

# 由 MIS 系統至 ERP 系統之資料導向移轉方法

(A Data-Oriented Migration Approach to Converting Legacy MIS into ERP System)

楊欣哲 (Shin-Jer Yang)、徐坤琳 (Kun-Lin Hsu)

東吳大學資訊科學系

Email: sjyang@cis.scu.edu.tw

## 摘要

由於網際網路技術成熟及其應用普及，因此傳統 MIS 移轉至 Web-Based ERP 系統(亦稱為新目標系統)成為資訊架構改造的首要目標之一。本文提出資料導向之系統移轉方法：策略-資料-功能(SDF: Strategy-Data-Function)，將針對如何從傳統架構的 MIS 系統移轉到 Web-Based 的 ERP 架構，以及能將企業既有的流程及資料完整地移轉到新目標系統的程序，首先從企業發展的策略面思考新目標系統的整體架構，再由傳統 MIS 系統及新目標系統的資料面進行整合之考量，二個系統間須建置資料移轉器以轉換傳統系統的資料到新目標系統的資料庫，最後採用新的技術架構開發新目標系統的各種功能。SDF 的方法較既有的系統移轉方法其轉換的程序更為順暢，而且新目標系統功能符合企業日後的需求，同時也保存了企業最重要的營運資料。

本文中 SDF 方法的各階段先以 UML 類別圖的設計方式說明之，再以虛擬碼(Pseudo Code)來描述執行程序。並和傳統上較常使用的方法依其功能作關鍵績效指標評估比較並加以分析，以證實 SDF 方法的效果與效率。由於目前系統移轉相關的研究並未提出更新的作法，因此 SDF 方法對企業所迫切需要的系統移轉專案提供一個更好的選擇。未來，SDF 方法於企業資訊策略分析中的策略行動方案中可再深入的探討，以提出更有效率的系統移轉程序。另外，可考慮進行傳統系統及新目標系統之間的 DMP 資料移轉器之細部設計方式，以提高資料移轉的完整性及一致性。

**關鍵詞：**ERP、資料導向移轉、SDF、關鍵績效指標

## ABSTRACT

Owing to advanced technologies and popular applications of Internet, the converting legacy MIS into Web-based ERP system (called Target system) is a key issue for revolution of information infrastructure. In perspectives of business development strategy and database migrations between legacy MIS and Web-based ERP system, this paper will propose data-oriented SDF (Strategy-Data-Function) methodology to resolve the common problems of current system migration and to migrate full procedures and all data of business organizations from legacy MIS into Target system. In

order to successfully converting a legacy MIS into a Target system, an effective SDF approach can fully migrate the required information and knowledge that resides in the legacy MIS to Target system.

First, this paper utilize UML class diagram to illustrate the design issue of each stage in SDF and also design a pseudo code to illustrate the operation procedure for each stage. Then, we compare the KPI among migration methods based on their features to verify the effectiveness and efficiency of SDF. Currently, there are not any other approaches to system migration. Hence, SDF approach is one of best choices for organization's system migration. In the future, we can further enhance the SDF functions to propose the more efficient procedure of system migration in the business information strategy. In addition, we can perform the detailed design of DMP data transformer between legacy MIS and Target system to improve the integrity and consistency of system migration.

**Keywords:** ERP, Data-oriented Migration, SDF, KPI

## 一、簡介

網際網路技術成熟和資訊科技之快速發展，促使企業推動 e-化的成效，已經成為企業未來競爭優勢的關鍵因素。而電子商務的應用普及化，更突破了企業間的藩籬，使得企業間的資訊能更快速且有效地傳遞與溝通。面對一個新的趨勢，企業 e-化的範圍不單只是侷限在企業內的資訊系統及資訊技術的應用，還擴及到企業合作夥伴的企業間 e-化。所以，企業 e-化的成效，必須是企業內部 e-化成效加上企業間 e-化成效的評估方式，才能成為企業未來競爭優勢的關鍵成功因素[1]。

目前傳統系統主要是以管理資訊系統 (MIS: Management Information System)為企業的基礎核心，但在企業 e-化的推動中，必須具備跨國界、以網際網路的系統環境，或稱為 Web-Based 環境。基礎上再持續整合其他新一代企業間的資訊系統，才能達到全球運籌的作業目標。根據資策會的調查分析報告可清楚的表達此發展趨勢，從 2002 到 2006 年我國 e-Business 市場規模中的 ERP 系統成長率由 8.8%降到 1.0%，而 CRM、SCM 等新一代企業間的資訊系統的市場規模則持續成長[2]。但是許多企業的 MIS 系統可能是較傳統的架構，如集中式或二層式 Client/Server 等架構，無法滿足新的需求。因此，許多企業必須將其行之有年傳統架構的 MIS 系統移轉到新的 Web-Based 架構上。

