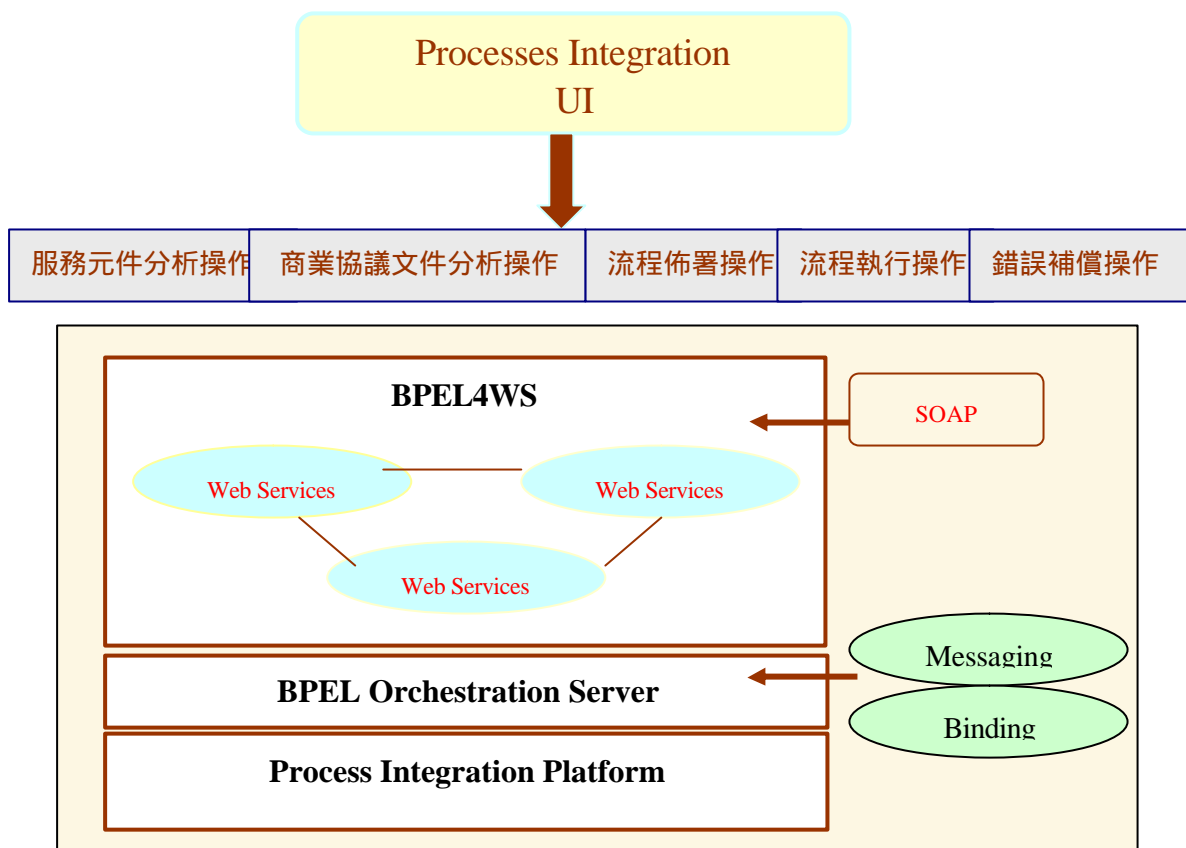


圖 28. 系統開發架構圖

當企業進行協同合作時，我們以賣方作為主導角色，其餘則為合作方角色，賣方可經由單一入口網頁，將有同一商業目的之協作夥伴所提供之 Web Services 元件網址先置入系統中，由系統進行服務元件的鏈結測試並將合作元件以 BPEL4WS 進行流程串接佈署並交由流程引擎進行運作與掌控。

