

天主教輔仁大學資訊管理學系在職專班碩士論文

指導教授：葉宏謨 博士

服務導向架構與商務流程管理整合之個案研究

－ 以「新傳興業股份有限公司」為例

A Case Study on Integration of SOA and BPM

Systems at the Century Furniture Company



研究生：余嘉適 撰

中華民國九十七年七月

論文題目：服務導向架構與商務流程管理整合之個案研究 - 以「新傳興業股份有限公司」為例

校（院）系所組別：輔仁大學資訊管理學系碩士在職專班

研究生：余嘉適 **指導教授：**葉宏謨 **論文頁數：**167

關鍵詞：服務導向架構、商務流程管理、企業資源規劃

論文提要內容：

隨著市場的多變，企業需要更有彈性的資訊技術以協助進行商務流程的快速調變及軟體系統之間的整合運行，服務導向架構和商務流程管理系統的發展便是針對企業的這項需求而產生。

服務導向架構可以協助企業解決軟體系統整合的問題，而商務流程管理系統則可增加企業在商務流程管理上的彈性，企業若能整合這兩項技術，應該可以為企業的運行帶來顯著的效益。

本研究運用 SOA-ERP 系統及商務流程管理系統實作個案公司的商務流程，並經由實作的過程與結果，研究服務導向架構和商務流程管理系統對於企業資訊系統的改變及管理的效益。

本研究引用的 SOA-ERP 系統包含相當典型的 ERP 服務及商務流程，所引用的商務流程管理系統亦合乎 WS-BPEL 標準。然而因個案公司的狀況特殊，其商務流程與典型流程差異極大，而企業主亦認定其流程乃個案公司之最佳選擇，於是本研究利用標準的商務流程工具重新組合標準的 ERP 服務，滿足個案公司的特殊需求。本研究記錄實作的過程與心得，並依實證的結果推論企業在導入服務導向架構與商務流程管理系統之後，可能帶來的改變，以提供企業在採用服務導向架構與商務流程管理系統時的參考。

A Case Study on Integration of SOA and BPM Systems

At the Century Furniture Company

Keywords: Service Oriented Architecture 、 Enterprise Resource Planning 、 Business Process Management.

In the ever-changing market, enterprises need a more flexible information technology to facilitate the rapid change of business processes and the integration between business processes and software systems. This demand leads to the development of service oriented architecture and business process management systems.

Service oriented architecture can solve the problem of software systems integration and business process management systems can increase the flexibility of business processes. The integration of these two technologies is considered to be able to bring significant benefits for the enterprises.

This study uses a SOA-ERP system containing a set of typical ERP services and business processes. This research also applies a business process management system complying with BPEL standard to customize the business processes of the case company. The case company's business process, which is verified to be the best choice by the owner, is dramatically different from the standard business process of the SOA-ERP system. This research uses a business process management system to composite the services of the SOA-ERP system to implement the very special business process of the case company. This study documents the implementation process and the experiences found. This study also proposes suggestions of applying service oriented architecture and business process management systems for the enterprises based on the findings.

謝 詞

首先我要感謝我的指導教授 葉宏謨 博士這一年半以來的用心指導，並且在各方面給予協助，讓我在撰寫論文的過程中，不論是理論知識或是實務經驗上都能獲益良多，十分感謝葉教授的指導。

另外，要感謝論文的口試委員 鄒濟民 博士和 邱瑞科 博士兩位老師撥空前來參加口試評審並惠賜許多的寶貴意見，使得這篇論文可以更加的完備。

最後，要感謝我的家人，這篇論文的完成，遠比預期的時間還來的久，所歷經的狀況也不少，多虧有家人的全力支持與協助才能讓我無後顧之憂的完成這篇論文。



目 錄

	頁次
圖 次	vi
表 次	viii
第壹章 緒 論	1
第一節 研究背景	1
第二節 研究目的	2
第貳章 文獻探討	3
第一節 服務導向架構	3
第二節 服務元件架構	8
第三節 商務流程管理	11
第參章 研究方法	19
第一節 研究問題	19
第二節 研究架構	20
第三節 研究步驟	24
第四節 研究標的	25
第五節 個案選擇	25
第六節 資料收集方法	26
第肆章 個案研究	27
第一節 個案分析	27
第二節 系統設計	34

第三節 系統實作	39
第四節 實作心得	58
第伍章 結論與建議	66
第一節 結論	66
第二節 研究限制	68
第三節 後續研究建議	68
參考文獻	69
附錄一 出貨模組專案報告	73



圖 次

圖 2-1	服務導向的企業軟體系統架構圖	6
圖 2-2	服務元件架構模組實作示意圖	10
圖 2-3	商務流程與工作流程架構圖	15
圖 3-1	商務邏輯與程式邏輯架構圖	19
圖 3-2	研究架構圖	21
圖 3-3	NEO SOA-ERP 系統架構圖	22
圖 3-4	IBM 商務流程管理解決方案架構圖	23
圖 3-5	研究流程圖	25
圖 4-1	個案商務流程圖	31
圖 4-2	傳統製造業商務流程圖	33
圖 4-3	個案的商務流程圖	33
圖 4-4	NEO SOA-ERP 系統預設商務流程圖	34
圖 4-5	個案的實作商務流程圖	36
圖 4-6	完成後的系統架構圖	38
圖 4-7	本研究的實作流程圖	39
圖 4-8	NEO SOA-ERP 系統從工令到出貨的服務呼叫順序圖	46
圖 4-9	WSDL 檔匯入獨立的程式庫專案	48
圖 4-10	服務組合圖	50

圖 4-11	WSDL 檔匯入後依介面作相對應的資料型態	51
圖 4-12	定義出貨服務的介面	52
圖 4-13	WS-BPEL 視覺化編輯器	53
圖 4-14	服務流程設計圖	53
圖 4-15	出貨模組的服務呼叫順序圖	54
圖 4-16	本研究所定義的介面變數	56
圖 4-17	Assign 活動操作圖	57
圖 4-18	WID 的模擬測試介面	58
圖 4-19	相同名稱的服務會造成辨識困擾	61
圖 4-20	介面不相容使得服務無法由工具自動串接	63



表 次

表 4-1 NEO SOA-ERP 系統從工令到出貨的相關功能列表.....	39
表 4-2 NEO SOA-ERP 系統從工令到出貨的相關服務列表.....	40
表 4-3 NEO SOA-ERP 系統從工令到出貨的服務列表.....	41
表 4-4 NEO SOA-ERP 系統的銷售訂單查詢服務.....	42
表 4-5 NEO SOA-ERP 系統核心模組的服入庫與出貨服務.....	43
表 4-6 從工令到出貨的所有服務列表.....	44



第壹章 緒 論

第一節 研究背景

從多年在企業界開發應用系統的經驗和觀察企業導入軟體系統的過程裡發現，每一個企業的運行模式在同一產業之內，雖然大同小異，但是這小差異的部份，通常就是該企業在多年的營運下所發展出來的獨特運行模式，可能是企業文化所造成的，但也可能是該企業在市場競爭的環境下，為適應和生存所發展出的專有領域知識，是企業存續的重要資產。

過去的幾十年間，資訊界不斷的以「最佳實務 (Best Practice)」的觀念，倡導企業界推動商務流程再造 (BPR, Business Process Reengineering) 以協助 ERP 系統的導入，透過對軟體系統的最少修改，甚至是不修改的情況下導入軟體系統，以提高導入的成功率，不過，基於不修改軟體系統的前提下，企業個體就必須要調整自身的商務流程，以符合軟體系統裡所預先設計好的「最佳實務」，但是許多的企業並不是很清楚所謂的最佳實務是否真的適合它們。

然而，許多企業希望經由 ERP 系統的導入，讓企業能藉資訊技術的協助，提高其競爭力，於是便接受「最佳實務」的提案，對企業本身的商務流程進行改變，配合軟體系統的需要而調整，讓企業的商務流程適用於所採用的 ERP 系統，以確保 ERP 系統的導入可以順利、成功。

不過，企業在調整的過程裡，很容易的便會發生障礙，每一家企業都有自身獨特且已運行多年商務流程，要在短時間內做大幅度的調整，人員抗拒的情形便會發生，同時，相關配套及應變措施也要隨著研擬與實施，這都並非是短時間內可以達成的，因此，在「最佳實務」的窺咒下，企業在 ERP 系統的導入上，經常會面臨到許多的障礙與風險。

近年來，服務導向架構因應企業整合軟體系統的需要而發展。在服務導向的架構下，商務的運行邏輯被以服務的型式封裝在軟體的元件中，企業經由組裝服務而產生所需的軟體系統。但是，服務導向架構是一種軟體系統的架構，缺乏對商務流程的管理能力，商務流程管理系統的出現，補強服務導向架構在商務流程上的不足，並提供服務導向架構的應用價值。

第二節 研究目的

以服務導向架構和商務流程管理系統的特性而言，企業的軟體系統若採用服務導向架構結合商務流程管理系統的方式設計與運行，將會為企業帶來許多的效益。不過，就現況而言，由於服務導向架構與商務流程管理系統仍屬較新的概念與應用，企業實際使用的案例並不多，因此，服務導向架構結合商務流程管理系統的效益可以被運用在企業的那些作業上，仍舊存在著許多待探討與研究的議題。

本研究的目的即是透過運用已商品化的 SOA-ERP 系統及商務流程管理系統，配合實際個案的需求，將其實作，以實證服務導向架構整合商務流程管理系統可以增加 ERP 系統的商務流程調整彈性。

第貳章 文獻探討

第一節 服務導向架構

在過去，企業所使用的軟體系統是個別獨立運行的，例如，人事系統、客戶關係管理系統（CRM，Customer Relationship Management）、企業資源規劃系統（ERP，Enterprise Resource Planning）及辦公室自動化系統（OA，Office Automation）等，彼此之間並不容易進行整合，僅可透過資料庫連結或軟體廠商所提供的程式界面（API，Application Programming Interface），以撰寫程式的方式達成跨系統整合的目的。

前述的軟體系統整合方式會受限於各軟體廠商所採行的標準和提供的開發工具，對於複雜環境下的整合需求，幫助有限，例如，並非所有的軟體廠都對不同的作業系統提供可用的程式界面，若企業的整合需求遇到此種情況時，便必需找軟體廠商或協力廠商協助撰寫客製的介面來解決此種問題，而在客製程式的需求下，企業整合的進度必定會受到延宕，更可能因此而造成企業競爭上的劣勢。

企業需要的是具彈性及易整合的資訊系統架構，可以隨著企業運行環境而動態進行改變與重組的一種架構，服務導向架構（SOA，Service Oriented Architecture）便是基於整合與彈性的基礎所建構的資訊系統架構，在此種架構下，企業運行所需的各種商務功能（Business Function）由封裝著商務邏輯（Business Logic）的

服務元件（Components）所組成，而傳統、個別運行的軟體系統將不再是企業軟體系統的中心，而且也不再是系統運行的邊界與分野，而是轉變成提供基礎服務元件的程式庫，成為企業建構軟體系統的素材。

一、服務

服務是服務導向架構裡的基本組成單位，同時也是程式碼真正儲存的地方，服務可以視為是一種軟體元件，它經由物件導向的程式設計方法，將各種邏輯封裝在一個元件裡，而此元件所封裝的內容，可以是一個商務工作、個體或其它可被程式化的邏輯（Erl,2005）。

從上述對於服務的描述看來，服務與物件導向裡的物件類同，但是，Erl（2005）指出，在服務導向架構下，一個服務至少需具備下列特質才能符合服務導向架構運行的要求：

1.鬆散耦合（Loose coupling）

服務之間只需維持最小的相依程度，僅供服務之間能用來感知對方的存在即可。

2.遵守合約（Service contract Services）

服務必需依循由單一或多個服務所共同描述的通訊合約。

3.自治（Autonomy）

在服務封裝內的邏輯由其自行掌控與管理。

4.抽象描述（Abstraction）

服務的描述是抽象的，而且不同於服務合約（Service Contract）中所記載的內容，因為服務所封裝的程式邏輯不

會對外提供直接存取的能力。

5.可再用性 (Reusability)

將具可再用性的程式邏輯切割成不同的服務，以達成可重複使用的效果。

6.可組合性 (Composability)

服務可以經由組合而形成另一個服務。

7.無狀態的 (Statelessness)

服務將儲存最少量的特定活動的資訊。

8.可被發現的 (Discoverability)

服務是被設計成可以被明白敘述的，如此情況下，服務才能被發掘機制 (discovery mechanisms) 所發現。

二、服務導向

企業使用的軟體系統所包含的程式邏輯可分為商務邏輯 (Business Logic) 和應用邏輯 (Application Logic) 等兩大領域，在過去的架構下，企業的資訊系統都必需經由分析瞭解後，配合企業個別的資訊架構，撰寫所需的應用系統。

而服務導向的做法，是將企業的軟體系統劃分成三個層，由商務流程層 (Business Process Layer)、服務層 (Service Layer) 及應用層 (Application Layer) 組成 (如圖 2-1)。最底層的應用層即是傳統軟體應用系統中的商務邏輯，屬於實際封裝程式邏輯的一層，經由上一層的服務界面將其組裝成服務後，提供給最上層的商務流程層使用。

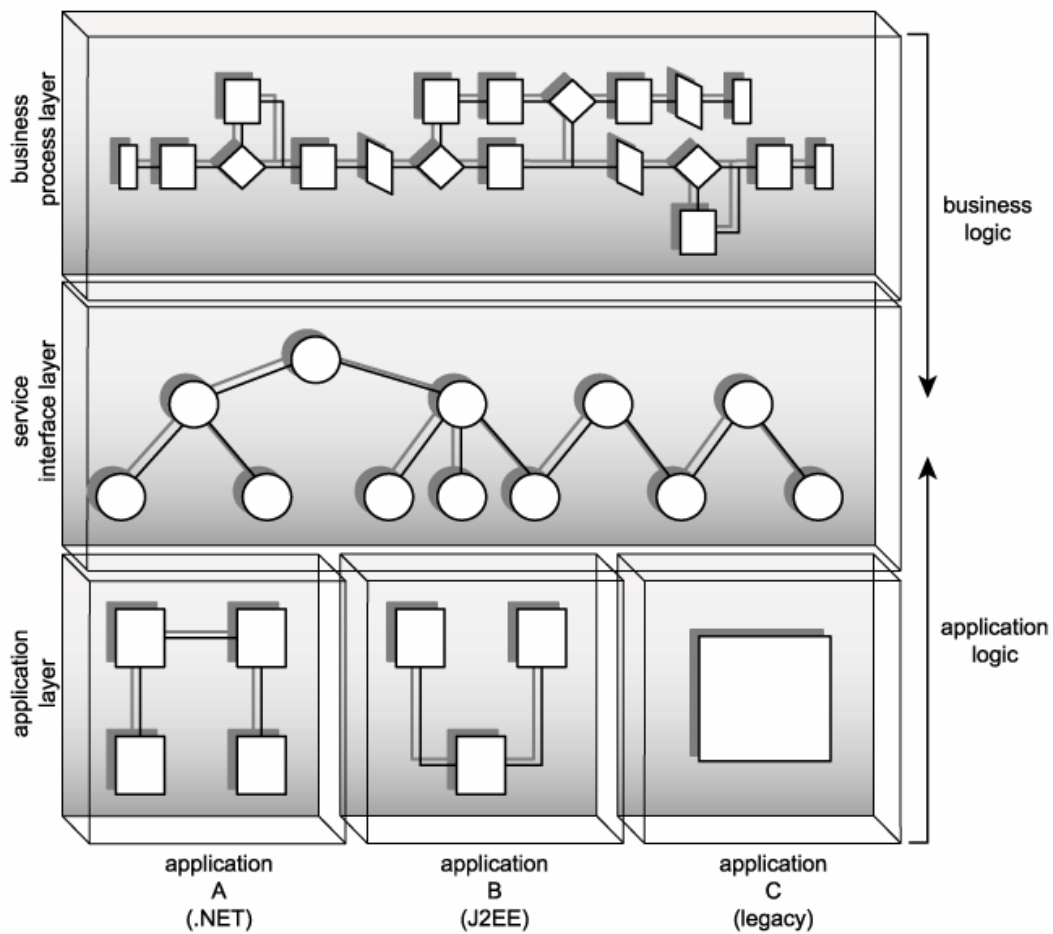


圖 2-1 服務導向的企業軟體系統架構圖

資料來源: Erl,2005

也就是說，在服務導向的架構下，企業所需的軟體系統是透過服務組裝的方式來達成，服務可以被任意的拆解與組裝成所需的商務流程，不同於傳統方式，需經由程式撰寫和修改來達成企業所需的商務邏輯。

三、架構

架構可視為是一種抽象化的範本，所有使用此範本的解決方案均需依照範本定義的內容來應用，其包含技術、範疇、規則、限制和設計的特質等項目，而服務導向架構是用以描述資訊系統的一種架構，而不是標準或框架。