但是 Web-Based ERP 系統(或稱為新目標系統)於開發技術或導入等方面都遠較傳統的系統架構更為複雜,再加上傳統資料檔案移轉到資料庫的整合難度,使得這個移轉過程變成一種艱巨的工程[20]。目前較被廣泛採用的方法包括架設閘道(Gateway)中間介面、大霹靂方法及蝴蝶法等。但是架設閘道實務的作法上需考慮相當多的因素,所以不太容易實作;大霹靂方法則較適用於小型系統,但對大型系統而言,風險及失敗率都很高,需要慎重的考慮;蝴蝶法基本上類似 Wrapping 的方式,在傳統系統與新目標系統間架設資料移轉程序,但移轉後的系統並沒有規劃到未來企業間資訊系統整合需求,因此新目標系統基本上只是傳統系統的更新版本。

這些常用的方法和企業資源規劃系統的移轉目標稍有不足之處,最主要的原因是沒有考量到企業未來發展的因素,而較著重在技術方面的要素,所以移轉後的系統基本上還是傳統系統的翻版。因此本文將針對如何從傳統系統架構移轉到 Web-Based 架構,以及能將企業既有流程及資料完整的移轉到新目標系統的過程,由企業發展策略引導資料庫移轉方法,到開發新目標系統的過程,提出一套 SDF (Strategy-Data-Function)方法的架構,並將 SDF 各階段以 UML (Unify Modeling Language)類別圖(Class Diagram)的方式將各階段的執行程序以說明設計流程,再以虛擬碼(Pseudo Code)的方式來描述執行程序,並和傳統較常使用的方法依功能作關鍵績效指標(KPI: Key Performance Index)評估比較並加以分析,以證實 SDF 方法的效果與效率。

本文的架構分為五章:第一章為摘要及簡介,概要說明整篇文章的大綱及內容。第二章則著重於文獻探討及相關研究,介紹各種系統移轉的方法,並說明各種方法的優劣,然後再導出本文所提出的 SDF 方法。第三章則詳細的說明 SDF 方法論的設計原理、架構、特性及細部設計說明。第四章以 Bisbal 等人所提出的系統移轉的程序評估分析方法加以改良後,使用 KPI 來比較各系統移轉方法的優缺點並加以分析。第五章作出結論並說明未來研究的方向。

## 二、文獻探討與相關研究

傳統系統移轉的程序包含了許多技術領域的研究,以一個移轉計畫而言,包括了反向工程、企業流程再造、資料結構的映對及轉換、資料的實際移轉、新目標系統的開發、人機介面整合及測試等不同技術領域的工作[8]。除技術上的考量,還包括其他非技術性的因素考量,如開發及建置的成本、時程的安排、品質的控管、風險的控制及系統的效率等因素[18]。本文的研究方向只說明與移轉方法有關的相關研究。目前較常採用的方法為在傳統系統外面開發一組介面稱為"Wrapping",它將傳統系統的資料、程式、應用系統及介面等都以新的

介面包裝起來,使得傳統系統呈現新的外貌及更好的操作介面[16]。

Aebi 從企業資料再造工程的角度提出了透過一中介系統(Intermediate System)來進行系統的移轉的模型方法,其模型包含二個部份:一部份是探索程序(Exploration Process),以確認移轉程序符合預期目標;另一部份是實施程序(Effectively Process)分為 5 個階段:問題定義、問題分析、開發中介系統、導入中介系統及開發新目標系統[6],其模型如圖 1 所示。