下圖 28 為本研究預期進行實體佈署時的企業系統環境示意：

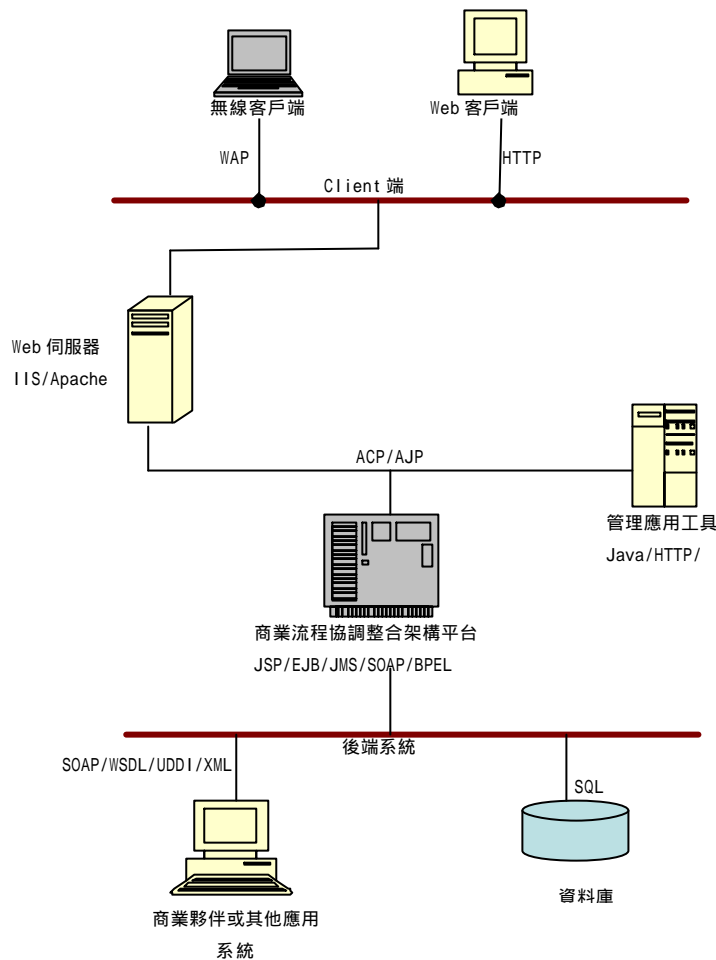


圖 29. 流程整合系統平台部署圖

第二節、系統建構與展示

壹、系統建構

進行系統建構步驟為下：

1. Web Services 紀錄分析與相關操作程序撰寫

此步驟主要為撰寫對流程參與的 Web Services 進行元件分析之功能，辨識出此元件之介面定義中所包含之 Binding、portType、Message 和 Type 等屬性參數資訊，方便流程整合實服務之調用。

2. 協同協議文件要素分析與相關程序撰寫

此步驟主要對於合作企業間所擬定之協同商業文件進行分析與儲存動作，如檔案來源、資料內容及合作或辦服務元件如何調用等，以供流程串接與佈署時之參考依據。

3. 流程部署相關屬性定義與流程串接順序排列

此步驟則根據前面機制所分析出來之資料進行抽象流程的描述與佈署，例如夥伴角色定義與鏈結束型態與訊息傳遞等相關屬性，並針對買賣雙方所制定之協同商業文件進行服務流程串接動作。

4. 流程執行相關動作定義

將上一步驟中所部署好之訂單流程置入工作引擎中執行，並監控流程執行狀況，以便例外情況發生時可即時作一處理。

6. 錯誤補償機制定義與相關程式撰寫

此部分對於流程本身可能發生錯誤情況進行錯誤捕捉定義並撰寫補償機制；而交易進行時，由買方或賣方因素所引起之例外情況，則針對雙方所擬定之錯誤補償機制為主，在此步驟先進行儲存動作，以供後續流程錯誤發生時之即時處理。

貳、系統展示

在系統建構部分，我們以一簡單訂單確定查詢流程為例；此例中，流程參與元件有 CRM Service 和 Checkout Flow。Checkout Flow 提供之服務為讓買方註冊資料、訂購產品等；CRM Service 則為賣方查詢買方資料及確認服務。下圖為系統之主畫面，底下再針對系統功能進行細部解說：



圖 30. 流程整合架構平台主畫面

1. 流程參與 Web Services 紀錄操作

在本範例中，流程參與服務夥伴元件主要有 CRM Service 和 Checkout Flow，我們則輸入此兩 Web Services 之 WSDL 網址進行分析動作與儲存，以供後續動作之參考。