四、網路服務

在討論服務導向架構時，許多人都會同時提到網路服務 (Web Services)，不過，服務導向架構與網路服務是兩種不同的資訊技術，服務導向架構並不等於網路服務，一套運用網路服務相關標準與技術所設計開發的軟體系統，並不能被歸類為服務導向架構的軟體系統。

Erl (2005) 指出，網路服務有許多的特質適合用於實作服務導向架構，因為兩者皆是基於以服務為基礎的運行方式，網路服務不是一種架構，而是一個框架 (Framework) (Erl,2005)，它是由一系列的資訊技術標準所組成的框架，這些標準被用來作為服務與服務之間的介面，並且提供服務之間的通訊協定 (Josuttis,2007)，這和服務導向架構是一種架構的本質是不同。

就服務導向架構而言，網路服務為服務導向架構的服務提供實作的介面和協定，透過網路服務所提供的各項標準使得服務導向架構的服務可以容易的被使用和整合，然而，這不表示服務導向架構一定需要經由網路服務才能被實作 (Josuttis,2007)，服務導向架構仍舊可以經由其它方式實作。

五、服務導向與物件導向

服務導向有許多的特性是根源於物件導向 (Erl,2005)，這使得服務導向與物件導向之間存在著許多模糊而相似的特質，不過，兩者之間仍舊存在著差異，例如，服務導向並沒有「繼承」這項特質，而物件導向則不具有「鬆散耦合」與「自治」的能力。

服務導向與物件導向之間的差別並不影響彼此共同存在或是混合應用的能力，相反的，此兩者之間是存在著互補與互相支援的關係 (Erl,2005)，而此種互補性將有助資訊系統從現行以物件導向為主的架構轉換到服務導向的架構。

第二節 服務元件架構

服務元件架構 (SCA, Service Component Architecture) 由是多家軟體廠商組成的開放服務導向架構協作組織 (OSOA, Open Service Oriented Architecture Collaboration) 所共同制定的服務導向架構標準，服務元件架構的制定是希望能夠為服務導向架構提供一個公開、開放的程式開發標準。

服務元件架構是一種以服務導向架構為基礎的程式設計模式，被運用於建構企業所需的應用系統和解決方案。服務元件架構的概念是將商務功能視為是由一系列的服務所提供的，將這一系列的服務組裝後即可產生用以解決特定商務需求的軟體系統。這種經組合而成的軟體系統可以包含針對特定功能設計的新服務元件，也可以包含已經存在於現行系統的功能。

一、服務元件架構的組成項目

服務元件架構包含下列項目 (Karmarkar et al.,2007)：

- 1、組合模式 (Assembly Model)：說明如何定義組裝式軟

體系統的結構。

2、元件實作規格 (Component Implementation Specifications) : 說明如何在不同電腦語言下撰寫商務服務元件的實作方式，如 Java, Spring, C++, WS-BPEL, PHP.....等。

3、繫結規格 (Binding specifications) : 說明如何使用現有的存取方式，例如，Web Services、JMS、RMI-IIOP、REST.....等存取服務元件。

4、政策框架 (Policy Framework) : 說明如何新增基礎架構性質的服務到組裝的軟體系統中，這些基礎架構包括：安全控管、資料交換及穩固的訊息傳遞等等。

二、服務元件架構的開發方式

模組 (Module) 是服務元件架構的基礎開發單元，也是實際部署使用的最小單位，每一個模組都必需包含匯出 (Export) / 匯入 (Import)、介面 (Interface)、參考 (Reference) 及實作 (Implementation) 的服務元件 (Service Component) 等元素 (如圖 2-2)。匯出和匯入是模組提供給其它服務使用或是使用其它服務的介面，模組可以經由匯出介面將自己提供給其他它模組匯入使用，同樣的，模組也可以經由匯入的方式將其它服務元件或模組匯入供自己呼叫使用。

模組的介面是模組被呼叫時的輸入和輸出界面，其作用和服務的界面一樣，因為，模組本身也一個服務元件。參考則是模組與其他模組或服務元件間溝通的方式。

每一個模組都必需包含一個實作的服務元件，實作的內容即

是服務組裝的方式，可以用 WS-BPEL、Java、C++ 或 PHP 等語言進行實作。

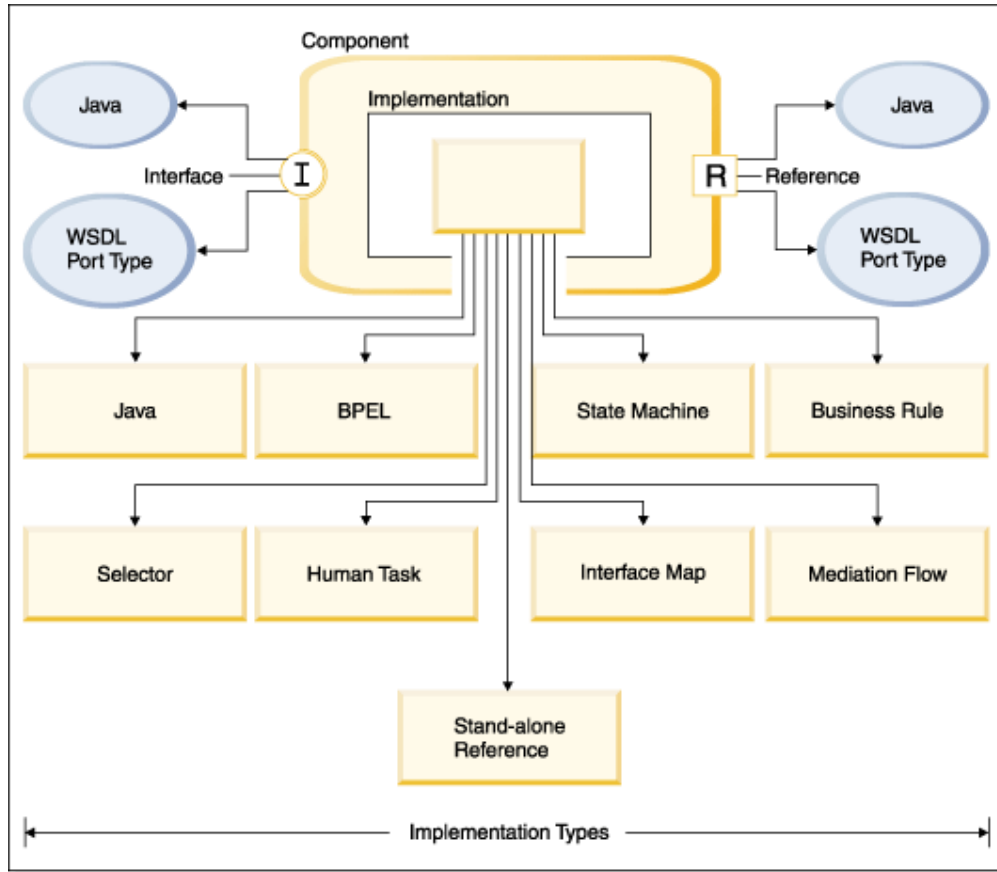


圖 2-2 服務元件架構模組實作示意圖

資料來源: IBM WebSphere Integration Developer 6.1 版使用手冊

三、服務元件架構的商業價值

服務元件架構是以服務導向架構為基礎的程式設計模式，除具有服務導向架構的各項優點外，對於運用此模式的企業可以提供下列商業價值（Weaver,2005）：

（一）、增進彈性

1、技術中立性：企業可以依需要更換實作的技術或協

定，而不需更改以服務元件架構所開發的系統。

- 2、可重覆使用性：服務元件架構的服務元件同樣實作服務導向架構的鬆散耦合特性，因此服務元件可以容易被運用在商務流程或組裝式的軟體系統中。
- 3、可組裝性：服務元件可經由服務元件架構的組合機制建構龐大或複雜的組裝式軟體系統。
- 4、高度調適性：商務分析人員可快速更換並測試新的系統運行模式，不需經由長時間的等待。

(二)、提高程式生產力

- 1、同步設計的能力：基於鬆散耦合的服務模式，服務元件與服務元件之間不存在絕對的關聯性，服務元件可以平行的各自獨立分開設計及開發，節省開發時間。
- 2、服務的重覆使用能力：服務元件架構不只可以建構新的服務元件，也可利用現存系統的功能組裝新的軟體系統，重覆利用現有系統功能。
- 3、由上而下的開發能力：軟體系統可以在組裝完成後再進行服務元件的開發，而不需要等個別的服務元件開發完成後才進行組裝，這使得服務元件的設計可完全依需求而設計，避免時間的浪費。

第三節 商務流程管理

近代商務流程管理（BPM，Business Process Management）的興起，應可追溯自 1990 年 Michael Hammer 在哈佛商業評論所發表

的“Reengineering Work: Don't automate, Obliterate”一文開始，並在 1993 與 Champy, James 出版“Reengineering the Corporation: A Manifesto for Business Revolution”之後，讓商務流程的管理開始為企業所重視。

在軟體廠商的產品介紹中，商務流程管理經常被視為是一種資訊技術或是軟體系統，在部分探討資訊科技相關的文獻裡，也認為商務流程管理是和技術有關的（Bloomberg and Schmelzer,2006）或者是由資訊技術所組成的（Behara, 2006）。不過，以商務流程管理興起的時機來看，資訊技術並不是商務流程管理的必要條件，資訊技術是輔助企業達成商務流程管理的一種工具。

商務流程管理不是一種資訊技術，也不是一套軟體系統或模式（Jeston and Nelis,2006），商務流程管理是一門管理的學科（Harmon,2005）。企業經由對其商務流程的管理達到提高企業營運效能的目的，凡是支援商務流程分析、設計、管理、配置及規範的觀念、方法和技術都可以被納入商務流程管理的範疇內（Weske,2007）。

一、商務流程

組織就像是一個轉換系統，將取得的人力、資金、材料或商品等輸入，經由一系列的加值活動予以轉換後，輸出服務或產品，而這一連串的、有順序的、有目的的加值活動即是商務流程（Rosemann,2001）。

實務上，加值的活動在企業裡是由各種不同功能的部門所運行的，這些加值活動被區分成許多不同的商務功能，例如製造、

會計、人事等，商務流程的實際運行即是由各種商務功能所組合而成。

另外，商務流程的發生並不侷限於組織內部的活動，同時也會發生在組織與組織之間的加值活動上。

二、工作流程

商務流程會隨著企業經營環境的改變而不斷的修正，因此，軟體系統必需不斷的配合修改與調整，才能維持軟體系統內的流程能與實際需要相符，但是，軟體系統的程式修改在傳統系統架構下，都是一段漫長而且具有風險的變更作業，因而導致企業為降低風險，必須儘量減少軟體系統的流程變更，或是改以人工作業的方式取代已存在的軟體作業，此種作法與結果明顯地的降低軟體系統導入的價值，同時也失去商務流程管理的目的。

為改善軟體系統裡，流程變更困難的問題，工作流程管理系統（Workflow Management System）於是被發展出來。工作流程（Workflow）指的是一連串經由電腦輔助所達成的自動化商務流程（Hollingsworth,1995），而工作流程管理系統則是指能夠定義、管理及依序執行工作流程的資訊系統（Hollingsworth,1995）。軟體系統具備工作流程管理功能之後，企業對於存在於軟體系統裡的自動化商務流程便可方便的進行變更與控管，進而增加軟體系統在流程管理上的彈性與效用。

工作流程管理系統是一種管理資訊系統，用以管理可以被自動化的活動及步驟，對於非自動化的人工活動並未納入管理，與以企業整體流程為管理對象的商務流程管理是不同的。

三、商務流程管理系統

商務流程管理系統（BPMS，Business Process Management System）是一套軟體系統，它將商務流程具體的表現在軟體系統裡，以利進行監視、控制與管理。

商務流程在企業內是以各種不同的型態被執行著，有人工的、機械化的、或軟體系統的，這些不同的作業方式必需要能夠統合在一個視覺化的界面上才容易被管理、控制和監視，而商務流程管理系統便是提供此種功能的軟體系統。

在商務流程管理的領域，商務流程管理系統屬於資訊科技的運用層面，利用資訊科技實作商務流程管理所需的各項功能，不過，這並不表示商務流程管理系統可以等同商務流程管理，商務流程管理系統應被視為是商務流程管理的一種實作應用，是被包含在商務流程管理領域內的。

四、商務流程管理與工作流程管理

依據 Hollingsworth（1995）的定義，工作流程管理是以電腦化的軟體系統協助企業進行自動化的流程管理，著重以資訊技術建構自動化的流程管理系統，以減少人工作業，加快流程運行的速度，而商務流程管理是針對企業內各項由商務功能所組成的商務流程進行規劃、管理、監控及改善等作業，不僅將流程自動化，還包括企業整體流程的管控與調變。

工作流程管理與商務流程管理都是由活動所組成，工作流程管理著重於特定商務功能內的活動管理，而商務流程管理則是著重於企業內所有商務功能所組成的活動管理，兩者之間存在互補的關係。在實務的系統架構上，商務流程管理系統會包含工作流程管理的機制，如圖 2-3。

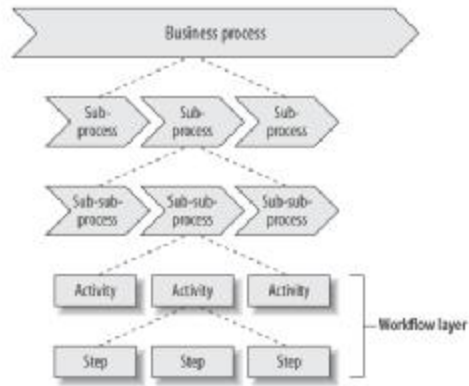


圖 2-3 商務流程與工作流程架構圖

資料來源: Josuttis,2007

五、商務流程建模 (Business Process Modeling)

商務流程模型 (Business Process Model) 是指對商務流程的具體描繪 (Bloomberg and Schmelzer, 2006)，內容包括活動執行的流程、控制和決策要點，觸發時機與條件等，包含流程執行時的所有內容，以圖型的方式展現，讓企業可以容易的瞭解商務流程的運行模式。

商務流程建模則是建構商務流程模型的方式和工具，經由運用適當的方法與工具建構商務流程模型後，商務流程模型可以被評估、分析、模擬，協助企業改進商務流程的運行。

六、商務流程執行語言

商務流程執行語言 (WS-BPEL, Web Service Business Process Execution Language) 最初是由 IBM、微軟和 BEA 三家軟體廠商

在整合 IBM 的 Web Services Flow Language (WSFL) 與微軟的 XLANG 技術後，所共同制定以網路服務為基礎的新一代自動化流程描述語言。

此語言在歷經變革後，已被制定成為一項技術標準，由 OASIS 組織進行編制，本研究所實作之版次為第二版，發佈於 2007 年 6 月，名稱由原來的 BPEL 更名為 WS-BPEL (Web Service Business Process Execution Language)。

WS-BPEL 是基於 XML 技術所制定的程式語言，用以描述服務所構成的商務流程和序列，使用此程式語言所制定的商務流程定義檔案可以被放置到商務管理系統的流程引擎上運行，經由使用者的觸發，流程引擎會依定義檔案內制定的流程，執行和呼叫相關服務。

WS-BPEL 與其它程式語言最大的不同在於 WS-BPEL 是針對服務導向架構下的商務流程管理需要所發展的流程描述語言，對於服務的處理具備較完善的機制，例如呼叫服務、處理回應要求等。

另外，WS-BPEL 是用 XML 標籤 (Tag) 定義的電腦語言，由元素 (Element) 構成程式的本體，利用組合和定義元素的方式便可以容易的將程式的撰寫工作予以視覺化，讓 WS-BPEL 的開發工具在程式邏輯撰寫的功能上，都能以拖拉和設定參數的方式完成程式設計工作，縮短流程調整的時間，降低流程變更的困難度。

WS-BPEL 已成為在商務流程管理系統中，用以定義商務流程的標準語言，企業若採用支援 WS-BPEL 的商務流程管理系統將會具有較高的可移植性，企業在選用商務管理系統時應將其列入考量的要素。

七、商務流程管理整合服務導向架構的益處

在服務導向架構下，商務功能所包含的商務邏輯被轉換成服務後，經由組裝的方式產生企業所需的軟體系統，以增加軟體系統在運用上的彈性，改善傳統架構在整合能力上的不足。

商務流程管理的目的是協助企業改善、管理與控制商務流程，使企業更容易達成營運目標，而商務流程的組成與管理的對象是商務功能，若商務功能在軟體系統裡能夠經由組合產生，則商務流程同時也就具備組合的能力，因為商務流程本身也可視為是一個更大的商務功能。