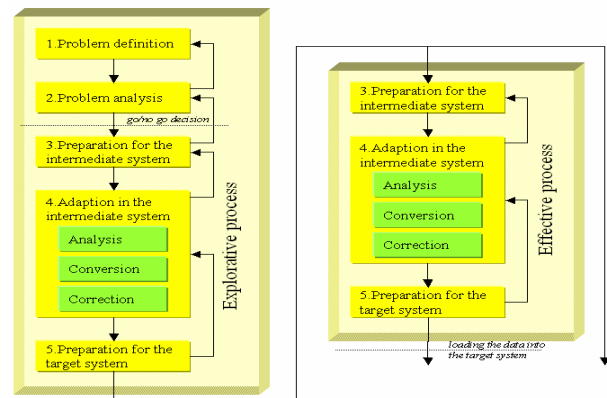


圖1. Aebi 資料再造工程系統移轉模型

較早提出的系統移轉方法有大霹靂方法(Big Bang Approach)將傳統系統以新的架構、工具、資料庫重新開發,並在新的硬體平台上執行。此外資料庫優先方法(Database First Approach)即是以資料庫為優先考量的系統移轉方法。資料庫最後方法(Database Last Approach)則以資料庫為最後考量的系統移轉方法[5]。而混合式資料庫方法(Composite Database Approach)則以前兩個方法混合考量的系統移轉方法[14]。上述這三種方法基本上採用傳統系統與新目標系統併存,並以閘道(Gateway)為中間介面來傳遞兩系統間的訊息。細微遞增策略(Chicken Little Strategy)改良了混合式資料庫的方法,但基本上仍是採用介面閘道(Interface Gateway)來控管傳統系統與新目標系統間傳遞的訊息[15]。

蝴蝶法(Butterfly Methodology)則提出不使用閘道的方法,於系統移轉的過程中降低傳統系統與新目標系統的關連,仍能保持兩系統間資料的一致性。此方法所採用軟體元件設計的概念,藉由 Data-Access-Allocator (DAA)控制傳統系統的資料存取動作,並將資料儲存到一系列的暫存區(Datastores),再透過資料換器(Chrysaliser)將暫存區的資料移轉到新目標系統的資料庫,此過程會執行直到暫存區的資料不需要再被移轉為止,其架構如圖 2 所示[9]。蝴蝶法包含六個階段:準備階段、了解傳統系統並開發新目標系統的資料庫階段、依據新目標系統資料結構建立資料暫存區階段、將傳統系統的元件(除資料外)移轉到新目標系統階段、逐漸地移轉傳統系統資料到新目標系統並教育訓練使用者階段、切換至新目標系統階段。

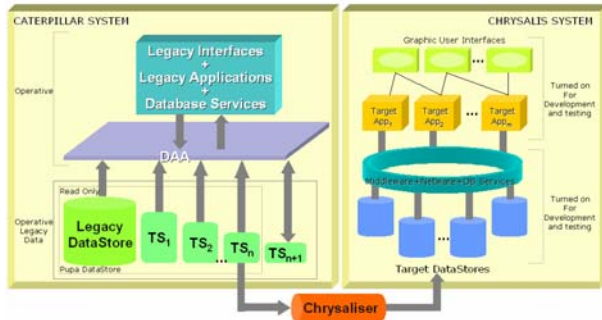


圖2. 蝴蝶法，由 TempStore TSn 移轉資料

Teta 等人提出於整個系統移轉專案的組成因素架構及監督的機制，並定義系統移轉方法是『系統移轉是一種程序，它包含了企業的變遷，其內容比將組織的資料庫系統、應用程式及程式介面的移轉來得更多』。到目前為止，各研究學者只針對系統移轉的程序提出多種方法論或技術上的作法，但是對於整個專案的組成因素及各階段互相影響的因子甚少提及。因此 Teta 等人將系統移轉專案的組成分為三個概念子模型，分別是意圖子模型(Intentional Submodel)、資訊子模型(Informational Submodel)及作業子模型(Operational Submodel)，如圖 3 所示。在每個子模型中，各有控制點，分別為移轉目標、移轉程序、移轉情境、系統移轉及 Legacy 系統，每個控制點中包含了影響此控制點的因子，並採用 AND/OR 的邏輯運算來組合各因子的階層關係。這個模型提供了整個系統移轉專案的概觀，使得執行移轉專案的團隊能掌握過程中之各種因素的關係[19]。