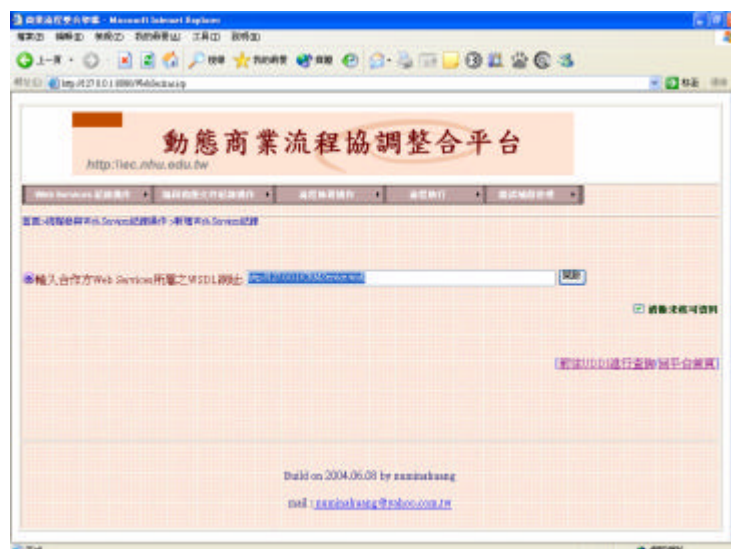


圖 31. 登入服務夥伴元件 WSDL 網址畫面

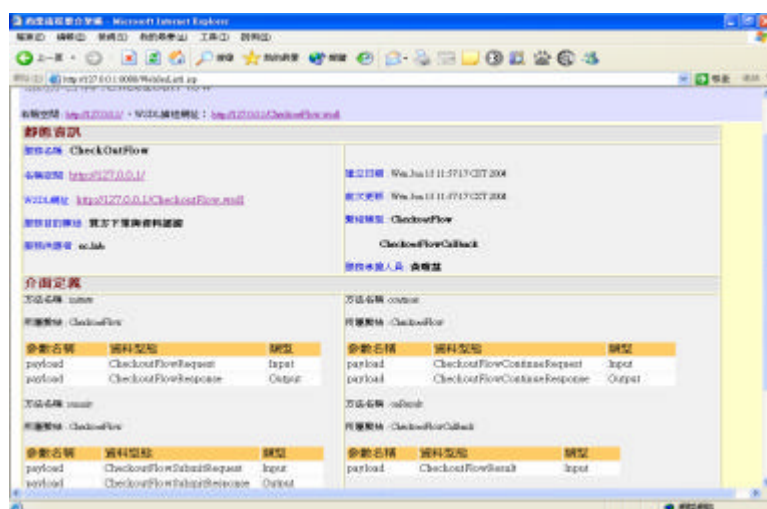


圖 32. 流程參與元件之 WSDL 分析

2. 商業協議文件紀錄操作

此步驟主要針對企業進行協同合作時所擬定之協議文件進行建檔與管理動作，可分析資料部份，在本示例中主要為買方之訂單資訊，不可分析之協議規則則為買賣雙方所同意之交易法規。我們則針對此兩份文件進行建檔與管理，以利後續流程進行時之查詢與安排。

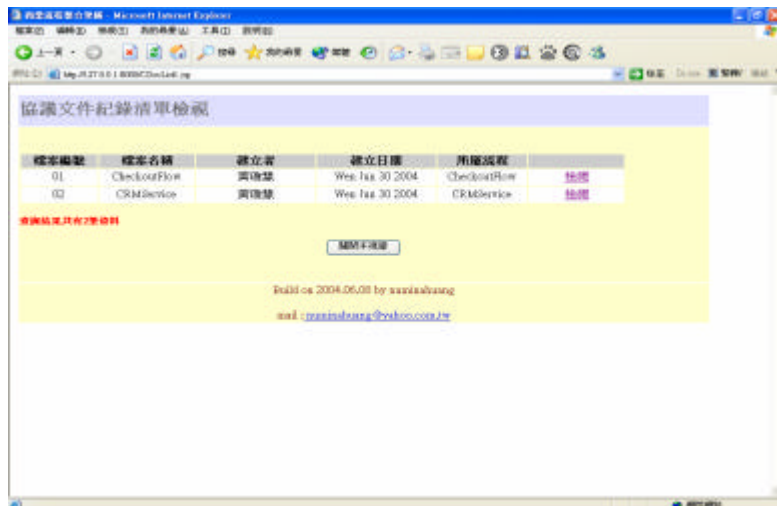


圖 33. 商業文件檔案建檔

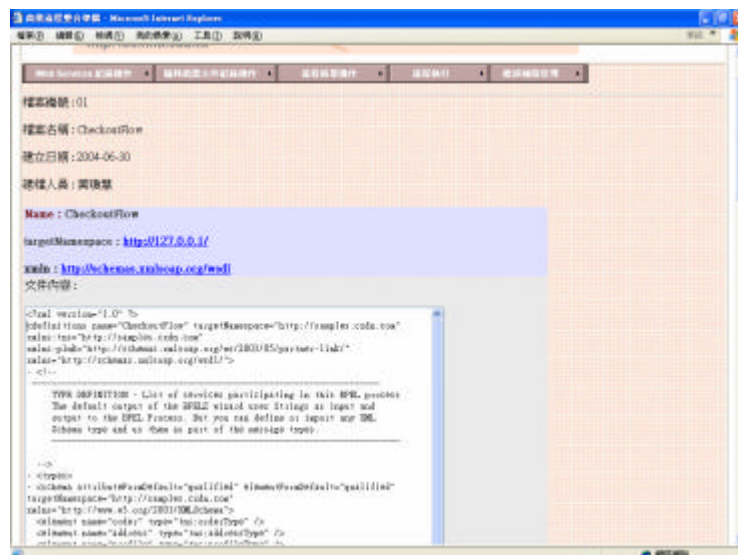


圖 34. 商業文件檔案查詢

3. 流程部署操作

目前支援 BPEL4WS 流程編輯器主要有 IBM BPWS4J 和 Collaxa BPEL Server。由於 Collaxa BPEL Server 有內建視覺化編輯畫面，因此我們採用此一編輯器作為流程編輯之選擇。



圖 35. 流程編輯器選擇與啟動

編輯 BPEL4WS 抽象流程先須先對流程之 WSDL 檔案作定義，因此我們即可將第一步驟中所儲存之服務夥伴 WSDL 文件置入編輯器中，再依據前面步驟所分析出細部資訊進行流程編輯。