商務流程管理與服務導向的整合可以為商務流程的管理帶來服務導向架構所具備的可組合性與可再利用性的能力，對於需要統整不同商務功能的商務流程管理系統而言，服務導向架構能夠提供其所需的整合及彈性功能。

在整合後的架構下，商務流程的商務功能是由分解後的服務所構成，商務功能的程式實作與商務流程可分開成兩個不同的軟體元件 (Erl, 2005)，商務流程的管理只需專注於商務功能的組合，不需在意每一個構成商務功能的服務是如何被產生的，服務與服務之間的運行則交由服務導向架構系統來管理與控制，使得商務流程的管理變得單純。商務流程管理整合服務導向架構後，將可為企業帶來下列益處 (Wahli, 2007)：

1. 協助辨識與消除商務流程的冗餘及瓶頸。
2. 經由瞭解商務流程在企業營運上的影響，降低風險。
3. 將商務整合的邏輯從實作的程式碼中分解出來。
4. 透過對產業標準的制定，增加商務流程的可移植並降低維護成本。

5. 自動化商務流程部署，減少人工手動作業。
6. 可立即部署、施行新組建的商務規則和流程。
7. 視覺化的商務流程實際執行效能指標。
8. 可針對未來的流程進行改善。

八、企業資源規劃運用商務流程管理的效益

ERP 系統內含許多企業營運所需的商務流程 (Davenport,2000)，在 ERP 系統的導入專案中，商務流程是影響 ERP 系統成效的重要因素 (蔡良玉，2002)。許多的客製化程式被需要的原因是為配合企業的商務流程而產生，而客製化程式的數量將會影響到 ERP 系統導入專案的時程。

現行的 ERP 系統在商務流程的變更功能上，除應用工作流程序子系統協助簡化表單的遞送工作外，在商務流程的重組與增減上，仍需要經由新增客製化程式才能達成，因此也就增加 ERP 系統導入的風險 (蔡良玉，2002)。

在商務流程管理系統整合服務導向架構下，服務是商務流程的組件，商務流程的底層活動 (Low-Level) 即是服務 (Josuttis, 2007)，商務流程便可經由服務的組裝與重複使用來達成 (Behara, 2006)。因此，ERP 系統若是服務導向架構系統，在整合商務流程管理系統後，則可在不需新增 ERP 系統客製化程式的前提下，完成商務流程的變更作業。

第參章 研究方法

第一節 研究問題

從文獻探討得知，商務流程管理系統在整合服務導向架構之後，商務流程的變更可以經由重組具有商務功能的服務來達成，這使得軟體系統的商務流程和包含商務功能的程式邏輯得以被分離成兩個獨立運行的層（Tier），而這兩個層可以進一步劃分給二個各自獨立運行的系統來管理，一個是負責商務流程管理的系統，另一個是負責管理服務的系統，兩者之間經由網路服務的界面溝通，彼此的變更不觸及對方的程式邏輯，因而增加軟體系統在流程上的調變彈性。

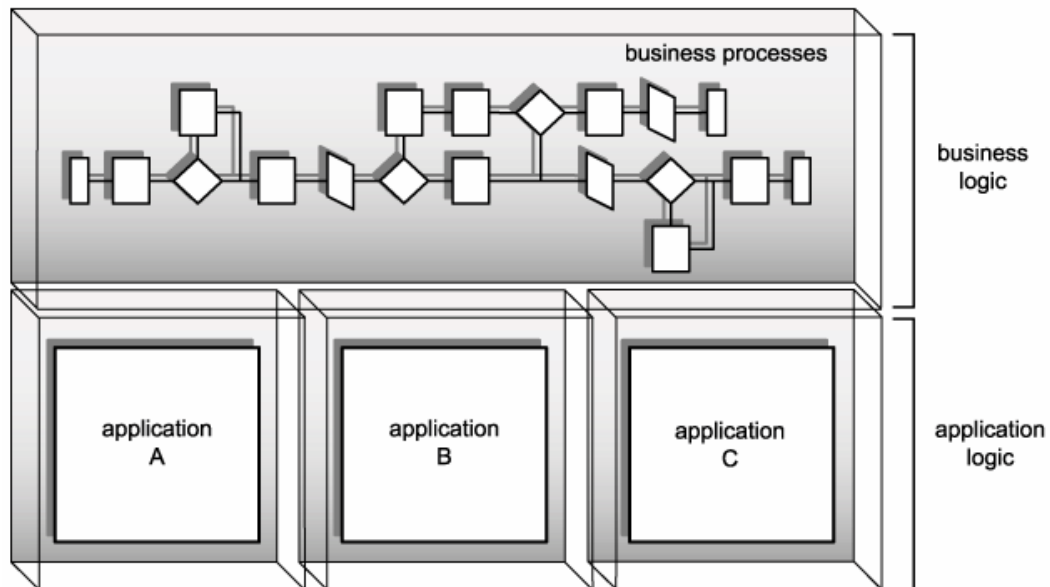


圖 3-1 商務邏輯與程式邏輯架構圖

資料來源: Erl,2005

市面上制式之 ERP 系統包含著企業運行的主要商務流程，ERP 系統所預先設計的商務流程與企業實際需要的商務流程之間所存在的差異程度將是影響企業導入 ERP 系統是否可以成功的關鍵因素。

在傳統的 ERP 系統下，商務流程與程式邏輯繫結在一起，商務流程的客製也就會涉及程式的修改，因而容易增加導入 ERP 的風險。

若企業採用以服務導向架構的 ERP 系統並整合商務流程管理系統，則商務流程可由服務組裝而成，商務流程的變更即是服務的重組，而使得 ERP 系統所預設的商務流程可以容易的經由組裝服務的方式達到企業所需要的商務流程，故可以不需開發 ERP 系統的客製化程式即可達成客製效果，此為本研究所要探討與實證的問題。

第二節 研究架構

本研究將選用一套商務流程管理系統及具服務導向架構的 ERP 系統，將其整合後，組裝個案所需的商務流程，以瞭解整合後的效益。

研究架構如下圖所示：

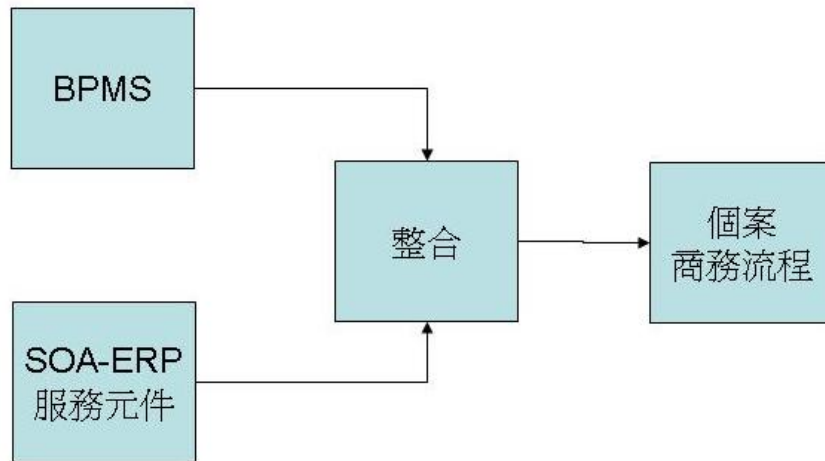


圖 3-2 研究架構圖
資料來源: 本研究整理

一、SOA-ERP 系統

NEO SOA-ERP 系統是寶盛公司歷經四年開發完成的 ERP 系統，NEO SOA-ERP 是一套百分之百的服務導向架構的 ERP 系統，由 3000 多個和 ERP 有關的服務構成。NEO 的商務報告（Business Reporting）系統，能整合不同 ERP 系統製作多公司合併財務報表。NEO 的財務會計系統、成本會計系統、和生產計劃（MPS/MRP）系統能整合企業現行的 ERP 系統，提升現行傳統 ERP 系統的能力。當然，NEO 也有配銷系統、採購系統、庫存系統、和製造外包系統。NEO 的每一個系統都能獨立安裝、運行並和其他系統整合，也能組合服務、彈性調整商務流程。

在架構上(如圖 3-3), NEO SOA-ERP 系統由服務組合而成並提供公開的網路服務給外部系統呼叫使用, 是一套以服務導向架構為基礎的軟體系統, 同時, NEO SOA-ERP 系統也能夠在多種平台和資料庫上運行, 具備高度彈性與整合能力, 是國內目前唯一實作服務導向架構的軟體系統, 故本研究運用此系統提供個案在商務流程上所需的各種商務服務。

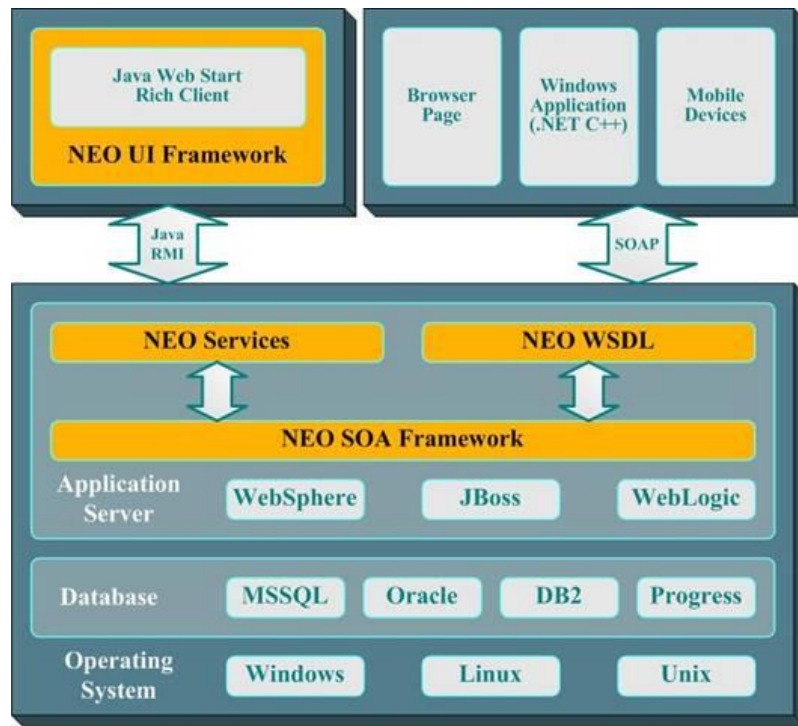


圖 3-3 NEO SOA-ERP 系統架構圖

資料來源: 寶盛公司提供

二、BPMS

近年來, BPMS 為各軟體廠商極力促銷的軟體產品, 推出的產品種類繁多。IBM 所推出的商務流程解決方案是市場上主要的產品之一, 其功能與架構完整, 產品內容包含商務流程的塑模、組

裝、部署和管理等功能（如圖 3-4），含蓋商務流程管理的計劃、執行、檢核及處置等循環，讓企業在一體作業下快速完成商務流程管理的應用系統。

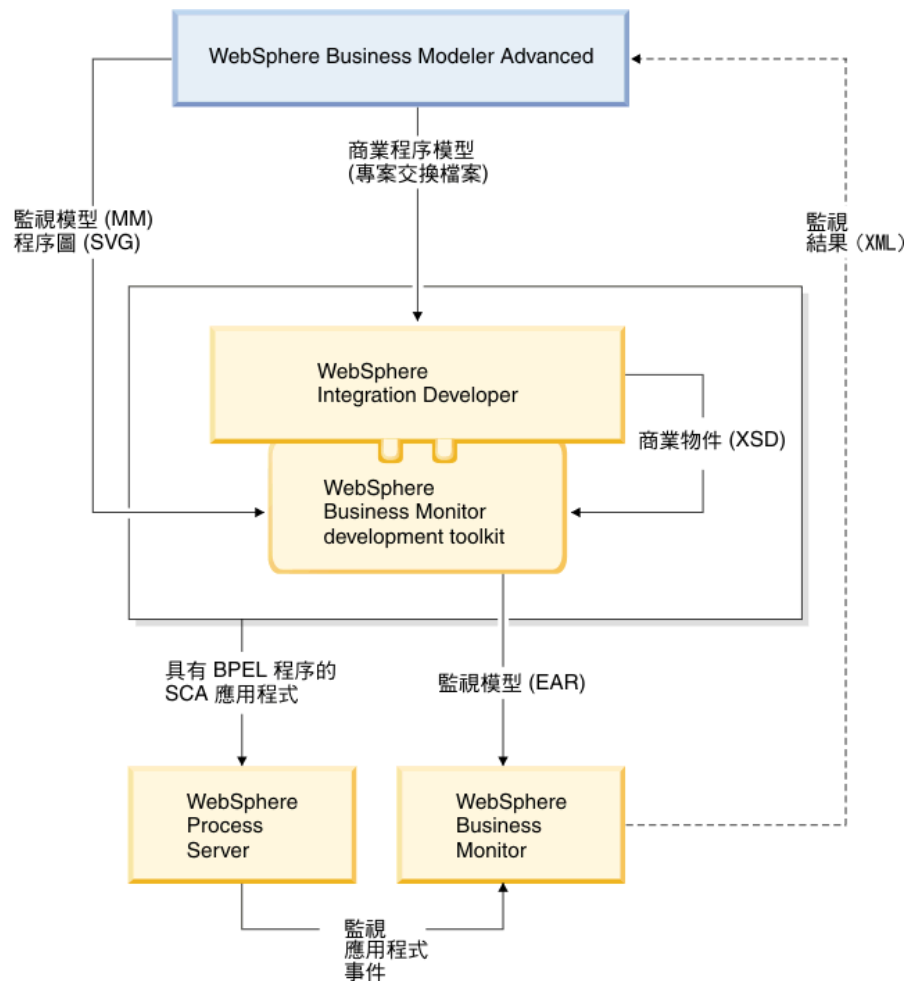


圖 3-4 IBM 商務流程管理解決方案架構圖

資料來源: IBM WebSphere Business Modeler Advanced 6.1.1 版產品概觀

寶盛公司的 NEO SOA-ERP 系統和 IBM 的商務流程管理系統在本研究進行期間正在推行一項合作計劃，故本研究配合此計劃之進行，採用 IBM 的商務流程管理解決方案的 WebSphere Business

Modeler Advanced、WebSphere Integration Developer 及 WebSphere Process Server 等產品作為本研究的開發與運行環境。

第三節 研究步驟

本研究採用個案研究法實證在服務導向架構與商務流程管理系統的整合環境下，企業的商務流程將可以不需要調整，而是經由重組服務的方式，在不需修改 ERP 系統程式的條件下，將 ERP 系統的商務流程調整成企業需要的流程。本研究將個案實際的商務流程需求在 SOA-ERP 系統上進行商務流程的變更，以瞭解實際運作的情況。

本研究將以個案研究為主體的架構下，分為下列三個步驟來進行：

1. 文獻探討

就服務導向架構與商務流程管理相關議題進行文獻探討。

2. 個案研究

針對個案在商務流程的需要與生成背景進行瞭解，從而分析其商務流程與 ERP 系統之差異。

3. 系統實證

將個案在商務流程上的需求實作在 SOA-ERP 與 BPM 整合的系統上，以驗證文獻探討的推論是否成立。



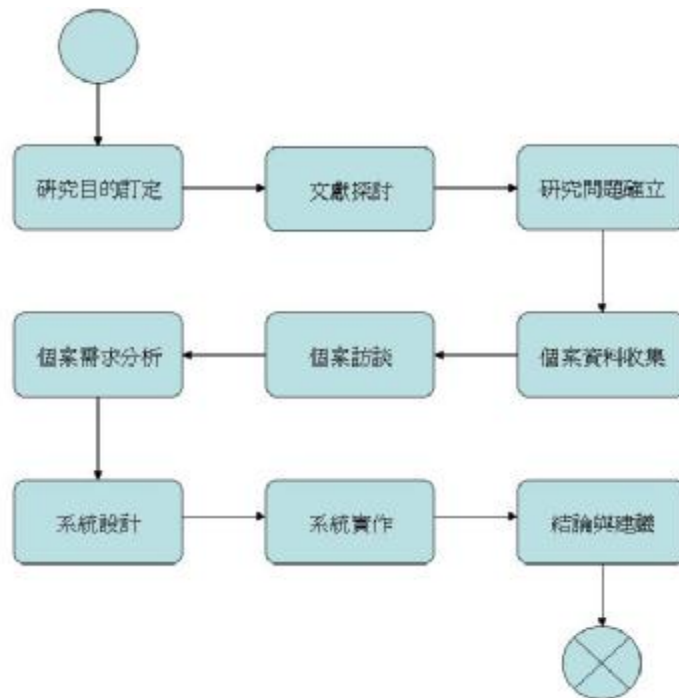


圖 3-5 研究流程圖

資料來源: 本研究整理

第四節 研究標的

個案現行的商務流程是否可以在不需要客製 SOA-ERP 程式的前提下，在 SOA-ERP 整合 BPMS 的系統上建置完成，為本研究所設定的研究標的。

如果個案可以在上述條件下完成建置，透過類型比對法推論，在 SOA-ERP 整合 BPMS 的系統環境下，企業將可因商務流程變更風險的降低而獲得利益。

第五節 個案選擇

Yin (1994) 指出，適合採用單一個案研究的類型有二種，第一，該個案是一個成熟理論的關鍵性個案；第二，該個案代表的是一種獨特或極端的個案。

本研究計劃以第二種類型的個案做為本研究挑選的考量因素，若此一獨特或極端的個案可以在本研究的設定環境下被實作完成，則可用以推論，其他一般性的個案亦是可行的。

第六節 資料收集方法

本研究為對個案進行瞭解，將採用下列方式對個案進行資料收集的作業：

1. 深度訪談法

經由與個案主的深度訪談，瞭解個案在現行的商務流程上的需求、考量與運作情形，以作為系統實作的分析、設計與建置依據。

2. 直接觀察法

為瞭解其商務流程的實際運作情形，而到個案現場實地觀察。

3. 文獻收集

透過文獻、網站等二手資訊的收集，增加對個案背景資料的瞭解，以提高需求與系統分析的準確度。

第肆章 個案研究

第一節 個案分析

一、個案介紹

個案全名為「新傳興業股份有限公司」，是一家辦公椅和電動窗簾的製造商，同時還代理、銷售國內外高價辦公椅及辦公桌等產品，依客戶需要，與自製的產品搭配銷售。

個案成立於民國 79 年，原來的母公司為新瀚實業股份有限公司，主要營業項目為石膏板隔間與輕鋼架天花板，並轉型生產台灣第一套辦公室用屏風系統，深獲市場好評，業績成長迅速，公司高層想留住優秀員工，所以採用內部創業方式，另外成立一家生產辦公椅的公司，這就是新傳興業股份有限公司成立的由來。