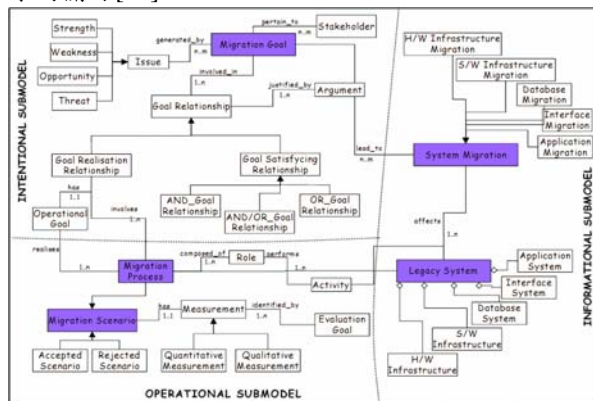


圖3. Teta 系統移轉方法的 Meta-Model

蔡春智提出了以使用者觀點為主以進行系統移轉的方法分為 6 個步驟，如圖 4 所示。以使用者觀點的系統移轉方法描述了移轉過程的細部程序，著重於每個動作的說明[3]。但是因為是以傳統系統的使用者操作介面為優先的考量因素，和 Teta 等人提出系統移轉方法的定義有一些不同。使用者導向之系統移轉方法第 1 步驟為建構來源系統操作流程並定義系統畫面欄位，主要進行來源系統使用者需求收集、繪製來源系統流程圖、定義來源系統畫面欄位及確認來源系統流程與系統畫面定義完整等程序；第 2 步驟為建構目標系統操作流

程並定義系統畫面欄位，主要進行目標系統使用者需求收集、繪製目標系統流程圖、定義目標系統畫面欄位、及確認目標系統流程與系統畫面定義完整等程序；第 3 步驟為比對來源與目標系統流程與畫面欄位，主要進行找出對應流程、找出對應系統畫面欄位、確認系統畫面欄位對應完成等程序；第 4 步驟為找出與來源系統畫面對應之資料庫欄位，主要進行新目標系統及傳統系統資料庫定義及對應等程序；第 5 步驟為建構來源系統資料庫模式，主要進行來源系統資料庫反向工程、物件角色建模、歸納企業物件及確認資料已分析完成等程序；第 6 步驟為進行系統移轉，主要進行進行資料移轉及確認系統移轉成功並切換系統等程序。

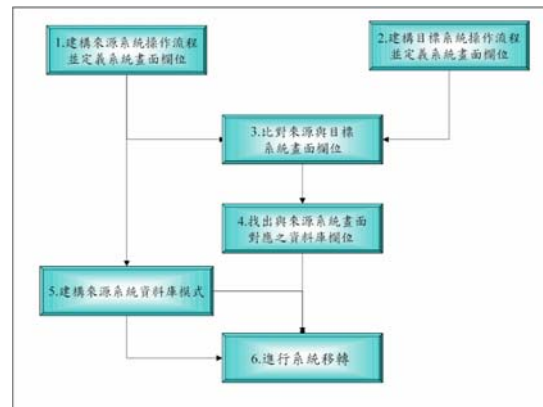


圖4. 使用者導向的系統移轉方法

總之，各移轉方法皆有其優劣，說明如後。開發移轉介面 Wrapping 的方法雖然能快速地將系統移轉，而且降低了使用者轉換的成本，但是只有改變傳統系統的外觀，內部的結構仍保留原有的樣式，難以整合未來企業間的資訊系統。Aebi 的模型雖然能較穩定且安全地進行系統移轉的程序，但是移轉後的系統沒有規劃到未來企業間資訊系統整合需求。大霹靂方法較適用於小型系統，但對大型系統而言，風險及失敗率都很高。資料庫優先方法、資料庫最後方法、混合式資料庫方法、細微遞增策略等四種方法的開道(Gateway)中間介面於實務上的作法須考慮相當多的因素，所以不太容易實作。蝴蝶法基本上類似 Wrapping 的方式，在傳統系統與目標系統間架設資料移轉程序，但移轉後的系統沒有規劃到未來企業間資訊系統整合需求。Teta 的模型提供了整個系統移轉專案的概觀，使得執行移轉專案的團隊能掌握過程中的各種因素的關係。但是如何針對各因素的變化及控管尚未提及。蔡春智提出的使用者導向之系統移轉方法以傳統系統的使用者操作介面為優先的考量因素，對於企業的變遷因素並沒有考慮進來。由於各研究方向和 ERP 系統的移轉目標稍有不足之處，最主要的原因皆未考量到企業未來發展的因素，而較著重在技術方面的要素，所以移轉後的系統基本上還是為傳統系統的翻版。因此，本文將提出一個較新的方法 SDF 以整合企業移轉系統的策略及移轉的程序。

### 三、資料導向之系統移轉方法：SDF

本研究提出之系統移轉方法稱為『策略-資料-功能(Strategy-Data-Function)，或稱為 SDF 方法』，顧名思義，先從企業發展的策略面思考新目標系統的整體架構，不單只是企業內部未來的需求，還包括了企業間的連線需求。再由傳統系統及新目標系統的資料面進行整合之考量，二個系統間須建置資料移轉器以移轉傳統系統的資料到新目標系統的資料庫，由於新目標系統的資料庫架構應大於傳統系統的資料結構，因此資料移轉器還應包括資料補修之功能。最後採用新的技術架構開發新目標系統各種功能，由於新目標系統定位在 Web-Based 架構，所以人機介面和傳統系統有相當大的差異，而商業處理規則可參考傳統系統，或者適當的調整後套用之。

