IFBP：一個以資訊流的角度探索企業流程的機制

黃士銘
中正大學會計與資訊科技研究所
俞福鈞
中正大學資訊管理研究所

摘 要

在企業流程管理(BPM)的活動中，流程的設計與程序診斷兩階段是主要的核心工作，因為這將決定企業未來組織架構、業務流程等攸關企業未來走向之架構。現今在程序診斷階段中，企業舊有之流程之取得，一般皆由負責專案之顧問至企業中經由實地訪談使用者的方式取得，不但耗費時間，顧問本身經驗與實力對於企業流程的分析亦有所影響，而訪談同時亦會影響企業流程的運作，間接增加企業人力成本。由於現今企業經營已大量的使用資訊系統 因此企業的許多作業流程知識亦已存在於資訊系統的資訊流之中。是故，在本論文中提出一套整合轉換機制 IFBP，其可利用企業現有的資訊流程將其轉換成企業現有作業流程，提供診斷企業現行流程之一輔助工具，減少所需時間。IFBP機制主要包含有移植、整合、與轉換等三個階段，它的輸入為企業內各系統的資訊流程圖(DFD)，而輸出的結果則為企業流程塑模常使用的事件驅動流程鏈(EPC)圖。本論文最後以 Information Capacity 理論來驗證轉換的方法的正確性，並以雛形系統為實例個案來驗證可行性與評估績效。

關鍵字： 企業流程管理、企業資源規劃、資料流程圖、事件驅動流程鏈、資訊承載。

IFBP: A Methodology for Business Process Discovery Based on Information Flow

Shi-Ming Huang
Department of Accounting and Information Technology, National Chung Cheng University
Fu-Yun Yu
Department of Information Management, National Chung Cheng University

Abstract

Knowledge management (KM) has become a critical activity in business administration. However, there exists a significant gap between knowledge management and information technology: automatic classification of dynamic business knowledge, since business knowledge is often dynamic and accumulative. This paper aims to explore and reduce the gap so that information technology may really assist businesses in managing knowledge. An agent-based collaborative knowledge classification and management model CKCMA is proposed. The agents collaborate with each other in order to closely link people to knowledge, and vice versa. They negotiate with each other to classify input knowledge documents and requests, and accordingly identify which agent should be in charge of the related KM activities, including knowledge elicitation, accumulation, distribution, and sharing. An experiment is designed to empirically evaluate CKCMA's multiagent knowledge classification technique. In the experiment, the feasibility and contributions of context-based automatic knowledge classification are verified.

Keywords: Business Process Management (BPM), Data Flow Diagram (DFD), Event-driven Process Chain (EPC), Information Capacity.

壹、緒論

資訊科技的日新月異改變了人們對於商業構想、設計以及運作的觀念。為了邁向全球化的經營模式，許多的企業從傳統集中式組織朝向分散式、甚至是虛擬組織的方向前進。為了達到此願景，企業已察覺到適當地運用資訊科技的重要性，並且已建置許多的大型應用程式系統以輔助日常的商業營運。是因如此，愈來愈多研究的焦點著重於企業流程管理(Business Process Management, BPM)的領域，甚至更有學者預測將來企業流程將會主導成為任何電子化解決方案之核心[Sheth et al., 1999; Basu and Kumar, 2002]。

當前企業為了達成永續經營，面對不明確的顧客需求及不斷變動的環境下，必須重新調整經營策略，並檢討應用程式系統之作業流程，盡量降低成本與提升生產力，企業流程的程序診斷與流程的設計就變得相對的重要。程序診斷部分，必須先從企業中獲取現行運作之程序，包括從始至終的程序過程、程序的組成元件、現行程序績效，接著找出現行程序中的缺失。而在流程設計階段，則針對診斷出的缺失，研擬出較為合理之新程序[Davenport, 1993; Grover et al., 1995; Huang et al., 2001]。

目前應用程式系統之流程規劃通常配合資訊技術來進行，例如近年來在產業界漸成風氣的企業資源規劃(Enterprise Resource Planning, ERP)系統。一般在導入 ERP 時，其在程序診斷階段中，一般皆由負責 ERP 系統導入之廠商派遣顧問至企業中經由訪談方式取得，而後再以人工方式繪製。由於此一部分必須仰賴人力完成，況且不同產業具備獨有之領域知識，非同一領域內之顧問其專業知識恐嫌不足，其所分析出的流程與實際情形恐有出入，如此一來，企業除了必須付出大筆的顧問費之外，還必須付出為了修正錯誤所導致的時間成本[Kliem, 2000; Huang et al., 2001; Hong and Kim, 2002]。

現階段在程序診斷的另一個問題，大多數的方法皆忽略企業內原本運行的資訊系統流程而直接進行使用者訪談。資訊系統流程之設計，其目的就是在支援企業中各流程之正常運行。現今大部分企業均已電腦化，在舊有系統中早已存有企業原本運行的所有營運資料以及原本企業的運作規則，其中就包含了企業的舊有流程[Butler, 1995; Marcus et al., 1999]。但是由於資訊系統資料流程設計方法與企業流程設計方式的分歧即所謂(Process Design Gap)，卻造成流程設計人員在轉換與使用上的困難[van der Aalst, 1999; Basu and Blanning, 2001; Chandrasekaran et al., 2003]。因此，若能利用既有系統之原始企業資訊系統與營運資料，經過整合轉換之後即可取得現行作業流程，再依照新需求加以修改。如此舊有資料不但得以保存，亦可符合企業最新需求，更可以減低轉換後錯誤發生之風險與修正成本[Basu and Kumar, 2002]。

有鑑於此，本論文中擬就企業現階段務流程之取得階段，利用企業現有資訊系統之系統原始設計相關資料，發展一套企業流程探索之機制 IFBP(Information Flow for Business Process)。IFBP 機制先分析企業現有各系統之資訊流程並將其整合成企業現有作業的資訊流程，然後再將之轉換成企業整體性之商業作業流程，以為企業分析其現有作業流程的基礎，以解決流程設計分歧之問題，加速流程診斷之運作。

貳、研究背景

一、企業流程塑模方法

一般常用之企業流程表示法有許多種，本研究中採用 EPC(Event-driven Process Chains)表示方式，因此以下僅就 EPC 部分加以說明[Scheer, 1994]。

EPC 為 ERP 軟體大廠 SAP AG 與德國薩爾大學(University of Saarland)所合作發展並開發出一工具 ARIS Tools，ARIS Toolset 在 Gartner Research 對 BPM 工具評比中，已蟬連數年的冠軍[Gartner, 2002]，其動機乃認為現今對於企業組織及營運業務的塑模工具太過複雜且太過於資訊技術導向。因此合力研發出此企業流程塑模觀念，並藉此作為 SAP R/3 ERP 系統導入時客戶企業流程的塑模工具[Keller and Teufel, 1998]。

EPC 除了以流程導向對企業組織與營運流程作業提出塑模之外，最重要的是其提供簡單的圖形符號達到塑模溝通之需求。基本上 EPC 是由主動元件及需某種營運情況才反應之被動元件所組成，而主動元件及包含具有觸發(Trigger)一個或多個功能作業的事件(Event)，而被動元件需要一個或一個以上之主動元件加以觸發，同時被動元件需根據預先設定之控制條件或營運規則加以判斷而做出反應，而所引發之反應結果將會產生一些事件狀態，這些事件狀態基本上又可作為主動元件去觸發關連之被動元件。表 1 為 EPC 之基本圖件符號：

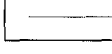
表 1：EPC 之基本圖件符號表

圖示	名稱	說明
	事件(Event)	用來描述狀態的發生並扮演觸發者角色
	工作或功能(Task/Function)	用來描述從初始狀態到終止狀態的轉換機制
	OR AND XOR	用來描述事件與功能的邏輯關係
	控制流(Control Flow)	描述事件及功能的邏輯相依性以及時間先後性
	資訊物件(Information object)	代表真實世界的實體或營運物件
	檔案(File)	代表真實系統中之檔案
	分派處理之組織單元或人員	代表公司的一組織單位或人員對功能作業之分派處理
	資訊流(Information flow)	用來定義對功能是否對資料作存取需求
	分派(Assignment)	用來描述處理功能所需的組織單元或資源