個案的資本額 800 萬，年平均營業額約 3500 萬，員工數 13 人，依照經濟部中小企業處頒布的「中小企業發展條例」定義，個案可歸類於製造業的中小企業。

個案的工廠座落於泰山鄉全興路上，位於中山高速公路五股交流道出口附近，是傢俱產業生產與銷售群聚的區域，因此，個案除自營銷售外，也出貨給附近的店家經銷。另外，若客戶的訂單有個案無法提供的產品時，個案也會委由附近合作的廠商製造或採購。

傢俱製造業是成熟產業，競爭激烈，各家廠商必須為自己創造出獨特的利基才能夠永續經營。個案為提高競爭力，推動商務

流程的再造計劃，於前年再造成功。個案善用小企業及本身的特質，有效的縮短和變更商務流程，大幅減少不必要的流程和人工作業，增加營運績效。

在完成商務流程再造後，個案便計劃導入 ERP 系統，將生產與財務資訊電子化，以提供較即時的決策資訊。但是，由於個案是根據自身特殊的營運模式所改造而成的商務流程，和傳統製造業差異相當的大，市面上並無適合的 ERP 產品，只能尋找商務流程相近的套裝系統於購入後客製或是委外重新開發才能符合其需要。

然而，從本研究的文獻探討中得知，客製化程式的數量將會影響到 ERP 系統導入專案的時程，客製程式的數量越多，導入的時間就會拉長，所需耗用的人力與費用也就愈高，導入失敗的風險便相對提高，而重新設計開發的風險就更高。

個案希望能在不變更商務流程的條件下完成 ERP 系統的導入專案，剛好符合本研究要尋找的具備不適用「最佳實務」和必需客製 ERP 系統的條件，故選用作為本研究的個案。

二、營運特質

個案流程再造之後的商務流程是基於個案本身在營運上的多項特質而調變的，這些特質是個案在多年的營運過程中，經由不斷的調整與調適所蛻變而成的。若沒有這些特質，個案的商務流程便不具備獨特性，也就無法成為營運的利基。

本研究經由訪談與資料收集，將個案的營運特質進行瞭解與分析，以明白造成個案現行商務流程的原因，其特質與分析說明如下：

1. 接單式生產

個案是採用接單生產的模式，產品完成後直接出貨給客戶，除少數樣品外，無成品庫存，每一筆訂單的生產流程分明且訂單之間的相依度低。

2. 單階的材料表（BOM）

個案的成品是經由組裝各項零件而成，而零件都屬完整、單一的元件，沒有需要使用子元件，所以各項產品的材料表均為單階，材料的使用量可直接由出貨的數量來計算。

3. 低材料庫存

辦公椅的組成布面花色與材質種類繁多，但是每一筆訂單所需的使用量少，除了一、二種耗用量較高的布面有少量的庫存外，其他的布面材料為零庫存，依訂單需要才採購所需的使用量。

4. 生產期短

每一筆訂單的平均生產期間約 3~5 天即可完成，因此就算在出貨才計算庫存耗用資料，其資料的即時性仍在個案所能接受的範圍內。

5. 開放式倉庫管理

由於材料成本不高，價格變動亦較低，再加上存量也不多，故對於材料的出入庫不作管制，僅每月盤點一次，供校正財務資料，此為個案導入 ERP 系統後所想要改善的問題之一。

6. 人員精簡

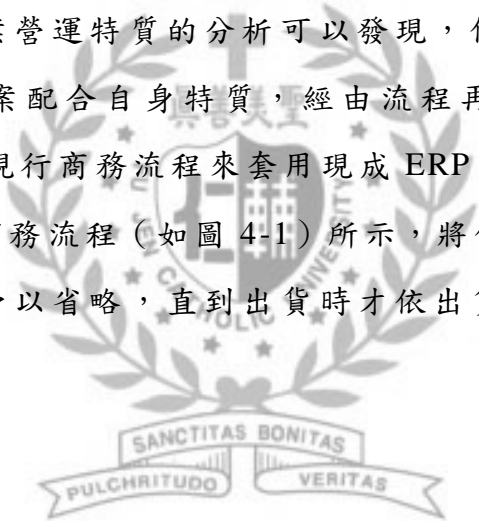
個案員工數共 13 人，而行政人員除個案業主外僅有一

人，其餘皆為生產製造與銷售人員，故個案希望減少不需要的人工作業，尤其是為讓資訊系統能夠運行和計算而進行的電腦資料輸入作業是個案最想免除的人工作業。

第三節 需求分析

近年來企業的商務流程再造大多是為配合導入 ERP 系統而推行，以 ERP 系統預設的「最佳實務」商務流程作為典範，將企業自身的商務流程調整成可以適用於 ERP 系統的流程，藉由最少量的客製化，降低導入風險。不過，此種方式並不適用於本研究的個案，經由對個案營運特質的分析可以發現，個案目前所運行的商務流程已是個案配合自身特質，經由流程再造後的「最佳實務」，不適合更改現行商務流程來套用現成 ERP 系統的商務流程。

個案現行的商務流程（如圖 4-1）所示，將傳統製造業有關工令與領料的作業予以省略，直到出貨時才依出貨的數量計算材料的用量。



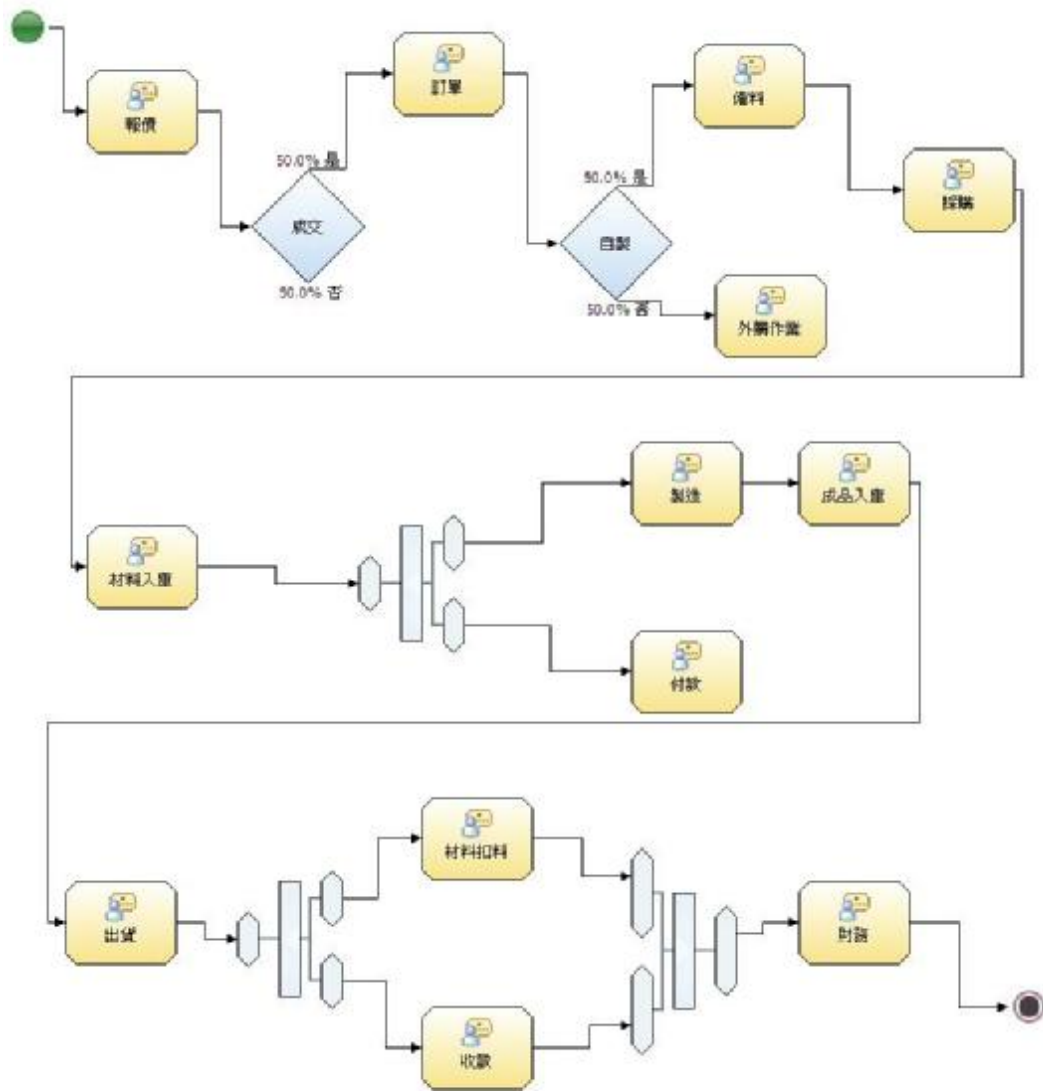


圖 4-1 個案商務流程圖

資料來源: 本研究整理

個案之所以採用此種商務流程，其主要目的在於減少不必要的人工資料處理作業成本，並加快作業流程。個案由於材料表為單階，又生產週期短，員工也不多，生產程序也不複雜，若只為

使 ERP 系統上的材料庫存能夠正確與實際相符而必需增加工令、領料、成品出入庫等作業，對個案而言並不具效益。

個案所需的商務流程主要是希望能夠將傳統經由工令、領料等流程而發生的材料庫存異動程序予以省略，然後延遲到實際出貨時，配合財務作業需要，在認列成本時，才一併增加成品及抵減材料庫存數量，如此，個案將可節省大量的人工資料輸入成本，同時，可真正自 ERP 系統移除其實務上不存在的商務流程。

在材料採購的請購作業上，因為員工人數少，組織階層扁平及 BOM 單純等原因，再加上員工對公司的向心力強，為免除不必要的人工作業，因此無設置請購作業，所以在 ERP 系統上也不需要執行請購作業的相關功能。

除請購作業外，個案的報價作業因報價內容與銷售訂單內容完全相同，而且未成交的報價單對個案並無保存的價值，所以個案的報價作業不需納入 ERP 系統內，將繼續採用人工報價。

綜合上述的需求，本研究將運用商務流程管理系統對 SOA-ERP 系統的商務流程進行調整，經由重組 SOA-ERP 系統的服務達到個案需要的商務流程。

三、商務流程模型

傳統製造業的商務流程從報價作業開始，將歷經銷售訂單作業、請購作業、採購作業、進貨作業、製造作業、及出貨作業等流程，商務流程模型如圖 4-2 所示。

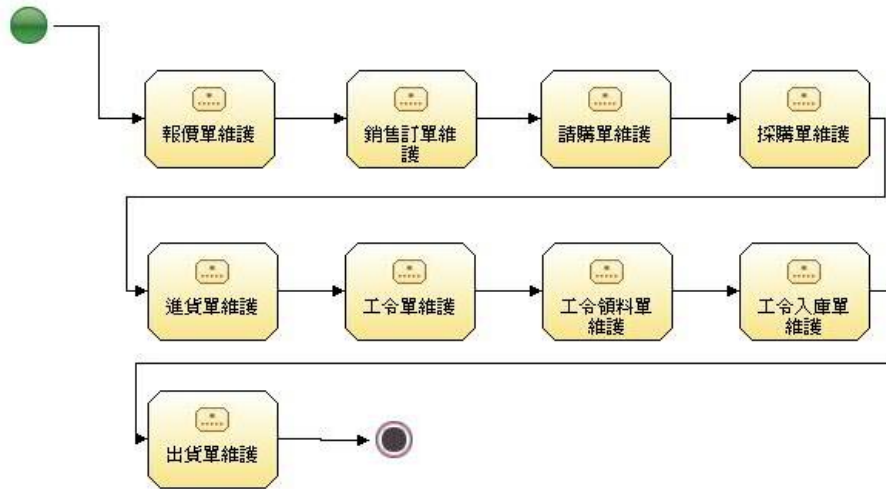


圖 4-2 傳統製造業商務流程圖

資料來源: 本研究整理

工令、領料及成品入庫等作業在製造業的商務流程裡被歸類為製造作業，而此部份在個案的商務流程中並不會發生，除此之外，報價作業將維持人工作業，而請購作業也要予以省略，所以個案的商務流程將會簡化成由訂單作業、採購作業、進貨作業、及出貨作業所組成，其商務流程模型如圖 4-3 所示。