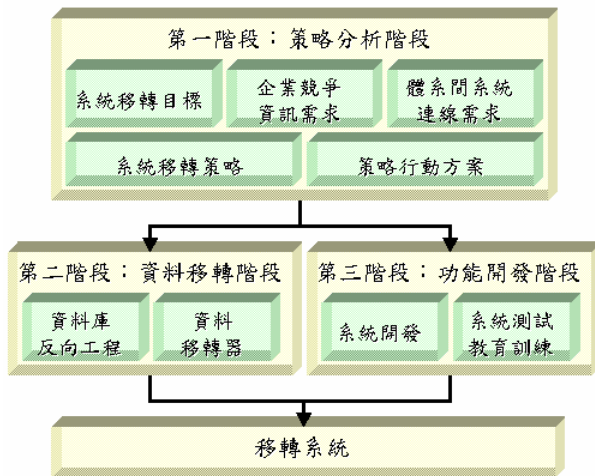


圖5. SDF 的系統移轉方法

SDF 方法的架構如圖 5，分為三個階段。第一個階段為系統移轉策略分析階段，由系統移轉目標、企業競爭資訊需求及體系間連線需求三個方面進行系統移轉策略的分析，分析結果採用策略地圖描述之，以形成系統移轉策略，並以關鍵績效指標(KPI, Key Performance Index)作為檢核的依據，再依據策略擬定策略行動方案，以定義系統移轉各步驟的目標及程序。策略行動方案中分為四大部份，分別是技術構面、資料構面、功能構面及整合構面。技術構面規劃新目標系統在資料庫、系統開發、網際網路及企業間系統連線各方面的技術考量。資料構面為 SDF 方法的第二個階段，依據執行策略規劃新目標系統的資料結構，並分析與傳統系統間的差異，差異部份則開發資料移轉器以補修傳統系統的資料，並定期移轉傳統系統的資料到新目標系統的新資料暫存區中，於新資料暫存區中的資料則利用資料庫的工具及資料庫操作語法以檢測新資料輸出結果和傳統系統的資料輸出結果是否一致，以確認資料移轉的正確性。功能構面及整合構面為 SDF 方法的第三個階段，功能構面包括開發人機介面、建置商業規則、系統整合測試及使用教育訓練等程序，其中開發人機介面採用技術

構面中規劃的開發技術，而且由於 Web-Based 系統的特性，因此可和資料移轉階段同時進行。在第三個階段當已完成系統整合測試及使用者教育訓練後，因為資料在第二階段已進行移轉，則可直接切換到新目標系統。整合構面考量的是企業內部其他系統及企業間系統的整合，也可以說是新目標系統的擴充階段。SDF 的方法程序如圖 6。

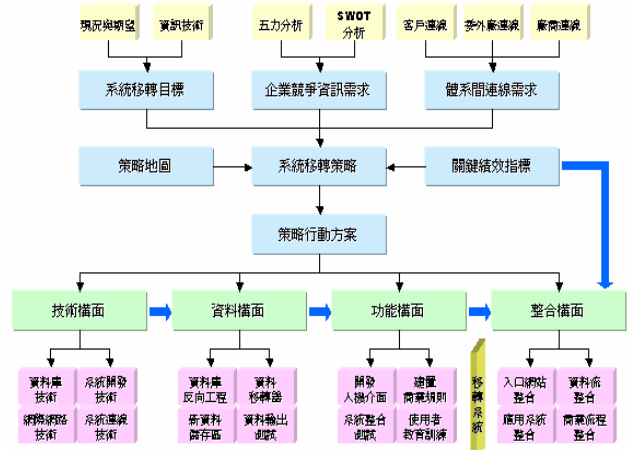


圖6. SDF 的系統移轉架構

在策略分析階段中首先要從三個角度了解企業未來的改變，分別是系統移轉目標、企業競爭資訊需求及體系間連線需求。於系統移轉目標中，由現況與期望及資訊技術的方面了解傳統系統的目前現況(AS-IS)及未來期望的模式(TO-BE)，並以情境模式來說明現況與未來的差異，再以資訊技術的方面分析未來模式的可行性，所分析的結果以記錄、收集、彙總、分析與呈現的分類方式整理之。