二、資訊流程的表示法

常用的資訊流程表示方式許多種，本研究的目的為分析現有資訊系統的資料流程，因此選用傳統的資訊系統分析與設計較為廣泛使用的 DFD(Data Flow Diagram) 圖作為資訊流程的表示方式[Whitten and Bentley, 1999]。DFD 具備以下四種元件：外部實體(External Entity)、功能(Function)、資料儲存體(Data Store)、資料流(Data Flow)，如表 2 所示。

表 2：DFD 元件表

圖示	名稱	說明
	外部實體(External Entity)	代表在系統範圍之外的實體
	功能(Function)	代表一項必須執行的特定工作項目
	資料儲存體(Data Store)	用來承載作業資料
	資料流(Data Flow)	在各元件之間作資料交換

三、企業流程管理的相關研究

近幾年來由於企業電子化的快速發展，企業流程發掘與設計的相關研究變得相對的重要，特別是對於不同應用程式系統間，所採用的流程塑模方式的分歧(gap)，對於新應用系統的建置與整合有著重要之影響性[Basu and Kumar, 2002]。表 3 列舉出一些和本研究較為直接相關的研究與其研究內容摘要。

表 3：企業流程管理相關研究摘要

研究者	研究內容摘要
[van der Aalst, 1998] [Kwahk and Kim, 1999]	發展高深的線性代數以及圖形理論的方法，去分析與發掘現在企業流程模式當中，可能存在之衝突、錯誤。
[Ranganathan and Dhaliwal, 2001][Themistocleous et al., 2001][Huang et a., 2002]	在企業導入大型的 ERP 系統以及進行大行 BPR 專案時，很重要的關鍵因素就是對於企業現在流程之診斷與分析之正確性、完整性，以及如何快速將原有企業流程迅速轉換到新系統當中。
[Marcus et al., 1999][Sheth et al., 1999][Huang et al., 2001]	發展流程導向之企業流程參考模式，用以分析與架構整合性企業流程，說明資料導向流程設計之缺點，以及流程導向之重要性。
[Basu and Blanning, 1999][van der Aalst, 1999][van der Aalst, 2000][White, 2002]	透過線性代數、中介圖形(meta-graph)理論與樣例化(pattern)等機制，轉換不同的流程設計方法與異質流程整合。

從這個摘要整理當中我們可以得知，目前在企業資訊系統以及企業流程管理等相關研究領域中，研究人員所著注的焦點已經從過去的資料導向之流程設計方式，漸漸轉移至較以企業流程導向之流程設計方式，而如何能夠有效將現有之企業流程加以診斷，發掘出其中所可能存在之衝突、錯誤，協助新系統快速導入將會是未來愈形重要之研究議題。

雖然已經有研究發現資料導向與流程導向轉換之重要性，但是目前仍卻少相關之系統化方法有效將兩者間作一迅速轉換，是故本研究就是希望能夠從過去企業用來表達流程的資料導向 DFD 圖中，發掘出原有之企業流程並以流程導向的 EPC 圖來表達企業流

程，透過開發這樣系統化轉換的方式，期望可以協助與彌補顧問訪談方式，所可能產生之分析不完整情況，進而彌合(bridging)兩種流程塑模方式之間的分歧(gap)。

參、流程探索與整合模型

本章節主要陳述 IFBP 機制的運作方法。IFBP 機制主要包含有移植(Transformation)、整合(Integration)、轉換(Conversion)等三個階段，如圖 1 所示，它的輸入為企業內各系統的 DFD 資訊流程圖，而輸出的結果則為企業流程塑模常使用的 EPC 圖。

一、階段一：資訊流之移植(Transformation)

DFD 屬於圖形表達之方式，為了能從中萃取出企業之流程，我們必須將之改以一結構化之語意模型方式表達，才能有效的萃取出流程資訊。要將 DFD 圖轉成語意模型，則首先需對圖上所有元件的陳列方式進行瞭解，即所謂陳列架構模型；然後再對每一元件的語意進行深入的分析，即所謂語意模型。另由於整合時兩組 DFD 可能包含不同數量之階層，因此在建置語意模型時每一組 DFD 都必須轉化為單一階層，將所有元件在單一階層中呈現，即使兩組具備不同階層數的 DFD，亦能加以整合。

(1) DFD 圖形轉換成陳列架構模型

由於一開始對於任一 DFD 內之結構並不瞭解，因此必須從 DFD 中啟始之外部實體開始辨識，再根據其輸出之資料流，辨識出下一元件，如此過程不斷持續，直到將整個 DFD 中每一元件都走訪(Travel)完畢為止，此時即可得到欲整合之 DFD 的陳列架構模型。換句話說，陳列架構模型之取得即是藉由每一被辨識出的元件的輸出，將每一元件相連，因此著重的是每一元件的輸出，因為每一筆輸出即為 DFD 圖形走訪的根據。我們依據 Butler 等人[Butler et al., 1995]之研究，並加以擴充，定義出數項標籤語言(Tag)用以分辨各項不同的 DFD 元件以為陳列架構模型的表示方法，如表 4。

表 4：陳列結構中之各項標籤說明

標籤(Tag)	說明	備註
Source_entity	系統中來源輸入點	只能出現在第 0 階層
Source_DS	某一功能節點的來源輸入點	不能出現在第 0 階層
Sink_entity	系統中最終輸出點	只能出現在第 0 階層
Sink_DS	某一功能節點的結果輸出點	不能出現在第 0 階層
Process	DFD 中各階層功能節點	
External_I	用於 Process 標籤中，表示輸入之資料流是從父功能節點取得	只出現在第 1 階層及以下之各階層 DFD
External_O	用於 Process 標籤中，表示輸出之資料流將傳送回父功能節點	只出現在第 1 階層及以下之各階層 DFD