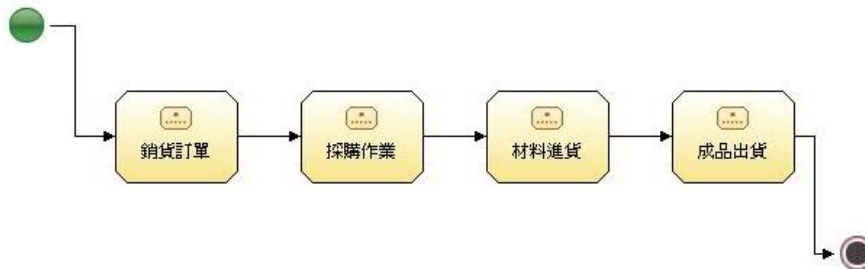


圖 4-3 個案的商務流程圖

資料來源: 本研究整理

第二節 系統設計

本研究將經由重組 NEO SOA-ERP 系統的部分商務流程來達到個案所需的商務流程，NEO SOA-ERP 預設的商務流程程序如圖 4-4 所示。

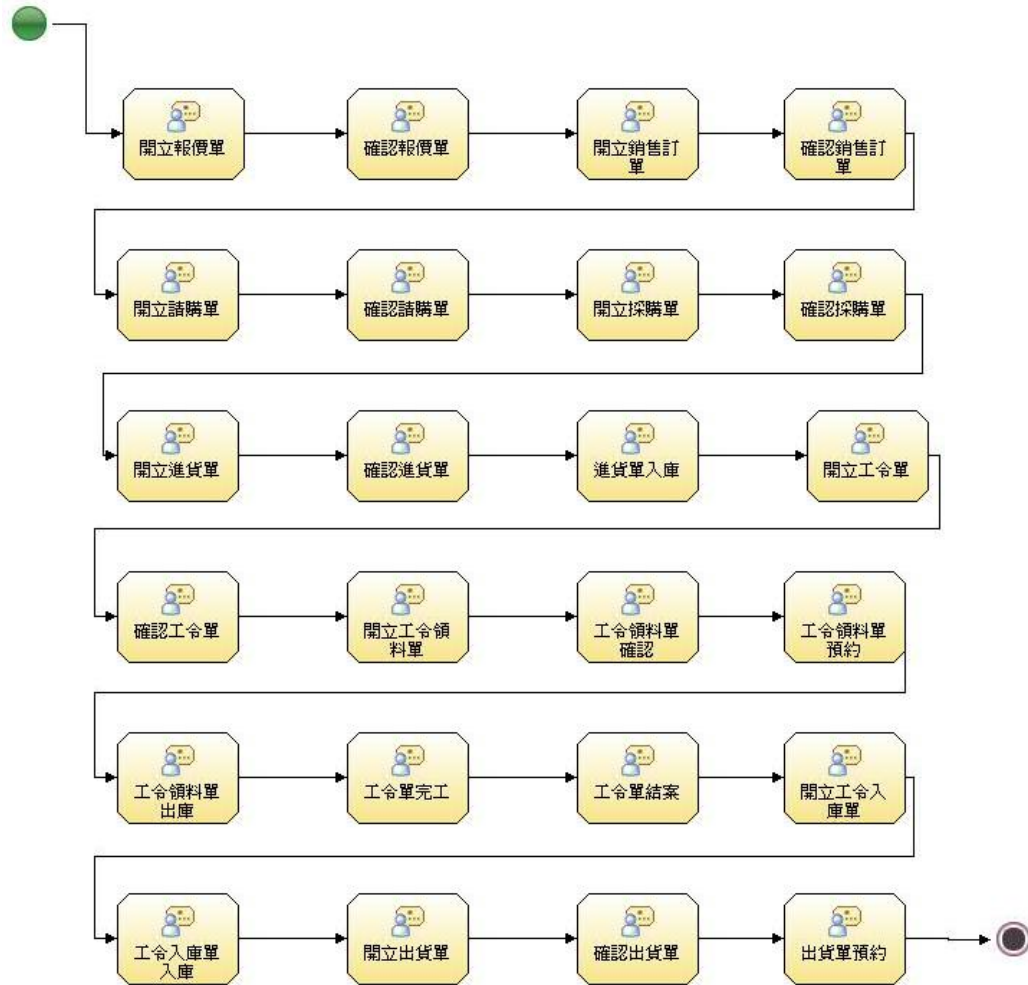


圖 4-4 NEO SOA-ERP 系統預設商務流程圖

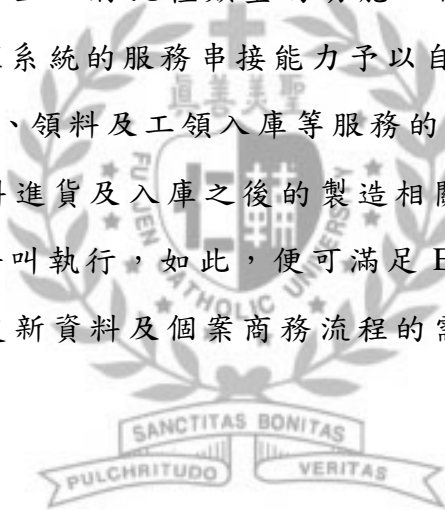
資料來源: 本研究整理

在 NEO SOA-ERP 預設的流程裡，有關材料庫存及出、入庫的資料更新必須經由工令、領料等相關服務的操作才能達成，也就是，若要能夠在出貨時扣減材料庫存，則一定要經由工令及領料

等作業來對材料庫存進行操作。同時，成品庫存的入、出庫作業也和材料庫存一樣，必須以工令作業為基礎，經由工令入庫作業來增加成品庫存數量。

上述的工令、領料和工令入庫等作業在個案的流程裡並不存在，故不須因為 ERP 系統的運算需要而強迫使用者刻意的進行操作，這也是個案在使用其它 ERP 系統所遭遇的難題。這些多出來的操作功能與實務作業不同，常常造成操作人員的困擾，降低操作人員登打資料的意願，而造成登打資料的不完整。基於這個考量，本研究在設計上，將此種類型的功能，利用服務無介面的特性與商務流程管理系統的服務串接能力予以自動化。

另外，將工令、領料及工領入庫等服務的呼叫順序往後調整，把原本應該在材料進貨及入庫之後的製造相關服務移到使用者執行出貨作業時才呼叫執行，如此，便可滿足 ERP 系統需要有事件驅動相關服務以更新資料及個案商務流程的需求。調整後之商務流程模型如下：



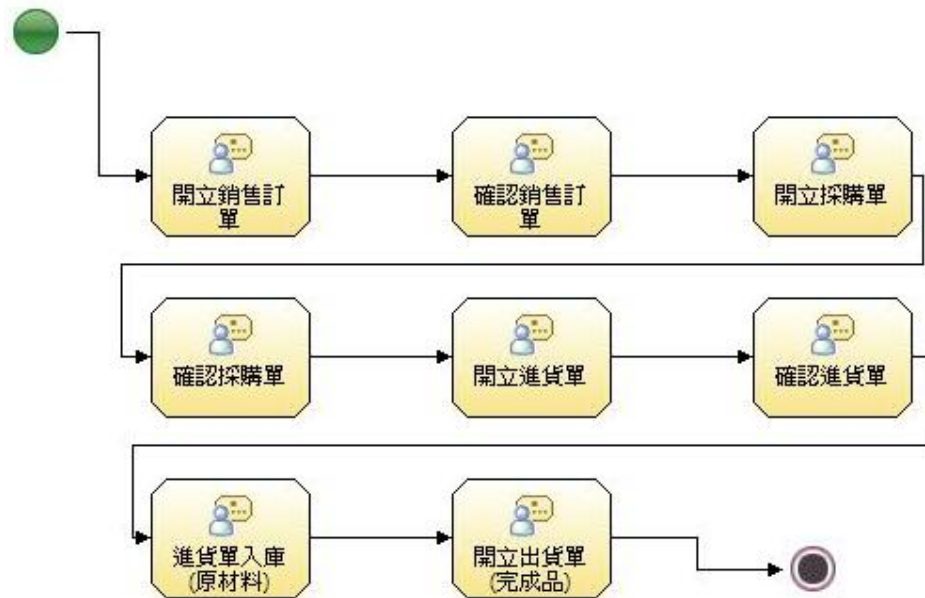


圖 4-5 個案的實作商務流程圖

資料來源: 本研究整理

本研究設計的做法是將工令、領料及工令入庫等一連串的服務呼叫合併到使用者實際需要執行出貨作業時才自動呼叫這些服務來完成材料與成品庫存的異動，原本各項作業所需操作的畫面將不會出現，對使用者而言，材料進貨、入庫之後便可執行成品出貨作業，而出貨作業完成後，材料庫存及成品庫存也會跟著更新。

調變後的出貨作業功能將會設計成一個單獨的新服務，將原本 NEO SOA-ERP 系統所提供的工令、領料、成品入庫及出貨等服務封裝起來，使用者在操作出貨作業時，系統所呼叫的新出貨服務對使用者而言，和呼叫 NEO SOA-ERP 系統內建的出貨服務是沒有

差別的。

這個新的出貨服務將由商務流程管理系統來實作開發，而不在 NEO SOA-ERP 內進行。實際上，由於 NEO SOA-ERP 系統是一個服務導向架構的系統，當然可以在 NEO SOA-ERP 系統內經由重組服務而產生新的服務，不過這樣的作法便失去服務導向架構元件化的特色及使用商務流程管理系統的價值。ERP 和 BPM 是不同的專業，理應由不同廠商提供，再組合成軟體產品，正如所有硬體產品都是專業分工一樣。NEO 既然是 ERP 專業，就不應該在 ERP 系統中發展商務流程管理功能。商務流程管理系統可以快速、容易的經由視覺化圖形界面重組與測試新服務，而每一個新建的服務均符合服務元件架構與 WS-BPEL 標準，使得新服務在產生之後便可被直接用來與其他服務組合產生另一個新的服務。同時，此一新服務將不受限於 NEO SOA-ERP 的服務，因為此一新服務的內部僅包含呼叫服務的流程與資料交換，若日後不使用 NEO SOA-ERP 系統時，在其他的 SOA-ERP 系統上，只要調整此服務呼叫新系統提供的服務名稱與順序後，即可繼續使用此出貨服務而不需重新撰寫或客製 ERP 系統。

在系統架構的設計方面，雖然商務流程管理系統可將個案所需的完整流程全部建置於其上，達到完全客製的結果，但是，在實務上，個案並不在意系統架構的方式，其關注的重點在於可否達成目標的商務流程。而本研究的重點在於商務流程管理系統對於 SOA-ERP 系統的效益，故由商務流程管理系統創建個案所需的整個商務流程的實質效益不大，故僅就個案流程變異的部分改由商務流程管理系統製作所需的新服務，其他流程仍維持由使用者自行進入 NEO SOA-ERP 系統操作介面，以縮短客製時程並符合期

望。

在此架構之下，個案的訂單作業、採購作業與進貨作業不需要客製與調整，故可直接採用 NEO SOA-ERP 系統現有的功能與介面，而出貨作業則需使用商務流程管理系統所提供的介面進行操作，達成於兩個不同介面操作同一系統的服務與資料，如此的設計方式，同時也是為了呈現服務導向架構的整合能力。

技術上，若不要採用兩個分開的介面的話，也可在 NEO SOA-ERP 系統的前端加入一個客製介面來呼叫新組裝的出貨服務，一樣可達到預期的效果，但是為了實證 SOA-ERP 系統與商務流程管理系統整合下，可以在不需要客製 ERP 系統的條件下完成商務流程的改變，故本研究不採用此種方案。

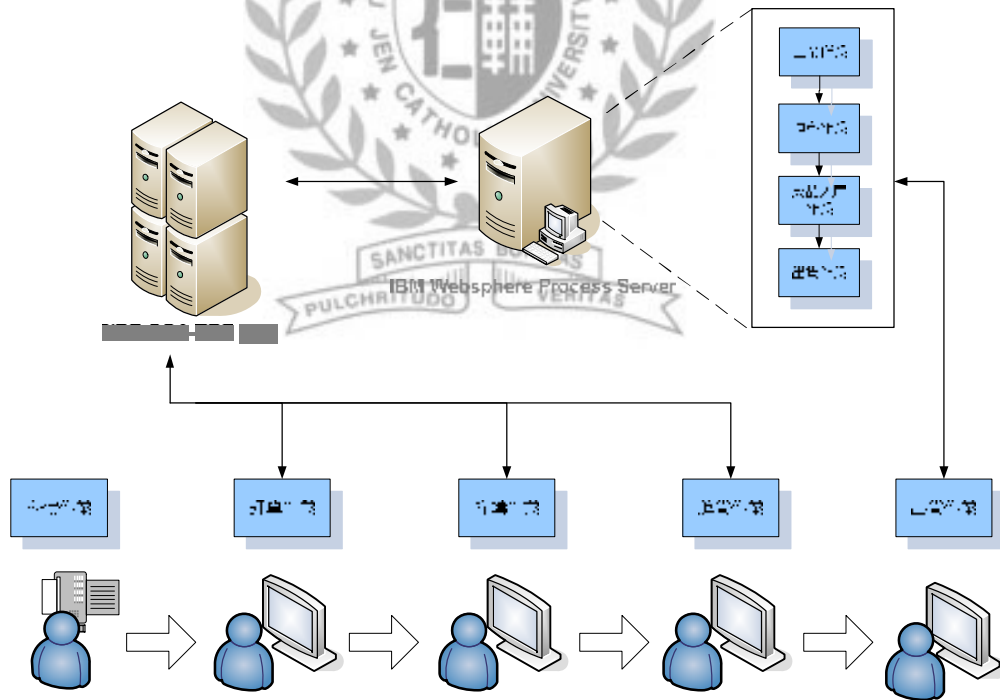


圖 4-6 完成後的系統架構圖

資料來源：本研究整理

NEO SOA-ERP 系統報價與請購作業在預設的流程上並不是必要功能，使用者不操作該功能即不會啟用，因此在 NEO SOA-ERP 系統的部分，不進行任何的修改和客製。

第三節系統實作

本研究的實作開發流程分為下列步驟進行：

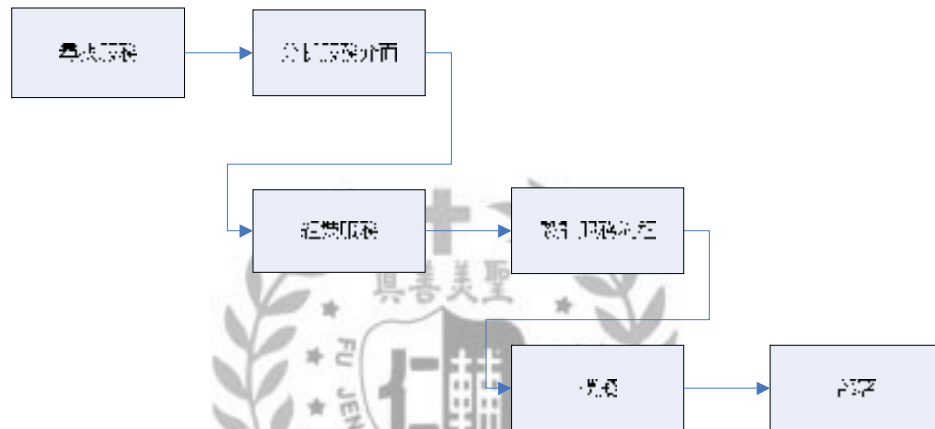


圖 4-7 本研究的實作流程圖

資料來源: 本研究整理

一、尋找服務

在個案需求分析與商務流程模型完成之後，接下便是要組裝新的出貨服務。首先，在組裝新服務之前，須從 NEO SOA-ERP 系統中整理出需要的服務，在 NEO SOA-ERP 系統預設的商務流程中，從工令到出貨之間的作業流程，其預設的操作步驟如下：

表 4-1 NEO SOA-ERP 系統從工令到出貨的相關功能列表

序號	功能別	表單別	動作
1	工令單維護	工令單	新增
2	工令單維護	工令單	確認

3	工令領料單維護	工令領料單	新增
4	工令領料單維護	工令領料單	確認
5	工令領料單維護	工令領料單	預約
6	工令領料單維護	工令領料單	出庫
7	工令單維護	工令單	完工
8	工令單維護	工令單	結案
9	工令入庫單維護	工令入庫單	新增
10	工令入庫單維護	工令入庫單	入庫
11	出貨單維護	出貨單	新增
12	出貨單維護	出貨單	確認
13	出貨單維護	出貨單	預約
14	出貨單維護	出貨單	出庫

整理出所需的功能後，再依照功能尋找相對應的服務：

表 4-2 NEO SOA-ERP 系統從工令到出貨的相關服務列表

序號	表單別	動作	服務名稱
1	工令單	新增	UC_MFG_WORKORDER. CRTWORKORDER
2	工令單	確認	UC_MFG_WORKORDER. CONFIRMBILL
3	工令領料單	新增	UC_MFG_WORKPICK. CRTWORKPICK
4	工令領料單	確認	UC_MFG_WORKPICK. CONFIRMBILL
5	工令領料單	預約	UC_MFG_WORKPICK. RESERVATIONBILL
6	工令領料單	出庫	UC_MFG_WORKPICK. INVENTORYOUTBILL
7	工令單	完工	UC_MFG_WORKORDER. WORKENDBILL
8	工令單	結案	UC_MFG_WORKORDER. CLOSEBILL

9	工令入庫單	新增	UC_MFG_WORKINVENTORYIN. CRTWORKINVENTORYINBATCH
10	工令入庫單	入庫	UC_MFG_WORKINVENTORYIN. INVENTWIYIN
11	出貨單	新增	UC_DIS_SHIPPINGORDER. CRTSHIPPINGORDER
12	出貨單	確認	UC_DIS_SHIPPINGORDER. CONFIRMBILL
13	出貨單	預約	UC_DIS_SHIPPINGORDER. INVENTORYAPPOINTMENT
14	出貨單	出庫	UC_DIS_SHIPPINGORDER. INVENTORYOUT

在 NEO SOA-ERP 系統裡，表單和表單之間的串接可經由批次新增的功能新增各種表單，減少使用者重複輸入資料的動作，因此，可得知 NEO SOA-ERP 系統有提供批次新增表單的服務，為減少服務串接的資料交換量，本研究將改採用各類批次新增的服務取代原本的新增服務，更新後的服務列表如下：

表 4-3 NEO SOA-ERP 系統從工令到出貨的服務列表

序號	表單別	動作	服務名稱
1	工令單	<u>批次</u> 新增	UC_MFG_WORKORDER. CRTWORKORDERBATCH
2	工令單	確認	UC_MFG_WORKORDER. CONFIRMBILL
3	工令領料單	<u>批次</u> 新增	UC_MFG_WORKPICK. CRTWORKPICKBATCH
4	工令領料單	確認	UC_MFG_WORKPICK. CONFIRMBILL
5	工令領料單	預約	UC_MFG_WORKPICK. RESERVATIONBILL

6	工令領料單	出庫	UC_MFG_WORKPICK. INVENTORYOUTBILL
7	工令單	完工	UC_MFG_WORKORDER. WORKENDBILL
8	工令單	結案	UC_MFG_WORKORDER. CLOSEBILL
9	工令入庫單	新增	UC_MFG_WORKINVENTORYIN. CRTWORKINVENTORYINBATCH
10	工令入庫單	入庫	UC_MFG_WORKINVENTORYIN. INVENTWIYIN
11	出貨單	批次 新增	UC_DIS_SHIPPINGORDER. CRTSHIPPINGORDERBATCH
12	出貨單	確認	UC_DIS_SHIPPINGORDER. CONFIRMBILL
13	出貨單	預約	UC_DIS_SHIPPINGORDER. INVENTORYAPPOINTMENT
14	出貨單	出庫	UC_DIS_SHIPPINGORDER. INVENTORYOUT