於企業競爭資訊需求中，由五力分析的方法分析企業於外在環境變化時的競爭策略，以及由 SWOT 分析的方法針對資訊需求的角度了解組織及環境變化時的影響及因應策略，依此結果比對及修正系統移轉目標程序所得的資訊。在 Porter 提出五力分析為：「一個產業所面臨的競爭情勢將視五種作用力的影響而決定」，這五種作用力分別為同業競爭者、潛在競爭者、替代品威脅、廠商議價能力及客戶議價能力。企業決策者可針對這五種作用力提出競爭策略以因應之，使得企業能獲得較目前更有利的競爭地位。分析者從這些策略中找出資訊支援的需求，並依據策略的分類分別提出資訊需求的規劃[17]。SWOT 分析是企業策略研究中一種常用的分析工具，為優勢(Strength)、劣勢(Weakness)、機會(Opportunity)及威脅(Threat)四字簡稱。按照企業競爭策略的完整概念，策略應是一個企業「能夠做的」(即組織的強項和弱項)和「可能做的」(即環境的機會和威脅)之間的動態組合。SWOT 分析從一開始就具有顯著的結構化和系統性的特徵。於內容上 SWOT 分析法的主要理論基礎強調從結構分析著手，對企業的外部環境和內部資源進行分析。本文的 SWOT 著重於企業資訊系統的分析，釐清企業資訊系統的發展策略，並和五

力分析的資訊策略規劃結論互相映對，以確認資訊策略的可行性。

體系間連線需求是近年來企業間 e-化逐漸發展所衍生出的需求，體系間連線指的是在同一個垂直體系間，上中下游各企業的主要系統以各種技術互相連結在一起，讓不同企業不同系統間可以互相拋轉交易的資訊，以加強資訊的透通度及加快企業反應時間。連線分為客戶連線、委外廠商連線及廠商連線三種類型。這類的需求是近年來政府在補助企業發展 e-化資訊應用時所強調的一個重點，補助計畫中要求企業發展 e-化時須涵蓋體系間成員的系統，即連線成果為審查的重點[4]。連線的方式一般而言有四種方式，AP to AP、Turnkey、Browser 及 Mail Parser，都是透過網際網路進行連線。當許多的企業獲得政府 e-化補助款後的成果是許多不同系統間的連線，這個機制將成為企業間進行商業行為的必要因素，因此這個因素也應在進行新目標系統開發時同時被考慮進來。從這三個角度所彙總的資訊策略，成為系統移轉策略，因此策略除了考慮到傳統系統的因素外，還加上了企業未來發展的因素及擴充的能力。為了讓移轉策略能被充份的執行，採用策略地圖的方法來描述策略，讓策略由隱性概念轉化為顯性目標，以利後續的執行。此外，關鍵績效指標的訂定是為了檢驗策略執行的成果，同時也是系統移轉後所欲達成的目標。指標的定義需視企業發展的策略及方向，它可以是營業額的提高、企業內部處理商業行為的時間縮短或是工作量的增加等方向，指標的定義是要可預期達到的，而非只是一個數值。關鍵績效指標將會用來檢驗策略的可行性及新目標系統完成後的達成率，以評估新目標系統的績效。

策略指出的是是一個方向及目標，策略要能準確的執行依靠的是策略行動方案，將執行策略的程序、步驟及方式逐步規劃，使得執行策略的人員能有依循的標準。策略行動方案分為四個構面，分別是技術構面、資料構面、功能構面及整合構面，技術構面由資料庫技術、系統開發技術、網際網路技術及企業間系統連線技術所組成，主要範圍在探討各種技術間的可能性及可執行評估，以決定新目標系統採用何種技術進行開發。資料構面則為資料移轉階段的執行重點，功能構面及整合構面則為功能開發階段的執行重點。其中資料移轉階段及功能開發階段可以同時執行，當完成系統功能開發、系統測試及使用者教育訓練，以及資料移轉後，即可進行系統移轉到新目標系統。資料移轉階段在 SDF 方法中佔有很重要的地位，傳統系統的資料大都是傳統的索引檔案，例如 COBOL 或是 CLIPPER 的資料檔案.DAT 或是.DBF，沒有完整的資料結構資訊，而且一般都是以文件的方式儲存或是直接撰寫在程式的備註中。這類的資訊將採用資料庫反向工程(Database Reverse Engineering)的方式萃取出資料結構的資訊，並依據移轉策略的定義，重新再製後，移轉為新目標系統的資料結構。