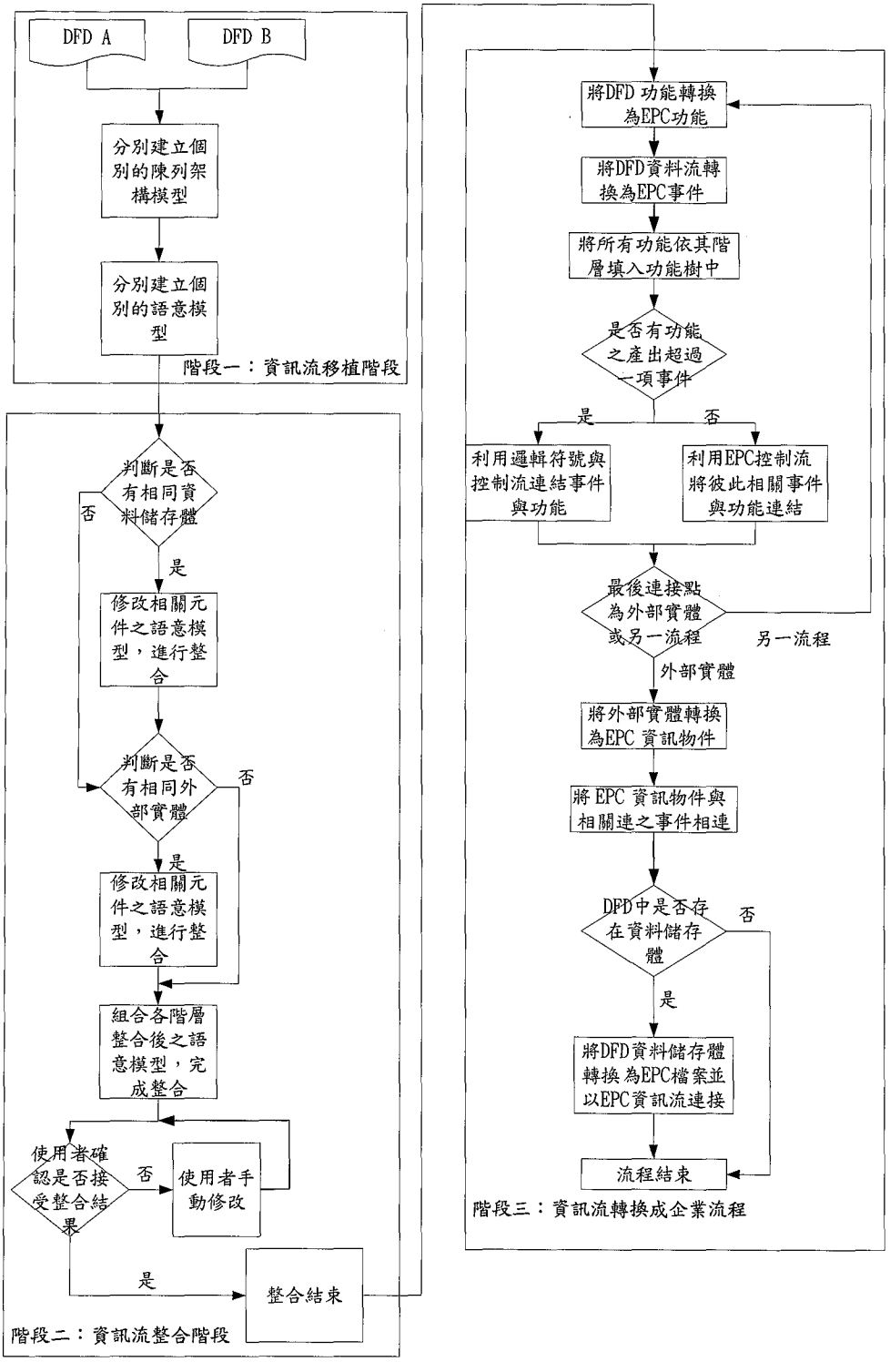


圖 1：IFBP 運作流程圖

本機制的轉換過程如下：

Step 1：從第 0 階層中的來源輸入點開始，依其種類利用標籤予以標示，並以括號標示出輸出之資料流與目的之功能節點

Step 2：來源輸入點都處理完畢後同樣將功能節點以標籤標示，同時標上階層代號，格式為”階層編號.功能節點編號”，其後同樣以括號標示出輸出之資料流與目的之外部實體。

Step 3：最後處理系統中最終輸出之外部實體，此處因為已為最終之輸出，故無輸出點，直接列出標籤即可

Step 4：將子功能節點及其輸入、輸出資料流加上標籤。若輸入之資料流來源不存在本階層中，為父階層中所產生，則需使用 External_I 標明。同理，若輸出資料流之目的不存在本階層中，則需使用 External_O 標明。

例如圖 2 為一簡易付款流程圖的陳列架構模型：

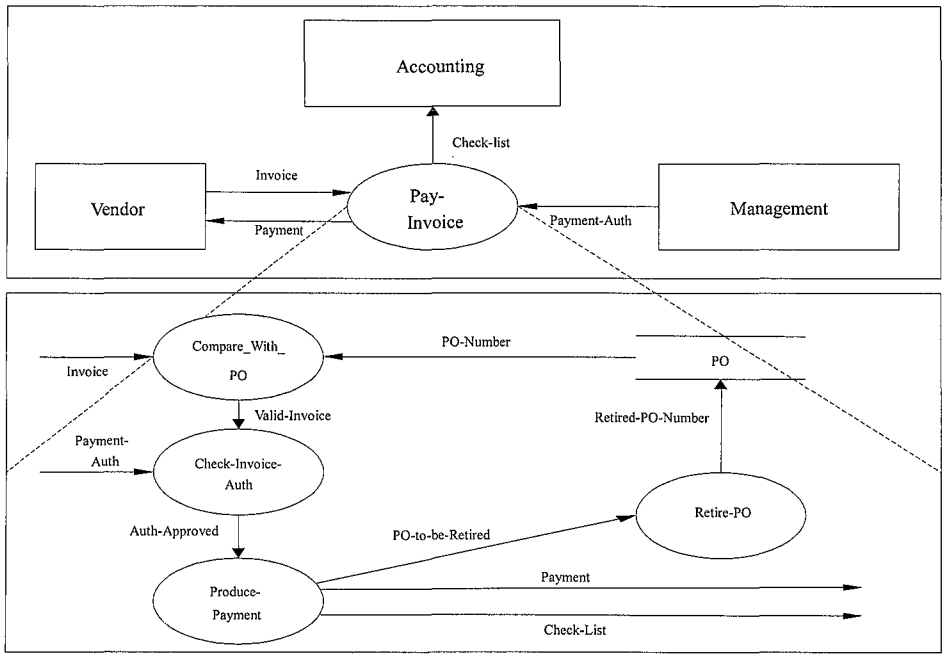
(2) 陳列架構模型轉換成語意模型

語意模型之建置主要在補足陳列架構模型之不足，陳列架構模型並沒有針對各元件的行為特性作一詳細的描述，換句話說，無法從陳列架構模型看出中看出每一元件所包含的完整運作行為。因此必須將陳列架構模型加以整理，以元件為對象，將其完整之運作行為依其所在階層加以描述。

本轉換過程中所使用之語意及符號意義如表 5 說明:

表 5：語意模型建置使用之符號說明

語意	意義	符號	說明
通用	指通用性質之語意表示方式	( )	代表一個集合
		{ }	代表功能節點之元件，與外部實體區分用
		[ ]	代表資料儲存體之元件，與功能節點區分用
		•	用於各元件之語意表達時使用，分隔輸入、輸出等不同語意之用
輸入與輸出	指每一元件之輸入與輸出語意	a	代表輸入資料流
		- *a*	代表輸出資料流
		Null	當一元件只有輸入或輸出時，則用 Null 當空缺之欄位補滿之用
分歧	描述一功能節點具有多筆輸出之關係	+	用於表達分歧之語意，功能節點可能同時產出兩筆輸出，但因目的不同，故以+將兩筆分歧之資料流相連



0:

(**Source_entity** Vendor (Pay-Invoice , Invoice)) /* Step 1 */

(**Source_entity** Management (Pay-Invoice , Payment-Auth)) /* Step 1 */

(**Process** Pay-Invoice 0 (Vendor , PAyment) , (Accounting , Check-List)) /* Step 2 */

(**Sink_entity** Vendor) /* Step 3*/

(**Sink_entity** Accounting) /* Step 3*/

1:

(**Source_DS** PO (Compare_With_PO , PO-Number)) /* Step 4 */

(**Process** Compare_With_PO 0.1 (Check-Invoice_Auth , Valid-Invoice) , **External_I** (Invoice))
/* Step 4 */

(**Process** Check-Invoice-Auth 0.2 (Produce-Payment , Auth-Approved) , **External_I** (Payment-Auth))
/* Step 4 */

(**Process** Produce-Payment 0.3 (Retired-PO , PO-to-be-Retired) , **External_O** (Payment , Check-List))
/* Step 4 */

(**Process** Retire-PO 0.4 (PO , Retired-PO-Number)) /* Step 4 */

(**Sink_DS** PO) /* Step 4 */

圖 2：DFD 範例與其陳列架構模型

本機制的轉換則直接針對各元件建立其語意模型，格式為”元件=輸入・輸出・元件本身”。為了要產生單一階層的語意模式，所有的陳列架構模型中的元件，若含有分解的子節點，則以分解後之元件取代之，如此循環運作，直至轉換完畢為止。範例圖 2 DFD 之語意模型如下圖 3 所示。

Vendor = Payment • $\overline{Invoice}$ • Vendor /* 原 Level 0 */

Management = Null • $\overline{\text{Payment - Auth}}$ • Management /* 原 Level 0 */