除上列的服務之外，由於工令作業的資料來自於銷售訂單，因此還必需使用到查詢銷售訂單相關的服務，以取得工令等服務所需的參數資料。

表 4-4 NEO SOA-ERP 系統的銷售訂單查詢服務

序號	表單別	動作	服務名稱
1	銷售訂單	查詢	DIS_SALESORDER. QRYSALESORDER
2	銷售訂單	查詢 明細	DIS_SALESORDER. QRYSALESORDERBYSYSID

上列的服務的功能均是為達到出貨時自動扣除材料的目的所必需呼叫的服務，對於 NEO SOA-ERP 系統內為預設的，但不是必

要的功能，例如各種表單的查詢與異動等相關服務，因在本研究中不會使用到，故沒有列入表中。

二、分析服務介面

彙整出服務項目後，便需要針對每一個服務的輸入和輸出介面進行瞭解，以得知呼叫服務時，必需傳入的資料項目，並分析服務所需輸入的資料項目來源是否足夠，或者在呼叫該服務之前是否有其它應該要先呼叫的服務，同時也要知道每一個服務回應的資料項目，這樣才能知道從那一個服務可以取得所需的資料。

NEO SOA-ERP 系統並未針對服務另外提供專門的說明文件，而是提供 JavaDoc 供查詢，故需要對 NEO SOA-ERP 系統的物件模型有一些概念，否則不容易瞭解類別與服務之間的差異。不過，NEO SOA-ERP 的分類是以實際作業來區分，例如製造、庫存及財務等，用操作介面的功能為參考，尋找服務時，至少有一個依循的方向。

經過對每一個服務的介面進行分析後發現，NEO SOA-ERP 系統在呼叫成品入庫與出貨等服務之前，尚需呼叫其它服務才能順利完成成品入庫與出貨的作業，因此多出幾個新的服務，表列如下：

表 4-5 NEO SOA-ERP 系統核心模組的服入庫與出貨服務

序號	表單別	動作	服務名稱
1	工令入庫單	入庫	UC_CORE_ONHANDINVENTORY. INVENTORYIN
2	工令入庫單	出庫	UC_CORE_ONHANDINVENTORY. INVENTORYOUT

3	出貨單	自動配置 預約庫存	UC_CORE_RESERVATIONINVENTORY. RESERVATIONTINVENTORY
4	出貨單	新增預約 庫存	UC_CORE_RESERVATIONINVENTORY. CRTRESERVATIONTINVENTORY

綜合上一個步驟所彙集的項目和上述表格的項目，本研究所需使用的 NEO SOA-ERP 系統服務清單如下（依表單別排列）。

表 4-6 從工令到出貨的所有服務列表

序號	表單別	動作	服務名稱
1	工令單	新增	UC_MFG_WORKORDER. CRTWORKORDERBATCH
2	工令單	確認	UC_MFG_WORKORDER. CONFIRMBILL
3	工令單	完工	UC_MFG_WORKORDER. WORKENDBILL
4	工令單	結案	UC_MFG_WORKORDER. CLOSEBILL
5	工令領料單	新增	UC_MFG_WORKPICK. CRTWORKPICKBATCH
6	工令領料單	確認	UC_MFG_WORKPICK. CONFIRMBILL
7	工令領料單	預約	UC_MFG_WORKPICK. RESERVATIONBILL
8	工令領料單	出庫	UC_MFG_WORKPICK. INVENTORYOUTBILL
9	工令入庫單	新增	UC_MFG_WORKKINVENTORYIN. CRTWORKKINVENTORYINBATCH
10	工令入庫單	入庫	UC_MFG_WORKKINVENTORYIN. INVENTWIYIN
12	工令入庫單	入庫	UC_CORE_ONHANDINVENTORY. INVENTORYIN
13	工令入庫單	出庫	UC_CORE_ONHANDINVENTORY.

			INVENTORYOUT
14	出貨單	新增	UC_DIS_SHIPPINGORDER. CRTSHIPPINGORDERBATCH
15	出貨單	確認	UC_DIS_SHIPPINGORDER. CONFIRMBILL
16	出貨單	預約	UC_DIS_SHIPPINGORDER. INVENTORYAPPOINTMENT
17	出貨單	自動配置 預約庫存	UC_CORE_RESERVATIONINVENTORY. RESERVATIONTINVENTORY
18	出貨單	新增預約 庫存	UC_CORE_RESERVATIONINVENTORY. CRTRESERVATIONTINVENTORY
19	出貨單	出庫	UC_DIS_SHIPPINGORDER. INVENTORYOUT
20	銷售訂單	查詢	DIS_SALESORDER. QRYSALESORDER
21	銷售訂單	查詢 明細	DIS_SALESORDER. QRYSALESORDERBYSYSID

最後，將所有使用到的服務依流程繪製出來（如圖 4-8），以便做為後續重組服務時的參考。



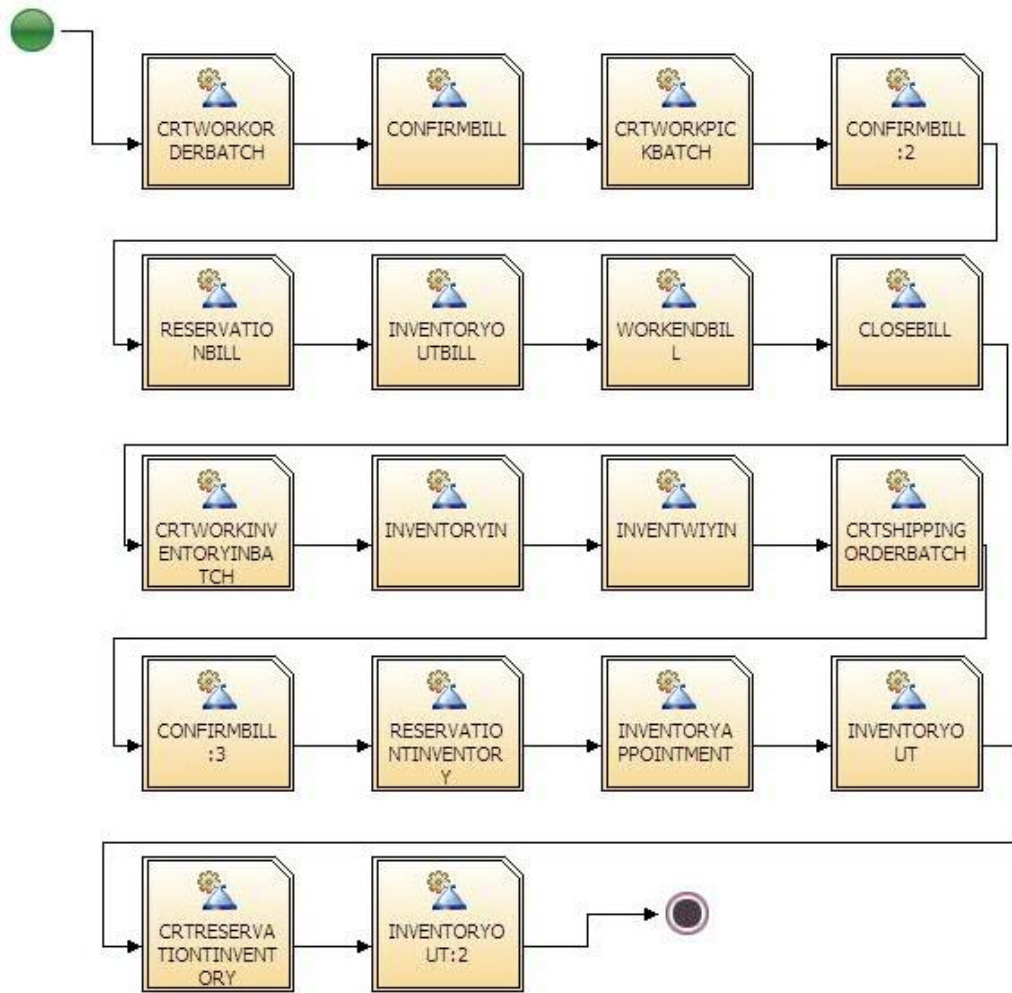


圖 4-8 NEO SOA-ERP 系統從工令到出貨的服務呼叫順序圖

資料來源: 本研究整理

三、組裝服務

本研究採用 WebSphere Integration Developer(以下簡稱 WID)作為組裝服務的工具，要在 WebSphere Integration Developer 中使用 NEO SOA-ERP 系統的服務，必須取得上述服務的 WSDL 檔，並將其匯入到 WID。

服務均需要 WSDL 檔的描述來取得呼叫服務的資訊，其中包括服務名稱、輸入和輸出資訊、通訊方式及服務所在的位址資訊，

雖然服務導向架構的系統不一定要使用網路服務作為通訊的介面與工具，而 NEO SOA-ERP 系統和 IBM 的 WebSphere Integration Developer 也支援其它類型的通訊方式，但本研究的實證結果希望可以套用於普遍的個案環境中，故採用具公開標準且通用的網路服務作為服務之間的通訊介面。

在 WID 中有二種專案模式，一種是 Library 專案，存放其他專案會共用到的元件，另一種則是用以開發服務的 Module 專案。本研究先開立一個新的 Library 專案將 WSDL 檔匯入到此專案中（如圖 4-9），以供後續其他專案使用。同時，若 WSDL 檔需要更新時，只需更新此專案即可，其他專案將會自動更新。



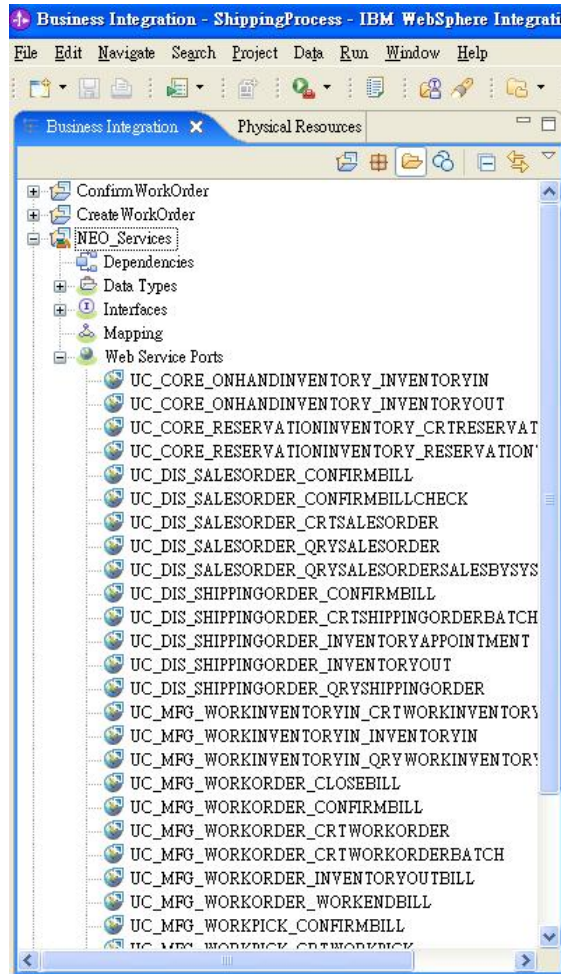


圖 4-9 WSDL 檔匯入獨立的程式庫專案

WID 的開發架構遵從服務元件架構的標準，所以在 WID 開發的每一個專案即是一個模組，而模組在概念上如同一個更大型的服務，裡面由許多的服務組裝而成，因此模組的本身也會有輸出與輸入的介面，提供為其它模組或服務呼叫使用。

本研究所要組裝的出貨服務也可以經由多個模組共同組成，例如，將工令、領料、成品入庫及出貨等服務，依表單別分成多個模組來開發。此種做法的好處是在開發時期可以切割成較細的單元進行開發與測試，較易分工合作，並且於系統上線後，若是

發生以表單為基礎的流程調整時，只需重排各模組即可，此種方法可以帶來較大的彈性，但是此種做法必需在模組與模組之間重新定義輸出、輸入介面，較為費時，本研究基於時程考量和個案需求的單純性，並未採用此法，而是將所有的服務都放在一個模組內，因此只需定義一個新的出貨服務介面即可，較為省時，開發方式也比較單純。

所有的 WSDL 檔匯入 WID 後，便可新增專案組裝新的出貨服務，WID 有兩種模組的專案類型，一種是一般的模組 (Module)，另一種是調解模組 (Mediation Module)，適合專門用來排列服務順序的模組，第二種類型應是本研究所要採行的做法，但是由於 NEO SOA-ERP 系統的服務之間需要大量的資料交換及服務介面之間的差異性大，無法直接運用「對映 (Mapping)」功能，故捨棄調解模組而改用一般的模組專案。

專案新增完成後便可開始組裝服務，服務的組裝工作是在組合圖 (Assembly Diagram) 中操作，將使用到的服務拉到圖中並定義匯入的服務與新服務的關係後，初步組裝的作業即算完成。組裝完成後的圖形如圖 4-10 所示，圖中左方的大型長方塊即是本研究所設計的出貨服務元件，命名為 ShippingProcess，而圖中右側的橫向長方條即是匯入被用來組裝的 NEO SOA-ERP 系統的服務。每一個服務與出貨服務之間的都透過一個定義在 WSDL 檔的介面來連接。

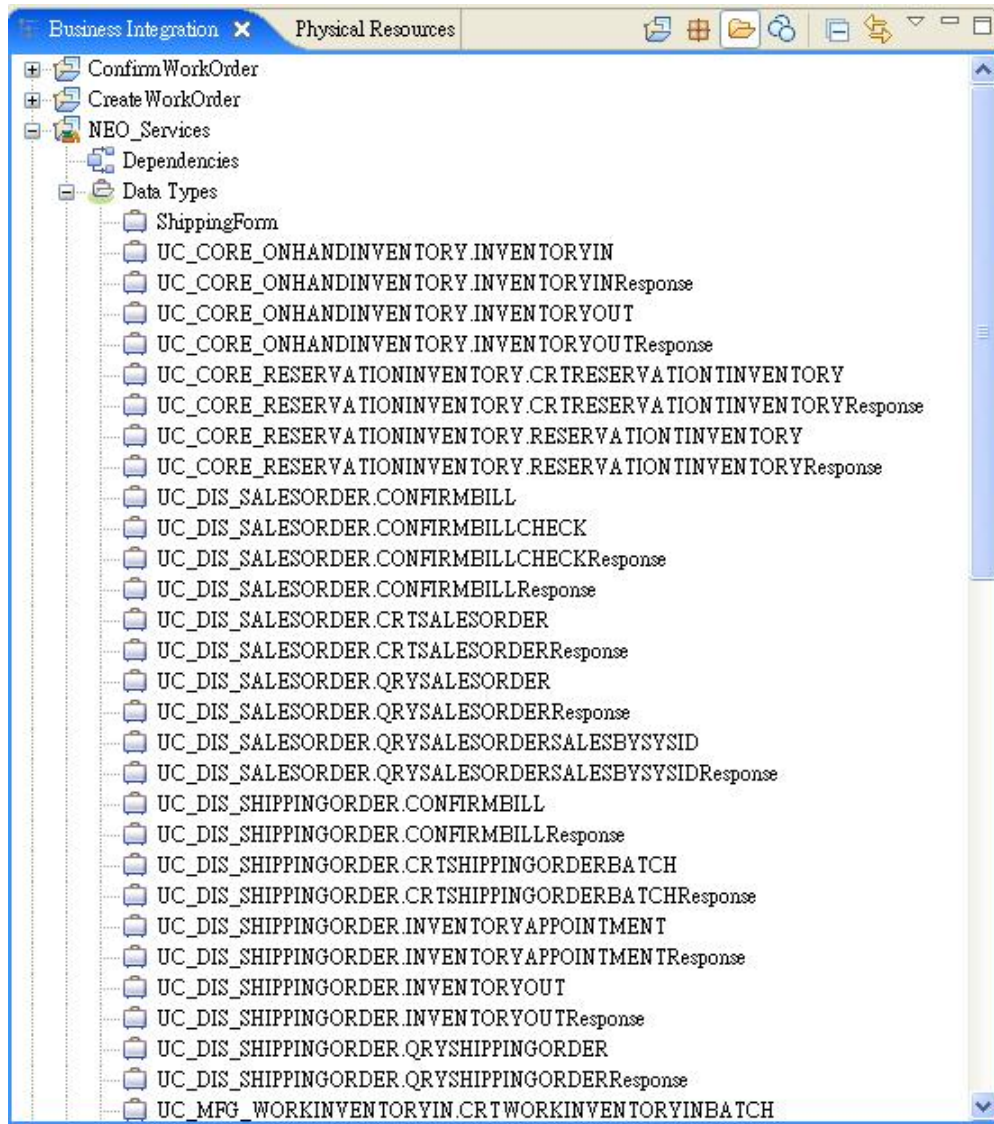


圖 4-11 WSDL 檔匯入後依介面作相對應的資料型態

本研究設計的出貨服務本身也是一個符合服務元件架構標準的服務，必須定義一個介面提供給其它模組或服務呼叫。本研究將呼叫出貨服務的輸入介面直接傳送給第一個呼叫的 DIS_SALESORDER.QRYSALESORDER 服務，因此將出貨服務的輸入介面定義成 DIS_SALESORDER.QRYSALESORDER 服務的輸入格式，而輸出的介面則將最後呼叫的