在 Jean 的論文中指出資料庫反向工程主要處理資料庫結構的抽取、分析及轉換，正向工程(Forward Engineering)從使用者的需求、資料庫概念的設計、資料庫邏輯的設計、資料庫實體的設計到資料庫可執行版本的設計等程序，所以資料庫反向工程是這些程序的『反向(Reverse)』[12]。資料庫反向工程包含了二個程序，一是資料結構的抽取(Data Structure Extraction)，另一個是資料結構的概念化(Data Structure Conceptualization)。Jean 以此架構從傳統系統的資料檔中抽取資料結構，經過整理後，再以資料庫的語法(如 SQL 語法)移轉到新的資料庫結構中。本文將採用此架構的抽取及概念化程序(Extraction and Conceptualization)，為了配合新目標系統移轉策略及物件導向技術的運用，後續移轉的資料庫設計實作將改用 UML 的類別圖來描述新目標系統的資料結構[10]。

因此資料移轉階段分為二個程序，第一個程序是資料結構移轉，其產出為新目標系統的資料庫結構；第二個程序是實體資料移轉，其產出為新目標系統的實體資料庫。方法如圖 7。

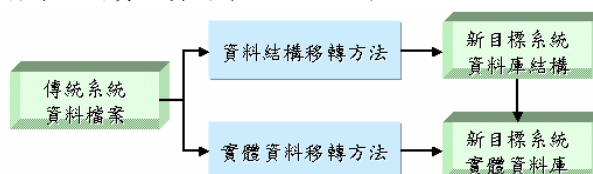


圖7. 資料移轉階段方法

資料結構移轉方法以資料庫反向工程的方法抽取傳統系統的資料概念描述(Data Conceptual Schema)。再以資料庫正向工程的方法，依據策略行動方法中資料構面的規劃，重製並改良傳統系統的資料概念描述，得到新目標系統的資料概念描述，再設計新目標系統的資料邏輯描述(Data Logical Schema)及資料實體描述(Data Physical Schema)，方法如圖 8。

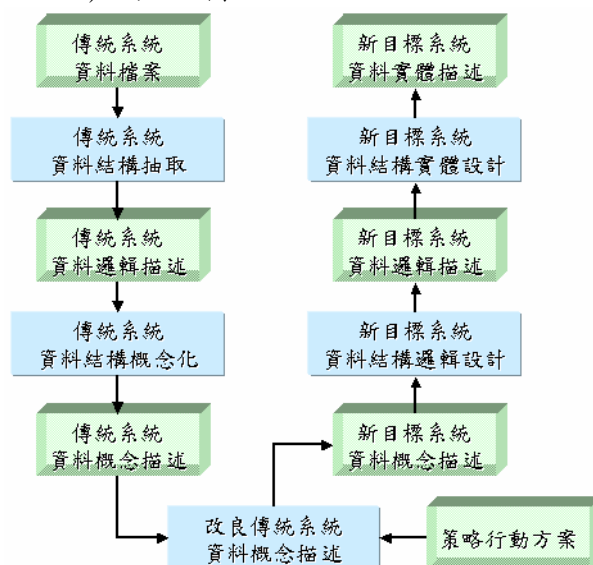


圖8. 資料結構移轉方法

實體資料移轉方法透過資料移轉器(Data Migration Part, DMP)將傳統系統的資料先移轉到新目標系統的資料暫存區，經由資料輸出測試(Data Output Tester, DOT)比對新舊資料的正確性後，再正式移轉到新目標系統的資料庫中。DMP 包含了二個部份，第一個部份是資料移轉規則，主要負責指示新舊資料的映對關係，以便讓 DMP 實作資料的移轉；第二個部份是資料補修介面，主要負責將新資料結構中舊資料無法提供或是缺少的部份補足，這些介面需視不同的資料映對關係而提供不同的補修介面。DOT 分為二個測試的部份，第一個部份是依照傳統系統的資料輸出需求同樣的在新資料暫存區中取得，以比對兩者是否正確；第二個部份是依照新目標系統的資料輸出需求，產生輸出結果，以確認正確性。若兩者資料的正確性有誤，則需重新修正資料移轉規則及資料補修介面後，再進行測試直到無誤，其方法如圖 9 所示。