Accounting = Check-List • Null • Accounting /* 原 Level 0 */

[PO] = Retired-PO-Number • $\overline{\text{PO - Number}}$ • [PO] /* 原 Level 1 */

{Compare_with_PO} = (Invoice,PO-Number) • $\overline{\text{Valid - Invoice}}$ • Compare_with_PO /*原 Level 1 */

{Check-Invoice-Auth} = Payment-Auth • $\overline{Auth - Approved}$ • Check-Invoice-Auth /*原 Level 1 */

{Produce-Payment} = Auth-Approved • $\overline{PO - to - be - Retired}$ • Produce-Payment + Auth-Approved • $\overline{Payment}$ • Produce-Payment + Auth-Approved • $\overline{Check - List}$ • Produce-Payment /* 原 Level 1 */

{Retire-PO} = PO-to-be-Retired • $\overline{Retired - PO - Number}$ • Retire-PO /* 原 Level 1 */

圖 3：DFD 之語意模型圖

二、階段二：資訊流之整合(Integration)

本階段主要說明不同 DFD 資料流程整合機制。本整合機制首先將兩組不同 DFD 進行整合，在將整合後之新 DFD 與下一組 DFD 進行整合，直至所有 DFD 整合完畢為止。整合過程如下圖 4 所示。整合機制以相同元件進行整合，亦即先尋找兩組 DFD 中相同之元件，作為整合時之依據，再將相同元件之語意模型做適當之修改，即可將兩組 DFD 進行整合。整合後並交由使用者加以判斷及確認。本機制之整合步驟說明如下：

Step 1：將兩 DFD 中相同之資料儲存體加以整合成一單一資料儲存體。

Step 2：將兩 DFD 中相同之外部實體加以整合成一單一外部實體。

Step 3：將兩 DFD 中相同之功能加以整合成一單一功能。

Step 4：將兩 DFD 中相同之資料流加以整合成一單一資料流。

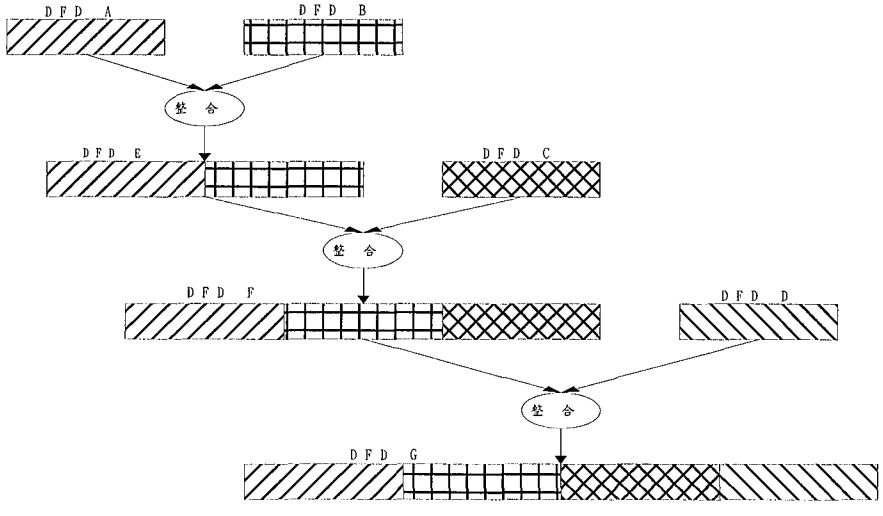
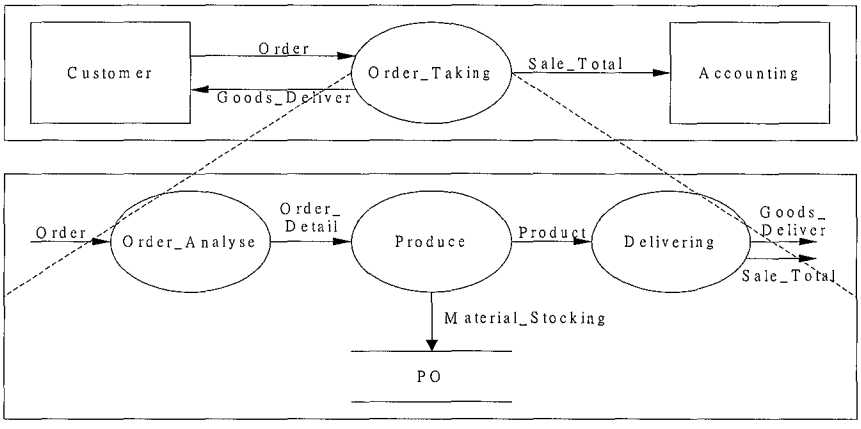


圖 4：整合流程圖



Customer = Goods_Deliver • $\overline{Order}$ • Customer) /* 原 Level 0*/

Accounting = Sale_Total • Null • Accounting /*原 Level 0 */

{Order_Analyse} = Order • $\overline{Order\_Detail}$ • Order_Analyse /*原 Level 1*/

{Produce} = Order_Detail • $\overline{Product}$ • Produce + Order_Detail • $\overline{Material\_Stocking}$ • Produce /*原 Level 1*/

{Delivering} = Product • $\overline{Goods\_Deliver}$ • Delivering + Product • $\overline{Sale\_Total}$ • Delivering /*原 Level 1*/

[PO] = Material_Stocking • Null • [PO] /*原 Level 1*/

圖 5：欲整合之 DFD 語意模型

例如，圖 5 為一簡易之接單流程 DFD，用來與圖 2 之 DFD 進行整合。

由以上範例，原 DFD 中 PO 之語意模型為[PO] = Retired-PO-Number • • [PO]，而欲整合 DFD 中 PO 之語意模型為[PO] = Material_Stocking • Null • [PO]，兩者為相同，可加以整合成新語意模型[PO] = (Material_Stocking,Retired-PO-Number) • • [PO]。同時原 DFD 中 Accounting 之語意模型為 Accounting = Check-List • Null • Accounting，而欲整合 DFD 中 Accounting 之語意模型為 Accounting = Sale_Total • Null • Accounting，兩者為相同可加以整合成新語意模型 Accounting = (Sale_Total , Check-List) • Null • Accounting。整合後之完整語意模型如圖 6 所示。

Vendor = Payment • $\overline{Invoice}$ • Vendor

Management = Null • $\overline{Payment - Auth}$ • Management

Accounting = (Sale_Total ,Check-List) • Null • Accounting

Customer = Goods_Deliver • $\overline{Order}$ • Customer

{Compare_with_PO} = (Invoice,PO-Number) • $\overline{Valid - Invoice}$ • Compare_with_PO

{Check-Invoice-Auth} = Payment-Auth • $\overline{Auth - Approved}$ • Check-Invoice-Auth

{Produce-Payment} = Auth-Approved • $\overline{PO - to - be - Retired}$ • Produce-Payment + Auth-Approved • $\overline{Payment}$ • Produce-Payment + Auth-Approved • $\overline{Check - List}$ • Produce-Payment

{Retire-PO} = PO-to-be-Retired • $\overline{Retired-PO-Number}$ • Retire-PO

[PO] = (Material_Stocking, Retired-PO-Number) • $\overline{PO\text{ - }Number}$ • [PO]

{Order_Analyse} = Order • $\overline{Order\_Detail}$ • Order_Analyse

{Produce} = Order_Detail • $\overline{Product}$ • Produce + Order_Detail • $\overline{Material\_Stocking}$ • Produce

{Delivering} = Product • $\overline{Goods\_Deliver}$ • Delivering + Product • $\overline{Sale\_Total}$ • Delivering

圖 6：整合後之 DFD 語意模型

三、階段三：資訊流轉換至企業流程(Conversion)

本節之目的是將前一節中整合後之 DFD 轉換成現今企業塑模方法論中常被採用之 EPC 表示方法。為方便說明，圖例部分仍採 DFD 之表示方式，以利說明。

Step 1：將階層語意模型中之功能(Function)轉換為 EPC 中之功能(Function)