UC_DIS_SHIPPINGORDER.INVENTORYOUT 服務的輸出介面格式指定給出貨服務的輸出（圖 4-12）。

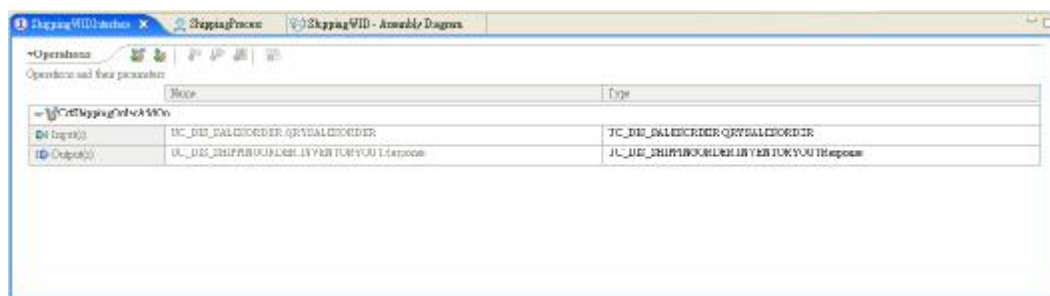


圖 4-12 定義出貨服務的介面

一、設計服務流程

完成組合圖之後，接下便是開始設計服務呼叫流程。WID 的流程編輯器是依照 WS-BPEL 標準所開發的視覺化流程編輯器（如圖 4-13），只要是放入組合圖的服務均可被使用於此。



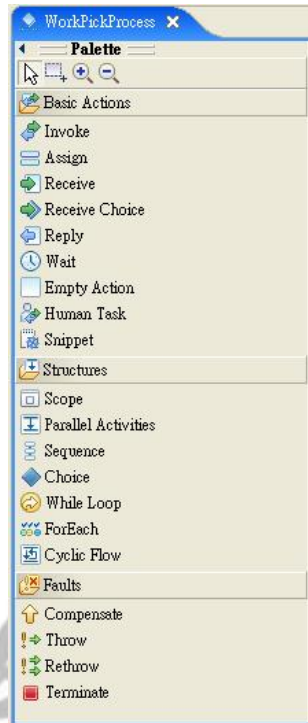


圖 4-13 WS-BPEL 視覺化編輯器

在流程定義裡，每一個流程均是由 Receive（接受）活動開始，結束於 Reply（回覆）活動（如圖 4-14）。Receive 是負責接受呼叫出貨服務所傳來的輸入資料，而 Reply 則是將服務處理的結果回覆給服務的呼叫端，Receive 和 Reply 活動之間才是放置重組服務流程所需的呼叫、分支判斷、指派等 WS-BPEL 定義的活動。

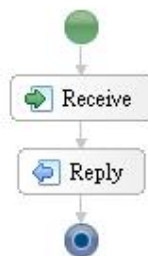


圖 4-14 服務流程設計圖

本研究的服務呼叫順序將參照圖 4-15 的流程設計進行組裝，利用 WS-BPEL 的 Invoke 指令依序呼叫各項服務。出貨服務的呼叫流程主要目的是將 NEO SOA-ERP 的部分流程予以自動化，因此無太複雜的邏輯運算，其中比較多的開發時間耗費於呼叫服務前的資料對映，在每次呼叫服務之前，將該服務所需要的輸入資料項目利用 WS-BPEL 的 Assign 活動予以指派後再呼叫服務。

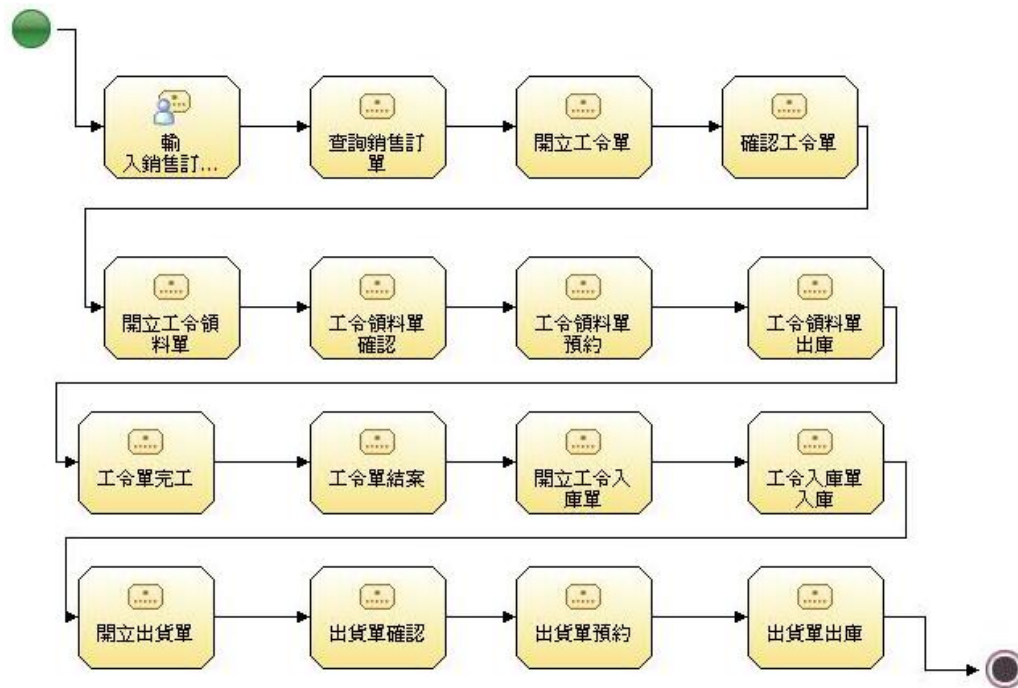


圖 4-15 出貨模組的服務呼叫順序圖

資料來源: 本研究整理

每一個被呼叫的服務在執行後會傳回該服務在介面裡所定義的輸出資料（服務是否傳回輸出資料是依據該服務的介面定義而定，服務的介面可以是要求（輸入）及回應（輸出）或只有要求而沒有回應等方式），這些輸出資料需要被儲存以供後續操作使用，故每一個 Invoke 活動在呼叫服務前都需指定一組輸入及輸

出變數給 Invoke 活動。

WS-BPEL 如同一般程式語言一樣可以自定變數，以儲存資訊供流程內部使用，Invoke 活動所需的變數可直接在 BPEL 圖型編輯器中定義並指定給 Invoke 活動，這些變數的資料型態是依據使用的服務介面而定，資料型態則是來自 WSDL 檔匯入時所建立的介面資料型態。

本研究為每一個服務分別建立一組輸入和輸出變數（如圖 4-16）並指定給每一個 Invoke 活動。



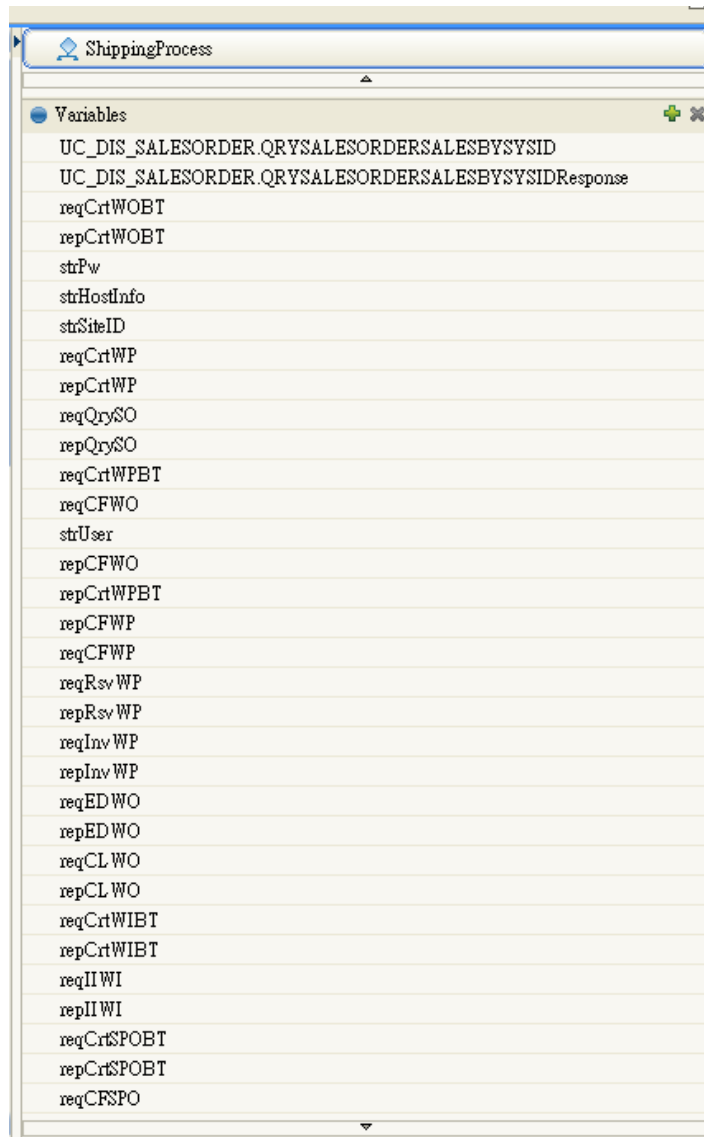


圖 4-16 本研究所定義的介面變數

由於 NEO SOA-ERP 系統所提供的服務在呼叫時需提供大量的表單資訊作為輸入，因此本研究利用 Assign 活動將個別服務所傳回的輸出資料指派給下一個呼叫的服務輸入（如圖 4-17），以達到自動化的目的。

Assign From	Assign To
ctrlsec	msgCnWOBTSENDERACCOUNT
ctrlpw	msgCnWOBTSENDERPASSWORD
ctrlordinfo	msgCnWOBTHOSTINFO
ctrlsiteID	msgCnWOBTDATA[1]SITEID
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]ISOWO	msgCnWOBTDATA[1]ISOWO
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]SEQUENCENO	msgCnWOBTDATA[1]SEQUENCENO
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]ITEMID	msgCnWOBTDATA[1]ITEMID
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]SCHEDULEDELIVERYDATE	msgCnWOBTDATA[1]SCHEDULEDELIVERYDATE
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]INVENTORYUMID	msgCnWOBTDATA[1]INVENTORYUMID
msgQoS0 DATA[1]BILLTOCUSTOMERID	msgCnWOBTDATA[1]SHIPTOCUSTOMERID
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]SHIPPINGWAREHOUSEID	msgCnWOBTDATA[1]SHIPPINGWAREHOUSEID
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]SHIPPINGSITEID	msgCnWOBTDATA[1]SHIPPINGSITEID
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]SALESORDERQUANTITY	msgCnWOBTDATA[1]WORESTARTQUANTITY
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]BOMID	msgCnWOBTDATA[1]BOMID
DC_DE_SALESORDER.QRYSALESORDERSALESSEVTSTIDResponse DATA[1]ECVERSION	msgCnWOBTDATA[1]ECVERSION

圖 4-17 Assign 活動操作圖

完成上述的各項設定之後，在 Receive 和 Reply 活動之間（圖 4-14 所示），依設計好的順序放入成對的 Assign 與 Invoke 活動，即可完成服務流程的設計工作。

有關本研究所實作的完整模組專案內容請參閱附件一。

二、模擬

在服務流程的設計過程中即可對流程進行模擬測試，WID 提供模擬測試的功能，並且在測試時可以手動輸入模擬資料（如圖 4-18）以模擬執行結果。

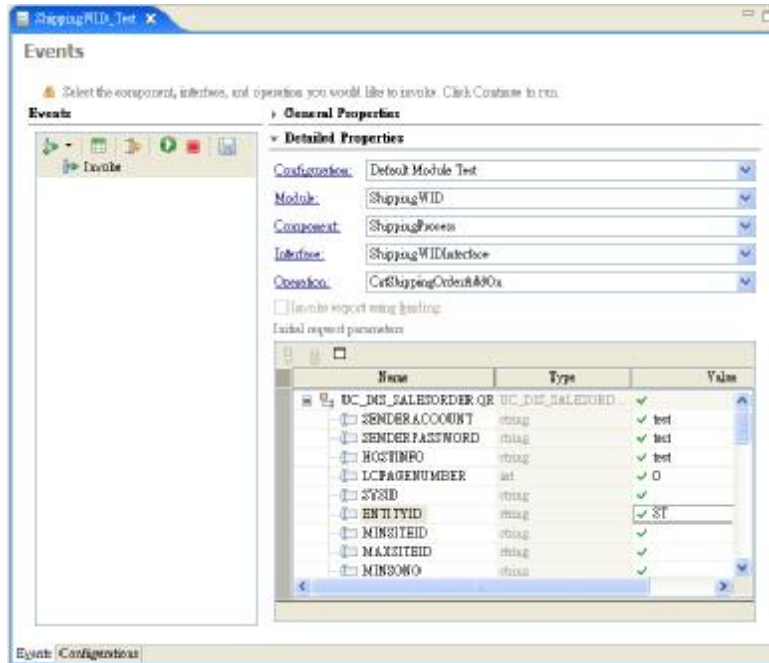


圖 4-18 WID 的模擬測試介面

本研究利用模擬測試的功能，在每完成一組 Assign 與 Invoke 活動的設計後進行模擬測試，並將模擬結果與人工操作界面所產生的結果進行比對，確保每一個步驟的正確性，同時可以達到設計完成即驗證完成的結果，也就是說，在流程開發完成之時，測試作業即可完成。

三、部署

開發完成的模組必須部署到流程伺服器才能被呼叫使用。本研究使用的 WID 可自行將開發完成的模組自動產生相關資訊後部署到 WebSphere Process Server 上。

第四節 實作心得

一、系統提供的服務功能完整性影響設計效率

服務如同 IC 元件一樣，IC 元件經過挑選後，被用來組裝設計

出新的電子產品，每一個 IC 元件均有其特定的功能，所使用的 IC 元件若有功能不足的情形時，就要考慮尋找輔助的元件或更換成其它功能更符合需要的元件。

服務在組裝流程的過程中也會有相同的情形。每一個服務都有其特定的功能以提供特定的「服務」，在組裝的過程中，若有需要的功能無法從服務中取得，那麼便需要自行製作或組裝其他服務，以滿足需要。

如果選用的系統可提供較多功能的服務，對於商務流程的設計開發將會更有效率，服務的組裝人員將可更專注於商務流程的需要，而不用多花心思在服務的開發工作上。

以本研究所使用的 NEO SOA-ERP 系統為例，提供查詢表單的服務多達 4~5 種，因此在流程設計時，便可經由不同的服務取得不同的資料結果，減少取得資料後的再處理程序。

因此，本研究建議服務導向架構的系統應該提供越多的服務越好。更多的服務意謂更多的選擇和較少的限制，使得商務流程的開發者可以設計出更有效率、更符合企業需要的商務流程，避免因服務的不足而需要花費大量的時間在開發或組裝新服務，而失去使用服務導向架構的利基與價值。

越多的服務並不是指漫無目的增加服務的數量，而是以提供「完整」的服務為前提的。所謂的「完整」是指系統的開發者應以服務被使用的需要性為考量，提供滿足不同使用需求的服務。非服務導向架構的軟體系統是封閉的系統，使用者只能透過介面操作系統所提供的功能，而在服務導向架構下，使用者可以直接參與系統的設計與開發，甚至是自行組裝服務以達到需要的功能。因此，服務的設計思維必須改變，必須以不特定未來使用者在應

用時需要的功能為考量，儘量的提供使用者應用所需的服務。

二、最小邏輯單元的基層服務

服務可以再組裝成新的服務，組合不同的服務產生出新的服務，以滿足商務流程的各種需要，這是服務導向架構與商務流程管理系統整合後所能帶來的效益之一。不過，本研究在實作中發現，服務雖然可以經由不斷的重新組合而產生新的服務，但是，由於多個服務組合在一起時便會發生介面銜接和服務功能含蓋範圍的問題。