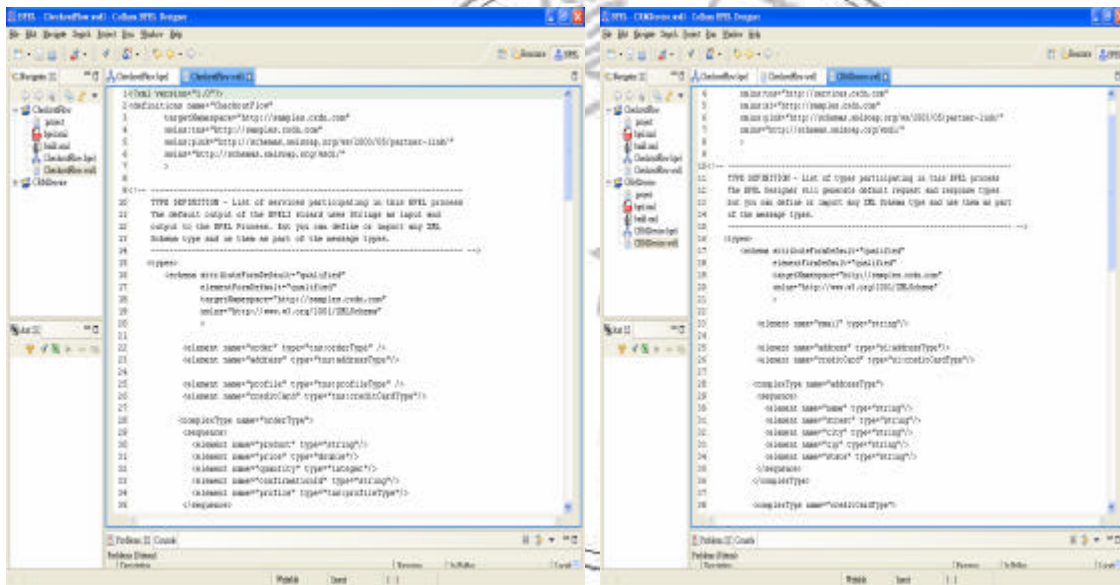


圖 36. 抽象流程 WSDL 文件定義

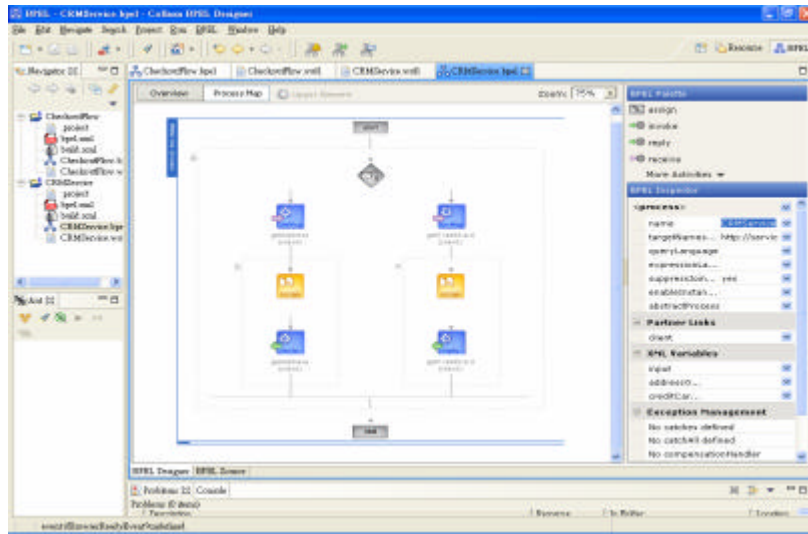


圖 37. CRMService 流程參數定義與佈署

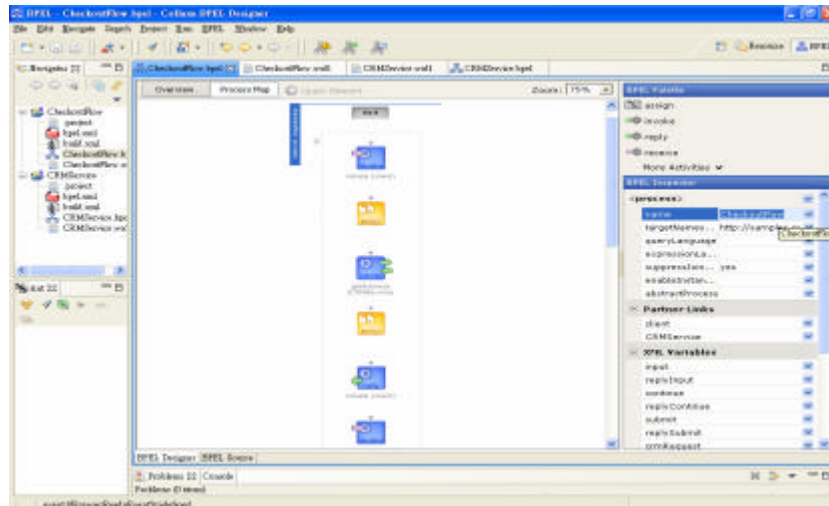


圖 38. CheckoutFlow 流程參數定義與佈署

4. 流程執行與監控

上述流程串接與佈署後便可將編輯好之流程置入工作引擎中執行並監控流程狀態。此步驟中我們藉由 Client 端買方操作介面與賣方流程管理監控界面作一對應，以表達流程進行時之動態表現。