表示方式	DFD 語意模型	EPC
圖例	Pay-Invoice	Pay-Invoice

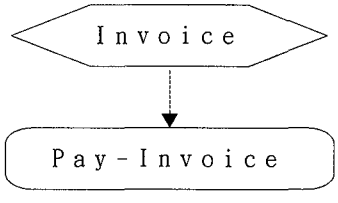
原因：原語意模型中之功能(Function)相對應至 EPC 中即為功能(Function)，可依據預先定義之控制條件做出反應。

Step 2：將語意模型中資料流(Data Flow)轉換為 EPC 中之事件(Event)

表示方式	DFD 語意模型	EPC
圖例	Invoice →	Invoice

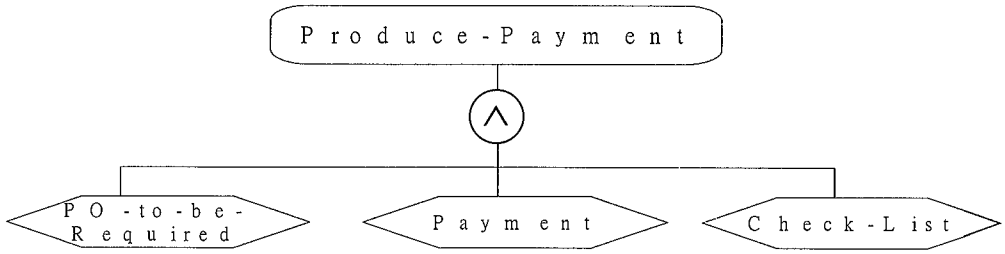
原因：在 EPC 表示方法中，任何流程皆由事件(Event)啟動，為一具有觸發(Trigger)一個或多個功能(Function)作業之主動元件。DFD 中之資料流(Data Flow)雖然傳遞的是資料，但是其意義相對於 EPC 中即為一特定狀態之產生，同時其傳遞之資料可視為被觸發之功能(Function)所需之輸入，故將其轉換為 EPC 中之事件(Event)

Step 3：利用 EPC 中控制流(Control Flow)將 step1 與 step2 中之事件(Event)與功能(Function)相連接，並注意方向及順序。



原因：EPC 中事件與功能間必須相連，不得有單獨存在之情形。

Step 4：若一功能(Function)之產出超過一項，則視情況使用邏輯符號(AND、OR、XOR)將功能(Function)與事件(Event)連接。



原因：一功能(Function)依據設定之規則做出反應後，可依其產出之結果產生不同之事件(Event)，如工廠中品管部門中品管這一個功能(Function)，檢驗結果有合格與不合格兩種事件，依據產生事件之不同，下一步就必須採取不同之功能(即丟棄或出貨)。

Step 5：視需要將語意模型中外部實體(External Entity)轉換為 EPC 中之資訊物件(Information Object)。

表示方式	DFD 語意模型	EPC
圖例	Vendor	Vendor

原因：外部實體(External Entity)之概念相對於 EPC 中即為資訊物件(Information Object)，因為外部實體是界定在系統外部，作為提供系統運作所需資訊之用，故此處以資訊物件為轉換對象。但是在 EPC 中，資訊物件只是一個輔助的元件，當作某個事件或檔案的註釋或說明，並不是一個必備的元件，因此轉換時在 EPC 中可不必表示。

Step 6：檢查語意模型中是否存在之資料儲存體(Data Store)，若有則將之轉換為 EPC 中之檔案(File)。

表示方式	DFD 語意模型	EPC
圖例	PO	PO

原因：DFD 中之資料儲存體(Data Store)相對應於 EPC 中即為檔案(File)，故可直接轉換。雖然在 EPC 表示方法中，主要是以流程為核心，不強制規定必須加進流程圖中，但在本論文中，資料儲存體是整合是一項非常重要的元件，同時為了流程描述的完整性，仍應將其加入。

Step 7：將原本語意模型中與資料儲存體(Data Store)連結之資料流(Data Flow)改為 EPC 中的資訊流(Information Flow)，並與相關之 Function 連接。

表示方式	DFD 語意模型	EPC
圖例	——→	——→

原因：在 EPC 表示方法中，必須對資料進行存取之流程，其連結方式由一般之控制流(Control Flow)改為資訊流(Information Flow)，以之辨別。

例如，前述之整合後 DFD 語意模型經轉換後結果如圖 7 所示：

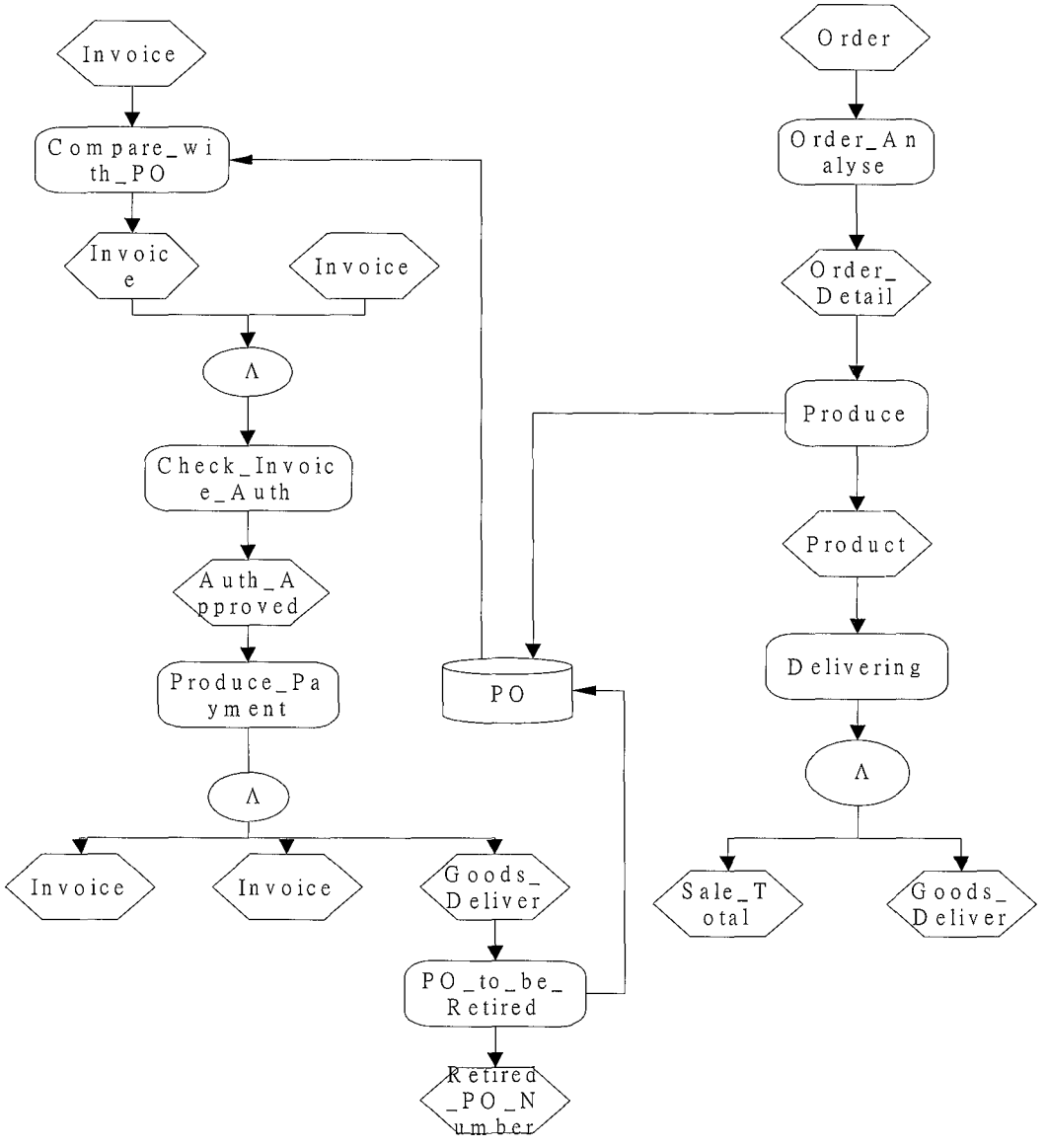


圖 7：整合後之 EPC 流程圖

肆、系統驗證與評估

本論文將以推論方式，證明由 DFD 之語意模型轉換至 EPC 表示方式時，在轉換過程中能夠將原本意涵完整轉換而不至於遺漏。最後本論文中將以一個案進行探討，利用與該個案公司合作企業流程診斷專案之機會，一方面以傳統訪談方式，取得其現行企業流程，另一方面則以本論文中所提出之轉換方法，以該個案公司現行之資訊系統設計資料，進行實地之操作，以評估本論文於實務方面之可行性。