越多的服務組合在一起，服務與服務的界接需求就會增加，如果每一個服務都具備複雜的介面時，因組合而產生的資料指派失誤便有可能削減服務在組裝上所帶來的效益。

每一個服務都有特定的功能，若是把單一的服務設計成可依參數的不同而提供不同的功能時，便失去服務的組合彈性與效率（一般的情況下，較多的功能需要較多的程式碼才能完成，程式碼的多寡是影響執行效率的因素之一），同時也會增加服務內部的程式碼複雜度。

因此，本研究建議，每一個服務應只提供單一的功能，不同的功能由不同的服務提供，而不是被包含在一個服務內。每一個服務的功能應儘量明顯而單純，尤其是基層的服務應該且僅包含最小的程式邏輯，如此才能增加組合上的彈性，而功能複雜的服務則應由簡單的基層服務所組合而成，避免從頭編寫程式的情形發生，但是，仍舊需維持單一功能的特性。

另外，服務的功能越單純所需的輸入資料便會越少，將有助於介面的簡化，而減少服務於組裝時的界接問題。

三、服務命名的唯一性

NEO SOA-ERP 系統中，所有表單的「確認」服務均命名為「CONFIRMBILL」，使用視覺化工具開發時，相同的名稱便會發生衝突，開發工具會自行將名稱變更，加入序號以供區分。本研究所使用的「確認」服務共有四個，因此在建模和流程設計的過程中，同名的服務會增加識別的困擾和流程錯誤的機會，降低開發的效率。



圖 4-19 相同名稱的服務會造成辨識困擾

服務的命名應儘量達到可辨識性和唯一性的目標，尤其商務流程開發工具多以視覺化方式進行，相同的命名在視覺化元件的表示上便容易失去代表性而造成失誤，因此，服務的命名應考慮名稱空間（Namespace）外，還需考慮到名稱的唯一性。

四、服務介面應儘量精簡

本研究在實作服務的串接時，遇到最耗時的問題便是 NEO SOA-ERP 系統的服務介面需輸入大量的資訊，尤其是新增表單的服務所需帶入的輸入項目就非常多，就算改用批次自動新增的方式，也是沒有獲得太大的改善。

服務的輸入介面若要求較多的輸入項目時，會在呼叫該服務時，因為輸入項目的不完整而呼叫失敗，尤其在無需使用者操作的服務間呼叫時，大量的輸入資料需要在呼叫該服務前，先找出

可提供輸入資訊的服務，在呼叫執行之後才能利用這些服務的輸出來指派給真正要呼叫的服務，造成原本看來只需要一個服務的功能，實作時卻需要更多的服務才能真正的取得希望的結果。

因此，服務的介面應該減少所需的資料項目，尤其是對於不需要使用者輸入資料或可以串接成自動化作業的服務，應透過組裝成新服務的方式，將原本需要在呼叫時由使用者指定資料項目值的動作，改由包含在服務內部的多個服務自動完成。以本研究的在批次開立工令單的例子而言，批次開立工令單的資料多來自銷售訂單，但是呼叫使用批次開立工令單時，卻需要使用者將銷售訂單的部分資料以人工方式指派給批次開立工令單的服務，所以使用者在使用批次開立工令單的服務單之前，必需先呼叫查詢銷售訂單與查詢銷售訂單明細等二個服務之後才能開始呼叫批次開立工令單的服務。

但是，並不是每張工令單都可以由銷售訂單百分之百的串接起來，為了兼顧服務使用的彈性，服務的開發者多會選擇採用開放式的欄位供使用者自行決定，不過，此種服務對使用者所帶來的困擾可能會多於彈性，服務的開發者應為此多設計一個已經組合完成而只需輸入銷售訂單號碼的批次開立工令單服務，如此便可簡化服務介面，使用者只需輸入銷售單號即可。

另外，如同本節在第一點所討論的完整性問題，服務應以使用者的需求為考量，批次開立工令單服務除現行的方式外，應該考慮再新增一個只需輸入銷售訂單編號便可批次開立工令單的服務，如此才能真正提供使用者需要的服務組合彈性和簡化的服務介面。

五、直接部署商務流程模型的可行性

IBM 的商務流程管理系統提供從商務流程建模、服務組裝、部署到監控等一系列可串連在一起的工具。本研究於初期曾嘗試採用整合開發的方式實作個案的商務流程，將個案的商務流程於 WebSphere Business Modeler 繪製後轉到 WID 中進行整合作業，但由於 NEO SOA-ERP 系統裡，服務和服務串接的介面不相容，因此無法直接在 WebSphere Business Modeler 繪製所需的商務流程後再轉入到 WebSphere Integration Developer 中整合。



圖 4-20 介面不相容使得服務無法由工具自動串接

若 NEO SOA-ERP 系統的服務介面可以簡化或新增專門轉換服務與服務介面的中介服務，則可將建模、組裝到部署作業一體化，加快商務流程設計與開發的速度。

商務流程的建模主要由非具商務流程知識的專家或使用者所進行的活動，若服務能直接被運用於商務流程的建模活動中，依據商務流程模型而轉換而成的系統將會更貼近企業的需要，同時也能幫助使用者及專家瞭解，系統所能提供的服務與限制，減少使用者與商務流程系統開發者之間的認知落差，將有助於軟體系統開發專案的推行，減少推行的障礙。

六、軟體開發的分工將更專業化

從本研究實作的經驗中發現，即使對 Java、WS-BPEL 等程式語言不熟悉，或是不清楚服務的程式邏輯為何，服務的使用者只要知道服務的功能及需要的輸入資訊為何，就可以運用服務來組合出需要的新服務。因此，設計開發軟體系統的工程師在服務導向架構下，只要能夠找到合適的服務並予以組合，便不需要如傳統架構一般，需要花時間撰寫程式碼。

從上述的角度看來，軟體系統的開發者和服務的開發者會朝二個不同的方向發展，軟體系統的開發者將越來越倚重視覺化工具從事服務重組的工作，與程式碼接觸的機會將越來越少，工作的技術性也會隨之降低，工作重心將轉移到如何利用服務為企業組裝出所需的軟體系統，以及如何發揮創意，利用服務的重組為企業帶來新價值，因此進入的技術門檻將會降低。

而服務的開發者則會朝向更專業、專精的方向移動，以個人的技術能力創造出滿足客戶需要的服務，工作對技術能力的要求會越來越高，並發展成為一個專門的工作領域，就像 IC 設計一般。



第五章 結論與建議

第一節 結論

本研究經實作後證明，個案能夠在不需對 ERP 系統進行客製化的條件下，運用商務流程管理系統達成變更 ERP 系統商務流程的目的。由本研究的結果可知，商務流程管理系統對 SOA-ERP 系統在導入和日後的商務流程變更上是有效益的。

本研究依據實作結果提出下列結論：

一、ERP 系統的導入方式將改變

服務導向架構的 ERP 系統整合商務流程管理系統之後，原本需費時、費力調整的商務流程將變的簡單，企業在導入 ERP 系統時將不需要變更現行商務流程來適應 ERP 系統的預設流程，這將降低企業導入的障礙，增加導入的成功機會。

二、軟體租賃服務模式將改變

在服務導向架構環境下，使用者對系統任何的操作都是在進行遠端呼叫服務的動作，服務所處的實際位置對使用者而言並不重要，再加上整合商務流程管理系統後，商務流程的客製工作變的容易與低風險。

過去的軟體租賃服務因軟體系統的客製困難及客製之後的維護成本提高，導致一套軟體系統必需適用於各類型企業，而降低企業使用的意願，限制住可以使用的客戶層和軟體租賃服務供應商的獲利能力。但是在服務導向架構系統與商務流程管理系統整

合的環境下，每一個客戶可以有不同的商務流程，但共用同一套服務導向架構系統，也就是說，雖然每一個客戶有不同的商務流程，但卻不需對系統進行修改，只需對個別客戶建立特性資料 (Profile)，存放個別客戶的商務流程，如此軟體租賃的供應商可提供具彈性又低成本的軟體租賃服務供企業使用，使得軟體租賃供應商與企業都能獲取利益，而增加企業使用的意願。

三、軟體產業分工的專業化

在服務導向架構與商務流程管理系統的整合環境下，商務邏輯、商務流程與使用者界面分為三個不同的層，而軟體系統開發的工作也會朝此三個方向分開發展。

商務邏輯是真實程式碼所對應的元件，即服務，此部分對程式撰寫的技術要求比較高，將會朝更專業的方向發展，慢慢成為一個新的服務元件設計產業。

商務流程是企業應用軟體系統的重心，現有的企業應用軟體開發商必需進行轉型，如同 IBM 近年來所進行般的轉型一樣，必需強化自身在商務領域的知識 (Domain Know-How) 才能幫助企業利用軟體系統解決問題，將利基從軟體技術轉向整合服務元件與領域知識為企業創造具價值的軟體系統。

而使用者界面將變的更多元化，使用者界面將完全脫離商務邏輯和商務流程的限制，可以實作在各種人機界面上，讓軟體的使用變的更有彈性，也讓參與的設計者類型擴增，不再限制於具技術背景的介面設計人員。

第二節 研究限制

服務導向架構系統與商務流程管理系統均是新興的資訊技術，相關開發工具與技術仍在發展中，本研究已盡可能研習相關的知識與技術，解決實作過程所遭遇的問題，但是仍有可能因為對工具的不熟悉而致使未採用較好的解決方案。

在個案的實作上，本研究因時間因素及專注於商務流程管理系統所帶來的影響，故實作範圍僅包含個案在導入 ERP 系統時，商務流程需要做最大調整的部分，並未包含其他導入時需進行小部分客製的商務流程，同時，也未考量除了商務流程以外的其他因素，將有可能影響本研究的代表性。

第三節 後續研究建議

本研究的個案為一中小型企業，需求以縮減商務流程為主。在實務上，企業變更商務流程的方式除縮減外，也可能是擴充，因此，建議針對需要擴充商務流程的個案進行相關的研究，以瞭解更多服務導向架構系統與商務流程管理系統整合後，為企業所帶來的效益。

參考文獻

1. 李訓仁，*BPM 資訊系統實務應用特性之研究*，中原大學資訊管理學系碩士論文，2007。
2. 尚榮案譯，Yin Robert K.著，*個案研究法*，台北，弘智文化，2001。
3. 紀尚德，*服務導向架構的 ERP 系統客製方法實證研究-以「新傳興業股份有限公司」為例*，輔仁大學資訊管理研究所碩士論文，2007。
4. 蔡良玉，*流程調整的策略與 ERP 應用的程度對 ERP 成效影響之研究*，元智大學資訊管理學系碩士學位論文，2002。
5. Behara, G. K. , *BPM and SOA: A Strategic Alliance*, Business Process Trends, May 02, 2006. Retrieved October 12, 2007, from the World Wide Web: http://bptrends.com/deliver_file.cfm?fileType=publication&fileName=05%2D06%2DWP%2DBPM%2DSOA%2DBehara%2Epdf.
6. Beisiegel, M., Blohm, H., Booz, D., Edwards, M., Hurley, O., Ielceanu, S., Miller, A., Karmarkar, A., Malhotra, A., Marino, J., Nally, M., Newcomer, E., Patil, S., Pavlik, G., Martin Raepple, M., Rowley, M., Tam, K., Vorthmann, S., Walker P., & Waterman L., *Service Component Architecture Assembly Model Specification SCA Version 1.00*, Open Service Oriented Architecture collaboration, 2007. Retrieved March 10, 2008, from the World Wide Web: http://www.osoa.org/download/attachments/35/SCA_AssemblyModel_V100.pdf?version=1.
7. Bloomberg, J. & Schmelzer, R., *Service Orient or Be Doomed!: How Service Orientation Will Change Your Business*. Hoboken, John Wiley & Sons, 2006.
8. Chapman, M., Ielceanu, S., Koenig, D., Rowley, M., Trickovic, I., & Yiu, A., *Service Component Architecture Client and Implementation Model Specification for WS-BPEL SCA Version 1.00*, Open Service Oriented Architecture collaboration,

2007. Retrieved March 10, 2008, from the World Wide Web: [http:// www.osoa.org/download/attachments/35/SCA_ClientAndImplementationModelforBPEL_V100.pdf?version=1](http://www.osoa.org/download/attachments/35/SCA_ClientAndImplementationModelforBPEL_V100.pdf?version=1).
9. Davenport, T. H., *Mission critical: Realizing the promise of enterprise systems*, Boston, Harvard Business School Press,2000.
 10. Erl , T., *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*, Upper Saddle Rive:Prentice Hall PTR, 2004.
 11. Erl , T., *Service-Oriented Architecture: Concepts, Technology, and Design*, Upper Saddle Rive:Prentice Hall PTR, 2005.
 12. Harmon, P., *Service Orientated Architectures and BPM*. Business Process Trends, Feb 22, 2005. Retrieved March 12, 2008, from the World Wide Web: http://www.bptrends.com/deliver_file.cfm?fileType=publication&fileName=bptem ailadvisor022205%2Epdf.
 13. Holdsworth, S., Ielceanu S., Karmarkar, A., Little, M., Patil S., & Rowley, M., *Service Component Architecture Web Service Binding Specification SCA Version 1.00*, Open Service Oriented Architecture collaboration, 2007. Retrieved March 10, 2008, from the World Wide Web: http://www.osoa.org/download/attachments/35/SCA_WebServiceBinding_V100.pdf?version=2.
 14. Hollingsworth, D., *The Workflow Reference Model*, Workflow Management Coalition, 1995. Retrieved October 25, 2007, from the World Wide Web: <http://www.wfmc.org/standards/docs/tc003v11.pdf>.
 15. IBM WebSphere Integration Developer Version 6.1 information center, IBM, 2008. Retrieved March 10, 2008, from the World Wide Web: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp>
 16. IBM WebSphere Business Modeler Advanced Version 6.1 information center, IBM,

2008. Retrieved March 10, 2008, from the World Wide Web: <http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.btools.help.modeler611.doc/doc/welcome/dochome.html>.
17. Jeston, J. & Nelis, J., *Business Process Management Practical Guidelines to Successful Implementations*, Burlington: Butterworth-Heinemann, 2006.
18. Jordan, D., Evdemon, J., Alves, A., Arkin, A., Askary, S., Barreto, C., Bloch, B., Curbera, F., Ford, M., Goland, Y., Guízar, A., Kartha, N., Liu, C. K., Khalaf, R., Köni g, D., Marin, M., Mehta, M ., Rijn, D. V., D., Yendluri, P. & Yiu, A., *Web Services Business Process Execution Language Version 2.0 OASIS Standard*, Organization for the Advancement of Structured Information Standards, 2007. Retrieved May 25, 2007, from the World Wide Web: <http://docs.Oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>.
19. Josuttis, N. M., *SOA in Practice*, Sebastopol: O'Reilly Media, Inc., 2007.
20. Kamoun, F., A Roadmap towards the Convergence of Business Process Management and Service Oriented Architecture, *Ubiquity*, Vol. 8 , Issue 14, 2007, pp.1-8.
21. Karmarkar, A., Malhotra, A., & Booz, D., *Service Component Architecture (SCA) Tutorial: Part 1*, Organization for the Advancement of Structured Information Standards, 2007. Retrieved April 15, 2008, from the World Wide Web: http://www.osoa.org/download/attachments/250/SCA_Tutorial_Part1_Asia-updated.pdf?version=1.
22. MacKenzie, C. M., Laskey, K., MacCabe, F., & Brown P. F., *Reference Model for Service Oriented Architecture 1.0 Committee Specification 1*, Organization for the Advancement of Structured Information Standards, 2006. Retrieved January 8, 2008, from the World Wide Web: <http://docs.oasis-open.org/soa-rm/v1.0/>

soa-rm.pdf.

23. Muehlen, M. Z., *Workflow-based Process Controlling: Foundation, Design, and Application of Workflow-driven Process Information System*, Berlin:Logos Verlag, 2004.
24. Portougal, V. & Sundaram, D., *Business Processes: Operational Solutions for SAP Implementation*, Hershey: IRM Press, 2006.
25. Wahli, U., Avula, V., Macleod, H., Saeed, & M., Vinther, A., *Business Process Management: Modeling through Monitoring Using WebSphere V6 Products*, IBM.Com/Redbooks, 2006.
26. Weaver, R., *The Business Value of the Service Component Architecture (SCA) and Service Data Objects (SDO)*, International Business Machines Corp, 2005.
27. Weske, M., *Business Process Management Concepts, Languages, Architectures*, Berlin: Springer, 2007.
28. Zimmermann, O., Doubrovski, V., Grundler, J., & Hogg, K., *Service-Oriented Architecture and Business Process Choreography in an Order Management Scenario: Rationale, Concepts, Lessons Learned*, Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, 2005 , pp.301-312.



附 錄 一

出 貨 模 組 專 案 報 告