流程開始時，買方會先進行資料登錄及定貨確認動作，賣方可藉由流程監控畫面得知買方之登錄狀態，若買方下單流程尚未完成，買方也可藉由監控畫面得知買方目前處於哪個登錄階段，以便做即時之處理。

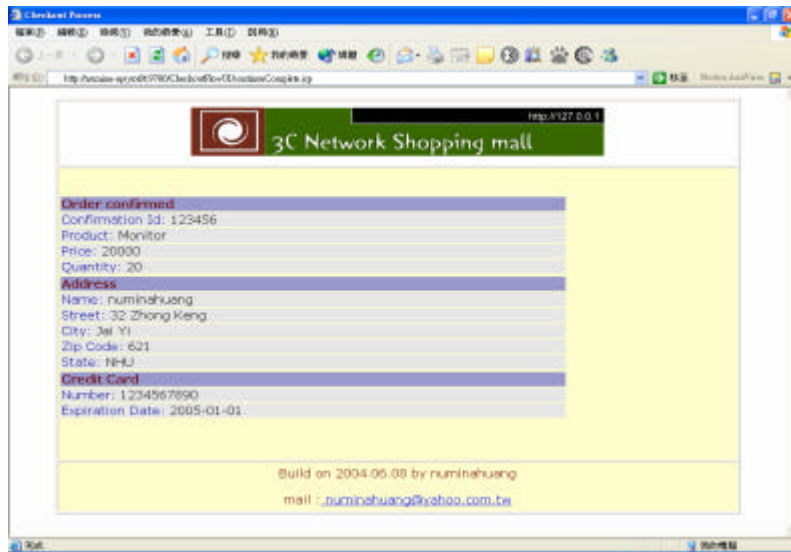


圖 39. 買方下單狀態

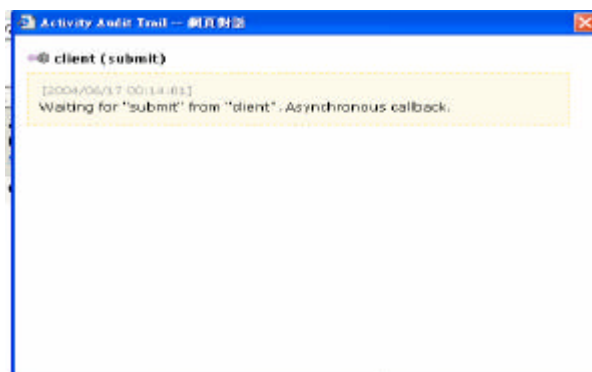
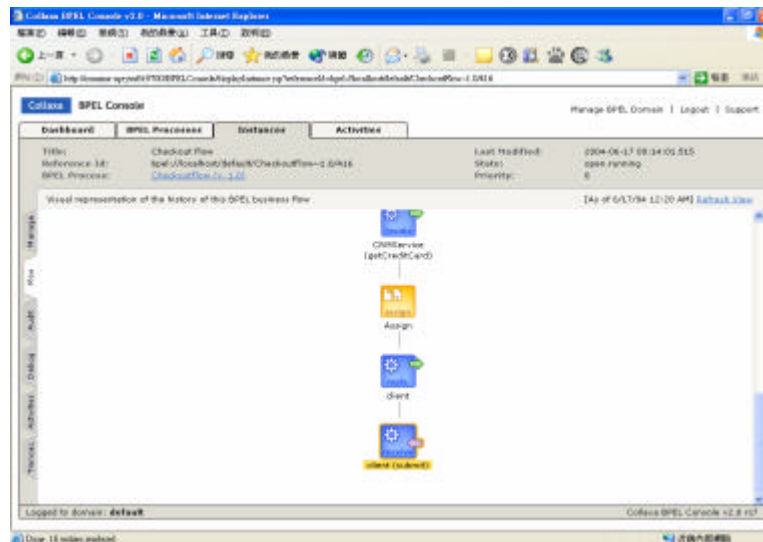