一、理論驗證

不同塑模(Modeling)方法之間轉換的理論驗證上，目前已經有許多研究學者發展出相關理論並加以運用，其中利用資訊承載(Information Capacity)的方法[Miller et al., 1993]，將可以用以驗證在模式轉換過程中，是否能夠達到無失轉換的程度，且此方法已經廣泛運用於模式轉換之相關研究當中[Kwan and Fong, 1999; Huang et al., 2002]。因此本論文中引用此方法，對 DFD 語意模型轉換至 EPC 表示方式時是否有資訊遭到遺漏進行驗證。在其 Miller 幾位學者(1993)的研究中提出五種資訊承載的層級，分別為單正向轉換(Functional)、單逆向轉換(Injective)、群單向轉換(Total)、群逆向轉換(Surjective)、完全雙向轉換(Bijection)，其表描述於下：

1. 單正向轉換

令 A 與 B 為集合，若其間的對應關係滿足如下之規則，即稱為單正向轉換。正規表達如下：

$$f: A \to B, \forall a \in A, \exists b \in B \bullet f(a) = b$$

2. 單逆向轉換

如果 A 與 B 之間的對應關係，相反過來亦滿足單正向轉換，即稱為單逆向轉換。正規表達如下：

$$f^{-1}: B \to A, \forall b \in B, \exists a \in A \bullet f(b) = a$$

3. 群單向轉換

令 S_1 代表原本模式，S_2 代表轉換後之模式，$I(S_n)$ 代表模式 S_n 所有語意之集合。若在 S_n 中每一語意之轉換對應關係都符合單正向轉換，則這樣的轉換關係是群單向轉換關係，也就是說 S_1 可以決定 S_2。正規表達如下：

$$f: I(S_1) \to I(S_2), \forall I(S_1) \in S_1, \exists I(S_2) \in S_2$$

4. 群逆向轉換

如果群單向轉換函式的逆向關係滿足單逆向轉換，則此關係為群逆向轉換。正規表達如下：

$$f^{-1}: I(S_2) \to I(S_1), \forall I(S_2) \in S_2, \exists I(S_1) \in S_1$$

5. 完全雙向轉換

如果一轉換函式能夠同時滿足上述四個條件，則稱為完全雙向轉換，並且能夠確保兩個模式之間的轉換是無失的，不管是正向或者是逆向。正規表達如下：

$$f: I(S1) \rightarrow I(S2), \forall I(S1) \in S1, \exists I(S2) \in S2 \wedge \forall I(S2) \in S2, \exists I(S1) \in S1 \bullet S1 \equiv S2$$

如欲所有資訊在轉換過程中都沒有遺漏，能夠完整的保存，最好能達到完全雙向(Bijection)程度，最少亦需達到群單向轉換(Total) [Kwan and Fong, 1999]。而所謂群單向轉換為只可進行正向轉換即由 DFD 轉至 EPC，若要由 EPC 轉回至 DFD 則可能出現資訊遺失的情形，由於本研究為由 DFD 轉至 EPC 之研究，並不考慮由 EPC 轉至 DFD，因而只要達到群單向轉換即可驗證本研究的機制。

在本機制中，Step 1、Step 5、Step 6 與 Step 7 因元件間之意義完全一致，屬於一對一之轉換方式，因此屬於完全雙向之無漏失轉換，i.e.符合完全雙向轉換(Bijection)的規則。

Step 2 中之資料流與事件屬於意義不相同之元件，因此無法滿足逆向之轉換，只能滿足群單向(Total)轉換模式，i.e.符合群單向轉換(Total)的規則。

Step 3 與 Step 4 目的在將兩元件相連結，由於元件間之轉換屬一對一轉換，用以相連結之資料流其來源與目的元件亦相同，因此亦屬完全雙向之無漏失轉換，i.e.符合完全雙向轉換(Bijection)的規則。

由此可知本論文中除 Step 2 屬於僅具備群單向(Total)轉換之外，其餘轉換規則均屬於完全雙向(Bijection)轉換。依據 Information Capacity 理論，在驗證時若為雙向的轉換則需達到完全雙向轉換(Bijection)的層級;若為單向的轉換則需達到群單向轉換(Total) 的層級。由於本研究為單向的轉換即由 DFD 轉成 EPC，因此只要驗證達到群單向轉換(Total)的層級，依據此理論即可說明本研究所提出的機制在轉換的過程中資訊無失。

二、個案探討

個案公司為一冷凍設備產品製造商，產品包括一般餐飲業營業用冷凍櫃與特殊用途冷凍櫃。該公司雖已通過 ISO9002 認證，但為了提昇公司的經營品質，該公司希望經由企業資源規劃的制度建立，來強化內部管理，以為該公司未來的事業擴充與發展做準備。我們將利用本論文所提出之 IFBP 機制，實際進行於個案公司中，由其資訊系統的角度去發掘其現有的作業流程(AS-IS Model)，作為實驗組[William et al., 1997]。

而個案公司目前進行中的企業資源規劃前的企業診斷工作，其用傳統的訪談方式進行流程探索，以為對照組，我們將比對此二者的結果作為本論文的績效評估，並做為未來發展之參考。本論文中因人力與時間因素，僅針對個案公司目前現行之訂單管理與收款流程為本論文之案例，進行討論。並亦相對應之銷貨訂單管理系統、應收帳款管理系統等兩子系統以 IFBP 機制進行整合，並將結果與其通過 ISO9002 認證之標準作業流程(SOP)文件比較。

IFBP 機制整合後之 EPC 表示方式如附錄一所示，而根據該個案公司於民國八十九年通過 ISO 9002 之標準作業程序之文件，該個案公司對接單作業流程如附錄二所示。在

該個案公司的業務接單標準作業流程文件中，規定了業務人員從拜訪客戶到合約簽訂，以及最後交貨收款的流程，不過綜而觀之，在這份標準流程文件中所定義的流程是屬於概念層的流程設計，對於許多較為細節的作業流程並不能非常完整的表達出來，以上表中押標金的申請來說，該向誰申請、申請表單為何、由誰簽核等等問題都不能在業務接單標準流程文件中表達出來，必須要查詢其它文件才能得知，一個流程被切成數段，無法整合的結果，將造成流程中的瓶頸點無法找出，流程再造自然也無法成功執行。

反觀由本論文提出轉換的機制所得到的流程，雖然礙於人力與時間，只有將銷貨訂單管理系統與應收帳款管理系統兩個系統作整合，但是與該個案公司之標準作業流程文件相比，在同樣的訂單管理與應收帳款流程部分，明顯的比標準作業流程文件中的流程來的詳細，流程中必須使用到的單據亦較為清楚，一旦將企業內所有之資訊系統之 DFD 設計資料整合，所得出之結果勢必比傳統流程探索方式所得到之流程圖更為詳盡、完整。

而在效率部分，流程整合之程序，因為是由電腦輔助系統執行，在現今個人電腦強大的運算能力之下，從 DFD 繪製到轉換成 EPC 表示方法為止，僅約需半個小時，本機制的雛形系統如圖 8 所示：與傳統訪談方式須耗時數天相比，的確能節省相當多之流程探索時間。以上比較的結果，可整理成表 6。

表 6：本論文所提整合方式與傳統流程探索方式之比較

	傳統流程探索方式	本論文所提整合方式
詳細程度	低	高
完整性	低	高
整合度	低	高
正確性	視顧問經歷而定	由現有流程轉換，正確性高且穩定
時間	長	短
效率	低	高

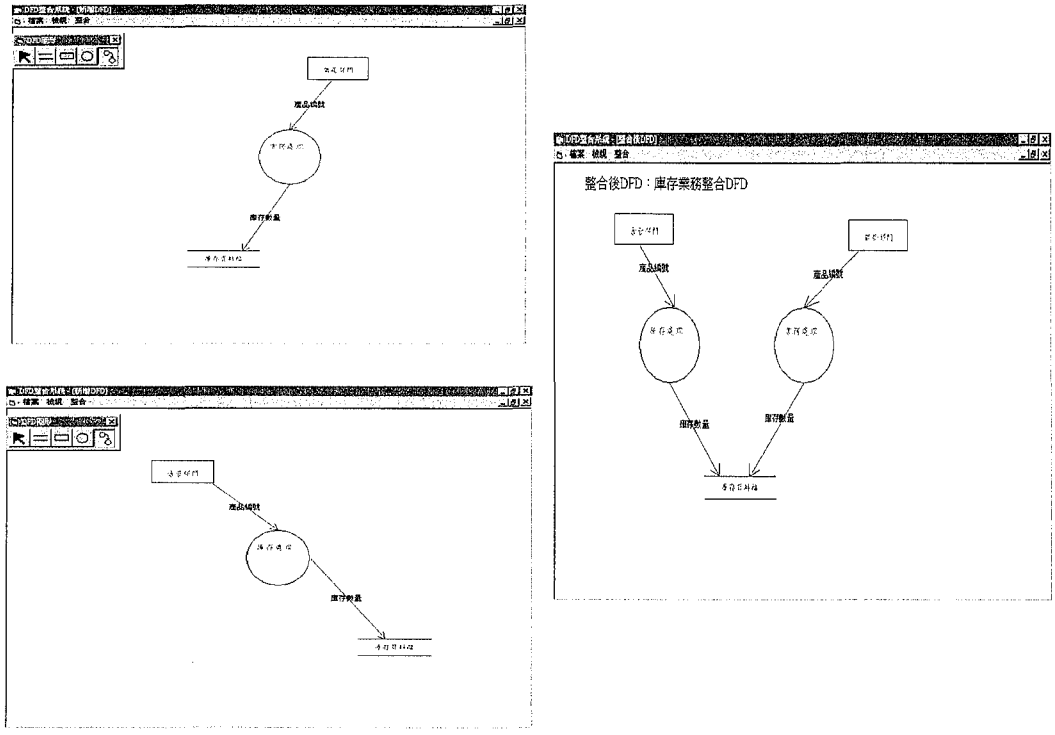


圖 8：IFBP 雛型系統-DFD 功能模組

伍、結論

近年來經濟環境的遽變，致使企業必須不斷力行各種合理化與精簡組織，並同時利用資訊科技的應用，將以往追求的改進作業效率改變成增加管理效能，以提升整體競爭力，增加公司營收與獲利。因此越來越多企業開始導入 ERP 系統，希望藉助其強大整合能力，進行企業流程再造，將企業內各項資源加以整合，替企業創造競爭優勢。但 ERP 的導入所需投資的金額極高，因而如何成功的導入 ERP 對企業而言是一件極為重要的事情問題。依據多位學者的研究[Fong and Huang, 1997; Bingi et al., 1999l Hong and Kim, 2002] ，企業流程再造的失敗是造成 ERP 導入專案失敗的一重要的關鍵因素。而企業流程再造的失敗主要的關鍵因素之一乃為對原作業流程診斷的失真，本研究的成果恰可對此問題提供一良好的解決方法。

本論文中提出之由現行系統中進行企業流程探索方式，可以有效加快這段流程探索的時間，在現今分秒必爭的時代，時間更是企業營運中追求的另一項重要指標。對專業顧問來說，不但可以加快流程診斷的時間，更可以去除不同產業中之不同領域知識對顧問流程診斷時影響的變數[William and Teng, 1995; Orman, 1998, Kliem, 2000]。

此外，對流程設計人員來說，由於對不同應用系統之流程塑模方式的不瞭解，加上，常常需要額外且長時間的學習與轉突。本研究不在於對 ERP 導入提出一套完整解決方

案，而是希望能解決異質系統流程間整合，以及流程設計分歧之問題。故本論文中提出的流程探索方式並不需要非常高深的技術能力，只要依照現行系統設計規格書中 DFD 圖繪製，經本系統處理後即可得到企業目前運作流程，供企業作為自行改善與整合流程之用。

在研究限制上，由於本研究的整合轉換機制採用規則基礎(Rule-Based)作為轉換基準，只能判斷原始資料的語法對錯，無法判斷其品質高低。如原始 DFD 圖所採用了字義太過模糊或是意義太過廣泛的字眼，如此將會影響到轉換出來的流程品質。因此未來若能將本整合轉換機制延伸，將企業中已有的資料庫資料一併加入，在原始設計資料之意涵太過模糊時，可以資料庫中的 ER Model 輔助，進行交叉比對，相信對於上述的問題應該能有相當程度之助益。另外就實務上而言，有些企業組織的資訊系統可能沒有完整的 DFD 圖、甚至沒有 DFD 圖，此情況將可能造成本研究的成果無法適用，還好目前的資訊系統反向工程技術已有顯著的進步能將程式碼轉換成資訊流程如 DFD [Fong and Hung, 1997]，又 DFD 圖對資訊人員而言較為熟悉，亦可由其來製作現存系統的 DFD 圖，如此或可讓資訊人員積極參與企業流程管理專案，提高專案的成功機率[William and Teng, 1995]。