圖 40. 流程監控畫面及流程狀態

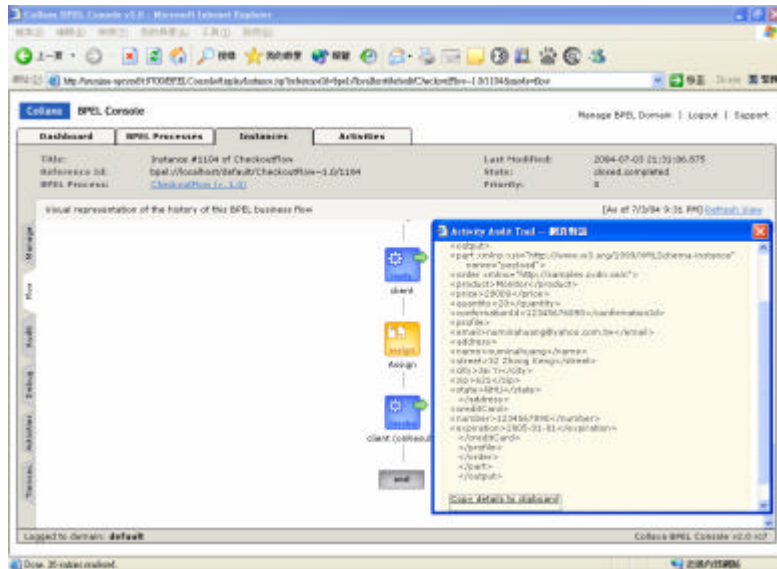


圖 41. 流程完全執行完之狀態

當流程完全執行完畢後，賣方便可藉由登錄之 mail 進行買方資料查詢，如：住址或是信用卡資料。

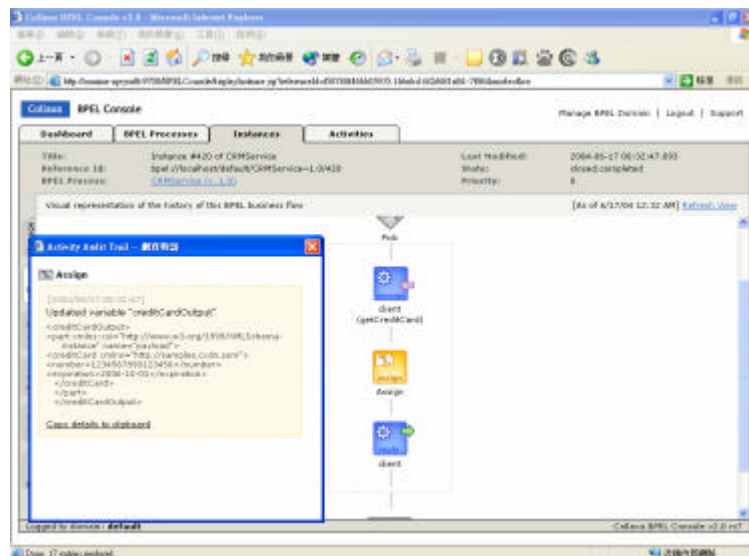


圖 42. 信用卡資料之查詢畫面

5. 錯誤補償機制定義

此步驟為定義流程內部錯誤與合作雙方所擬定之例外情況進行錯誤與補償程式撰寫，以供流程發生錯誤時之判斷與補救。由於目前所使用之 BPEL4WS 版本對於此部份交易細節仍尚未定義完全，如：交易進行中，若由買賣雙方所造成之流程終止，所需進行之交易補償定義等等，因此，本研究對於此機制並無進行實作。

第六章 結論與未來研究方向

第一節、結論

現今企業間的營運方式逐漸強調合作夥伴間的協同合作，也由於在協同作業的環境下，企業漸漸要求將 IT 技術和商業流程整合起來，以協同運作延伸企業觸角、提昇競爭力。

本研究主要提出一商業流程協調整合架構，期望透過本架構平台，實現跨企業夥伴間協同合作流程執行與自動化，提供企業作一選擇。在前一章節系統建構與展示部份，我們以一簡單示例驗證其可行性。透過本研究平台之單一操作介面，使用者可以將具有同一商業目標之流程參與元件與協同商業協議文件建置入本平台中，藉由一連串元件分析與流程串接佈署等動作，並透過流程引擎進行即時且動態之實際流程執行，在本研究示例中，使用者只需藉由視覺化之監控畫面，即可得知買方目前訂單確認動作之流程執行狀態與尚未執行完之後續動作。若流程執行動作超時，賣方也能及時進行處理；例如：是訂購系統本身問題還是流程問題，以便進行後續補償動作。

由於本研究之流程元件是以 Web Services 技術為基礎，企業間訊息傳遞只要符合 SOAP 通訊協定標準即能彼此溝通，而不受限於異質性環境問題，此外，Web Services 能以積木方式堆砌所需之服務元件，因此，以此技術為基礎之每個流程都可視為一個子流程，例如：系統示例中之訂單確認與查詢流程，亦可視為訂單管理系統中一子流程，藉由和其他發票流程、出貨流程等等可拼湊出更完整之商業流程，也適合協同作業下高彈性之流程變化。