參考文獻

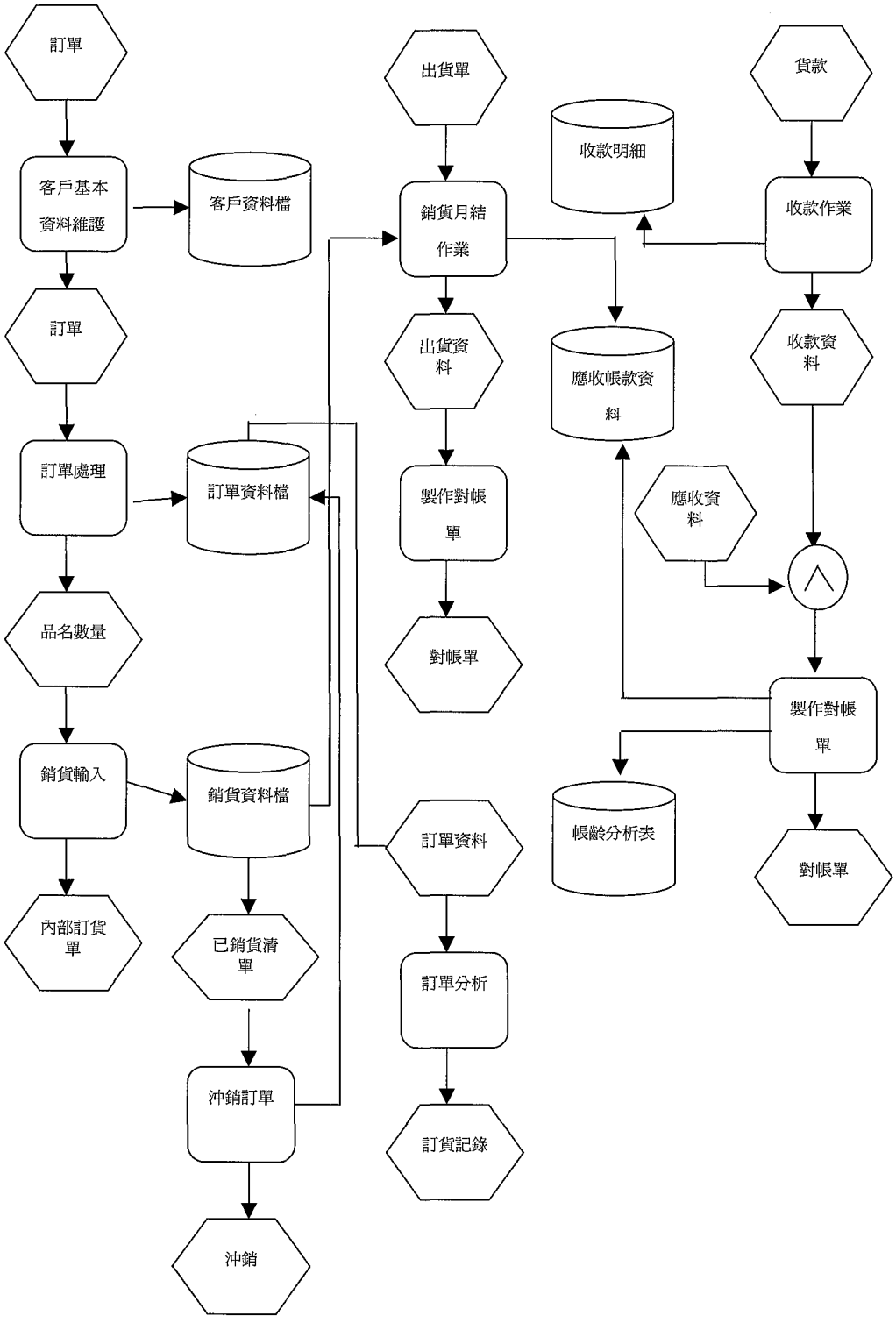
1. Basu, A. and Kumar, A., "Research Commentary: Workflow Management Issues in e-Business", *Information Systems Research*, 13(1), 2002, pp. 1-14.
2. Basu, A., and Blanning, R.W., "Workflow Analysis Using Attributed Metagraphs", *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, HICSS, 2001, pp. 3735 -3743.
3. Bingi, P., Sharma, M. K., and Godla, J. K., "Critical issues affecting an ERP implementation", *Information Systems Management*, 15(5)1999, pp.7-14.
4. Butler, G., Grogono, P., Shinghal R., and Tjandra I., "Analyzing the Logical of DataFlow Diagrams in Software Documents," *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Volume: 2 , 1995.
5. Chalmeta, R., Campos, C., and Grangel, R., "References Architectures for Enterprise Integration", *Journal of Systems and Software*, 57(3), 2001, pp. 175-191.
6. Chandrasekaran, S., Miller, J., Silver, G., Arpinar, I. and Sheth, A., "Composition, Performance Analysis and Simulation of Web Services", *Electronic Markets*, 13(2), 2003, pp.18-30.
7. Davenport, T.H. *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press, Boston,1993.
8. Fong, J. and Huang, S. M., *Information Systems Reengineering*, Springer-Verlag Pub., 1997, ISBN981-3083-15-8.

9. Grover, V., Jeong, S. R., Kettinger, W. J., and Teng, T., "The Implementation of Business Process Reengineering", *Journal of Management Information Systems*, 12(1), 1995, pp. 109-144.
10. Hasselbring, W., "Information System Integration", *Communications of ACM*, 43(6), 2000, pp. 32-38.
11. Hong, K. K. and Kim, Y. G., "The Critical Success factors for ERP Implementation: An Organization Fit Perspective", *Information & Management*, 40(1), 2002, pp.25-40.
12. Huang, S. M., Hsueh, H. Y. and Lin, B. "Information Sourcing from Internet and its Verification", *Journal of Computer Information Systems*, 42(3), 2002, pp. 94-105.
13. Huang, S. M., Kwan, I., and Hung, Y. C., "Planning Enterprise Resources by Use of a Reengineering Approach to Build a Global Logistics Management System", *Industrial Management & Data Systems*, 101(9), 2001, pp 483-491.
14. Keller, G., and Teufel, T., *SAP R/3 Process Oriented Implementation: Iterative Process Prototyping*, Addison Wesley, Berlin/Heidelberg, 1998.
15. Kliem, R. L., "Risk Management for Business Process Reengineering Projects", *Information Systems Management*, 17(4), 2000, pp. 71-73.
16. Kwahk, K. Y. and Kim, Y. G., "Supporting business process redesign using cognitive maps", *Decision Support Systems*, 25(2), 1999, pp. 155-178.
17. Kwan, I. and Fong, J., "Schema Integration Methodology and its Verification by use of Information Capacity", *Information Systems*, 24(5), 1999, pp.355-376.
18. Marcus, O., Carsten, H., and Ludwig, N., "Reengineering Organizational Structures from Within," *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences*, HICSS, 1999, pp. 1-10.
19. Miller, R. J., Ioannidis, Y. E. and Ramakrishnan, R., "The use of Information Capacity in Schema Integration and Translation", *Proceedings of the 19th International Conference on Very Large Data Base*, VLDB, Dublin, Ireland, 1993, pp. 120-133.
20. Orman, L.V. "A Model Management Approach to Business Process Reengineering," *Journal of Management Information Systems*, 15(1), 1998, pp. 187-212.
21. Ranganathan, C. and Dhaliwal, J. S., "A survey of business process reengineering practices in Singapore", *Information & Management*, 39(2), 2001, pp. 125-134.
22. Scheer, A. W., *Business process Engineering: Reference Models for Industrial Enterprise*, Second Edition, Springer-Verlag, Berlin/Heidelberg, 1994.
23. Sheth, A. P., van der Aalst, W. M. P., and Arpinar, I. B., "Processes Driving the Networked Economy", *IEEE Concurrency*, 7(3), 1999, pp. 18-31.
24. Smith, H., Neal, D., Ferrara, L., and Hayden, F., "The Emergence of Business Process Management", *CSC Research Service Report*, 2002, available at: http://www.cscresearchservices.com/process/.
25. The Delphi Group, "BPM 2002 Market Milestone Report", 2002, available at:

http://www.delphigroup.com/research/whitepapers.htm.

26. The Gartner Group Inc., "The BPA Market Catches Another Major Updraft", Gartner's Application Development & Maintenance Research Note M-16-8153, 12 June 2002, available at: http://www.gartner.com/gc/webletter/idsscheer/issue1/article1.html.
27. Themistocleous, M., Irani, Z., O'Keefe, R. M., and Paul, R., "ERP problems and application integration issues: an empirical survey", *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, HICSS, 2001, pp.3775 -3784.
28. van der Aalst, W. M. P., "Formalization and Verification of Event-Driven Process Chains", *Information and Software Technology*, 41(10), 1999, pp. 639-650.
29. van der Aalst, W. M. P., "Loosely Coupled Interorganizational Workflows: Modeling and Analyzing Workflows Crossing Organizational Boundaries", *Information & Management*, 37(2), 2000, pp. 67-75.
30. van der Aalst, W. M. P., "The Application of Petri Nets to Workflow Management", *The Journal of Circuits, Systems and Computers*, 8(1), 1998, pp. 21-66.
31. White, S. A., "Business Process Modeling Notation", *BPMI.org*, Draft 0.9, 2002, available at http://www.bpmi.org.
32. Whitten, J.L. and Bentley, L.D., *System Analysis and Design Methods*, McGraw Hill, New York, 1999.
33. William, J. K., and Teng, J. T., "Business Process Redesign and Information Architecture: Exploring the Relationships," Data Base Advances, 26(1), 1995, pp. 30-42.
34. William, J. K., Teng, J. T. C., and Guha, S., "Business Process Change: A study of Methodologies, Technologies, and Tools", *MIS Quarterly*, 21(1), 1997, pp. 55-80.

附錄一：訂單管理與收款流程整合後之 EPC 流程圖



附錄二： 個案公司業務接單 ISO 標準作業流程