第二節、未來研究方向

在商業流程描述語言部分，BPEL4WS 雖為 Web Services 技術提供更具實用性價值，在本研究中我們發現由於 BPEL4WS 尚屬發展階段，因此許多交易細節部分描述並不盡詳細，例如：錯誤補償方面；目前的 BPEL4WS 版本中，補償處理程式僅僅是補償

活動的包裝，在許多情況下，補償處理程式需要接收有關現實環境中當前的狀態資料並回傳關於補償結果的資料，因此，在未來的版本中也會針對此做更詳細的定義。

在傳輸安全性方面，因為流程進行中消息的傳遞可能有被竄改可能，針對此問題，未來研究時可以利用 WS-Security 來確保消息在傳輸過程中或駐留在目的地時不被竄改或偽造。

目前 Web Services 技術已經越趨成熟，相關配套技術也逐漸發展成型，如：WS-Coordination、WS-Transaction 以致於本研究所使用之 BPEL4WS 等；而產業間流程導向的整合標準主要有 Rosetta NET、ebXML 等。現階段將兩者加以結合之概念與作法仍在萌芽階段，因此若能將上述標準結合至本研究架構中，對於訊息傳輸與商業文件結構等等作更進一步的定義與呈現，相信會更趨完整。

參考文獻

中文部份

- [1] 毛立青,「協同商務之研究 - 以 TFT-LCD 產業為例」,元智大學資訊管理學系碩士論文,92年6月。
- [2] 王昌斌、黃書慶、黃瓊慧、胡莉玲,「以 Web Services 為基礎的企業應用整合平台」,2003 電子商務經營管理研討會論文集,逢甲大學,民國 92 年。
- [3] 王昌斌、黃瓊慧、黃天楷,「建構以 Web Services 為基礎的動態商業流程協調整合架構」,第五屆電子化企業經營管理理論暨實務研討會,大葉大學,民國九十三年。
- [4] 林章鈞,「協同商務的策略與遠景」,Taiwan_CNET,民國九十一年九月。
- [5] 陳世運,「淺談電子市集發展主流—協同商務」,FIND,民國九十年十一月二日。
- [6] 戚玉梁、彭淑芸、賴德優、張琪瑩, Web Services 探索與應用,全華科技圖書,民國九十二年五月。
- [7] 張裕益, UML 理論與實作—個案討論與經驗分享,博碩文化,民國九十一年二月。
- [8] 林宣佐,「運用企業流程語言描述關鍵企業流程之研究 - 以 BPEL4WS 應用於 CPFR 為例」,國立中正大學資訊管理研究所碩士論文,92年。
- [9] 陳慶蔚,「網路系統架構之競爭策略」,國立台灣大學商學研究所碩士論文,90年。
- [10] 陳曉萍,「企業電子畫下協同作業發展之研究」,國立政治大學經營管理學系碩士論文,91年。
- [11] 黃書慶,「建立以 Web Services 為基礎的應用整合架構」,南華大學資訊管理系碩士論文,92年6月。
- [12] 廖漢君,「UML:類別圖轉換至物件關聯模式之研究」,中山大學資訊管理學系碩士論文,91年6月。
- [13] 蕭瑞麟,「協同商務為 B2B 合作帶來新境界」,管理雜誌 332 期。

英文部分

- [14] Bongki Moon, Scott M. Baker, “Distributed cooperative Web servers”, *Computer Networks* 31, 1999. , pp 1215-1229.
- [15] Davenport, T.H., and Short, J.E. “The New Industrial Engineering : Information Technology and Business Redesign,” *Sloan Management Review*, Vol. 31, No. 04, pp.11-27, 1990.
- [16] David M. Peterson, “Power To The BPEL : A Technology For web Services”, *Business Communication Review ABI/INFORM Global*, pp 54, 2003.
- [17] Dinesh Ganesarajah, Emil Lupu, “Workflow-based composition of web-services : a business model or a programming paradigm?”, *IEEE Computer Society*, 2002.
- [18] Doug Davis, Glen Daniels, Steve Graham, Simeon Simeonov, Toufic Boubez, Yuichi Nakamura, Roy Neyama, *Building Web Services With Java*, SAMS, 2002.
- [19] Parodi A., Maresca M., Zingirian N., Baglietto P.,”Deployment of Service Oriented Architecture for a Business Community”, *IEEE Computer Society*, 2002.
- [20] Roller D., Leymann F., Schmidt M.-T., “Web services and business process Management”, *IBM SYSTEMS JOURNAL*, VOL 41, NO 2, 2002.
- [21] Eiji Yoshida, Kazutoshi Yokoyama, Shigeyuki Matsuda, “Requirements for Open Service Collaboration among Web Services”, *IEEE Computer Society*, 2002.
- [22] Hans Weigand, Willen-Jan, “Cross-organization workflow integration using contracts”, *Decision Support System*, Vol 33, pp 247-265, 2002.
- [23] Hirotaka Hara, Keisuke Yano, Sanya Uehara, “Collaboration Management Framework for Integrating B-to-B and Internal Processes”, *IEEE Computer Society*, 2002.
- [24] Howard Smith, “Business Process Management 101 ”, *BPMI.org* ,2003.
- [25] Jen-Yao Chung, Ying Huang, “A Web services-based framework for business integration solutions”, *Electronic Commerce Research and Applications* , 2 pp 15–26, 2003.
- [26] Medina-Mora, R., Winograd, T., Flores, R., Flores, F., “The action workflow approach to workflow management technology”, *Conference proceedings on Computer-supported cooperative work*, 1992, pp. 281 – 288.

- [27] Michael Hammer and Steven Stanton, “How Process Enterprise Really Work” ,Harvard Business Review, November-December, (1999).
- [28] Michael Harris, ”Defining : Collaborative Commerce”, EAI Journal, March 2002.
- [29] M. van Sinderen, J.P.A. Almeida, “Modeling of services and component Interactions”, Deliverable D3.2 – version 2, 2003.
- [30] Brian Blake M., “An Agent-Based Cross-Organizational Workflow Architecture in Support of Web Services”, IEEE Computer Society, 2002.
- [31]. Ping Jiang, Quentin Mair, Julian Newman, “Using UML to Design Distributed Collaborative Workflows: from UML to XPDL”, IEEE Computer Society, 2002.
- [32] Tracy Gardner, “Mapping from UML to the Business from UML to the Business Process Execution Language for Process Execution Language for Web Services (BPEL4WS) Web Services (BPEL4WS)”, OMG MDA Implementers’ Workshop Orlando, May 2003.
- [33] Shann Bayern, Mark A. Kolb, Duane K.Fields, Web Development with JavaServer Pages, 2nd Edition, Manning Publication Co. Nov,2001.
- [34] ”Best Practices in Collaborative Design”, white paper, Surgency, Inc. , 2001.

相關網站部份

- [35] IBM, ” 使用 BPEL4WS 的商業流程 ”
<http://www-900.ibm.com/developerWorks/cn/webservices/ws-bpelcol/index.shtml>
- [36] 林玉凡 “ 企業應用程式整合(EAI)產品之探討(上) ” ,
http://www.find.org.tw/trend/disp.asp?trend_id=1126
- [37] 林玉凡 “ 企業應用程式整合(EAI)產品之探討(下) ” ,
http://www.find.org.tw/trend/disp.asp?trend_id=1127
- [38] 李清培 “ .NET 研究室專欄 : Web services 之規劃策略與設計模式 – 企業觀點 ” ,
http://www.microsoft.com/taiwan/msdn/columns/dotNETResearch/webservicesent_00.htm ,2003.
- [39] 台灣電子商務研究中心 EC 報告 “ 電子商務與供應鏈管理之最佳化 ”
http://www.nii.org.tw/cnt/info/Report/20020401_01.htm

- [40] 鍾翠玲 “ 商業流程整合市場鬧滾滾 ” ,
<http://taiwan.cnet.com/news/software/0,2000064574,20044808,00.htm> , 2002
- [41] 鍾翠玲 “ 流程整合市場蓄勢待發 ” ,
<http://taiwan.cnet.com/news/software/0,2000064574,20077541,00.htm> , 2002
- [42] 蕭百齡 “ 組合式 Web Services 開發 ” ,
<http://taiwan.cnet.com/enterprise/column/0,2000062893,20085090,00.htm>
- [43] Barry Wilderman “ IT 管理：商業流程規劃 ” ,
<http://taiwan.cnet.com/enterprise/technology/0,2000062852,20067621,00.htm> , 2003
- [44] Web Services Architect
<http://www.webservicesarchitect.com/>
- [45] IBM Web Services website,
<http://www-136.ibm.com/developerworks/webservices/>
- [46] Sun Microsystems website,
<http://www.sun.com/>
- [47] BEA System,
http://www.bea.com/framework.jsp?CNT=homepage_main.jsp&FP=/content
- [48] Collaxa BPEL development documents,
<http://www.collaxa.com/developer.docs.html>
- [49] IBM BPEL4WS document,
<http://www-106.ibm.com/search/searchResults.jsp?searchType=1&searchSite=dW&query=BPEL4WS&searchScope=dW&Search.x=23&Search.y=10>
- [50] Microsoft .NET Information,
<http://www.microsoft.com/net/>
- [51] Java 2 Platform, Enterprise Edition (J2EE),
<http://java.sun.com/j2ee/>
- [52] Gartner website
http://www4.gartner.com/1_researchanalysis/research_overview.html
- [53] Aberdeen Group BPM research

http://www.aberdeen.com/ab_company/_vti_script/search_search.htm0.idq

[54] Hailstone, R. (2001.10), “Web Services – Vision, Hype and Reality,” Document# ES35H.

<http://www.idc.com>

[55] North America Web Services-Based Professional Services Forecast Update, 2003-2007 Sep 2003 , Doc #30066 , Market Analysis

<http://www.idc.com/getdoc.jsp?containerId=30066>