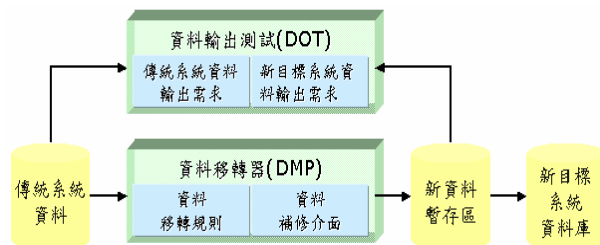


圖9. 實體資料移轉方法

功能開發階段和資料移轉階段可以同步的進行，Amjad 針對一般的應用程式開發過程提出如圖 10 的樣版，分為四個階段，分別是分析(Analysis)、結構(Architecture)、導入(Implementation)及佈建(Deployment)，這個過程會分為四次反覆的執行，每一次執行都會經歷這四個階段[7]，也許各階段的規模在不同的反覆過程中會不同，整個開發過程可能不只四次的反覆執行，也許會更多的次數。

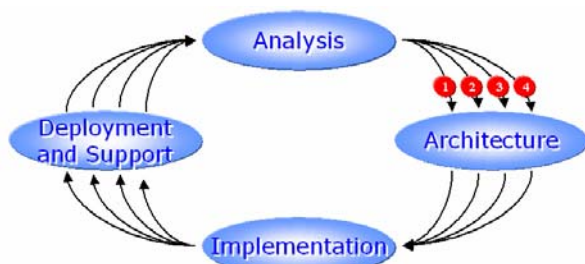


圖10. 一般的應用程式開發過程樣版

Maria 等人提出以框架為基礎重整工程的程序 PARFAIT，以 RUP 靜態結構為基礎，描述了誰/使用者(Who/Workers)，要做什麼事(What/Artifacts)，怎麼做(How/Activities)以及何時做(When/Workflows)。以 RUP 程序的起始(Inception)、細節(Elaboration)、建構(Construction)及移轉(Transition)四個階段使用於一般的重整工程過程活動[13]。在每個階段中還包含了必要的活

動流程，例如在起始的階段，第一個步驟分析系統範圍內的特色及核對系統的框架，第二個步驟觀察傳統系統的特色及系統框架及範圍的歸類，第三個步驟對照傳統非功能性框架的特色。完成這些活動並產出所需的文件後，再進到第二個階段，這些階段是依照遞增及互動(Incremental and Interactive)的特性在進行。各階段的詳細活動說明如圖 11。

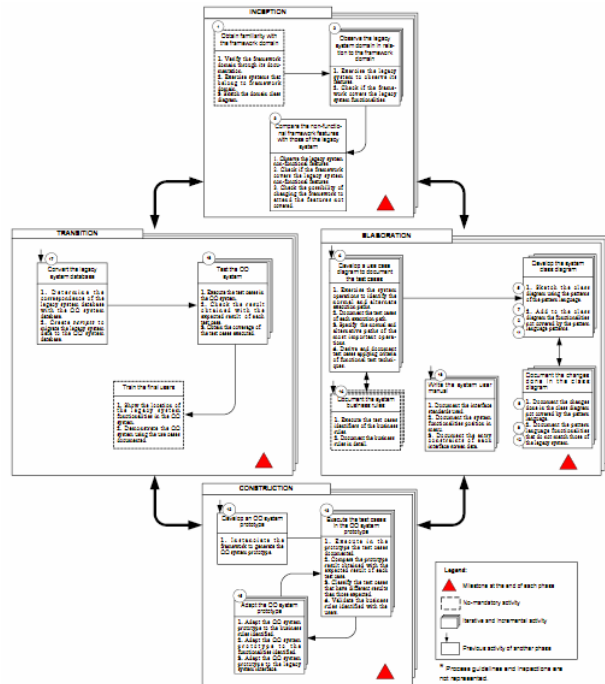


圖11. PARFAIT 重整工程階段活動說明

本研究採用 PARFAIT 的重整工程方法，增加策略分析階段所獲得的策略行動方案為起始階段的第一個步驟，用來作為重整傳統系統的指引，得出未來新目標系統的框架，再評估系統框架的適用性及可行性，確認無誤後，進入第二個細節階段。於第二階段從使用案例及類別圖等 UML 圖來表達新目標系統的功能，並由傳統系統中分析出現行的商業規則，再進行差異分析。差異分析的結果可回到第一階段再進行評估及對策略行動方案的檢驗，確認無誤後，進入第三個建構階段。於第三階段依據第二階段產出的 UML 圖示開發原型系統，並撰寫測試案例以進行系統檢驗。系統的檢驗還可比對第二階段產出的商業規則以評估其差異，檢驗程序中可同時進行系統手冊的編寫，系統檢驗確認無誤後，進入第四個移轉階段。於第四階段進行系統的移轉、系統測試及教育訓練等程序，每個循環可回到第一階段再從策略行動方案的檢視，開始另一個更詳細的過程，直到新目標系統正式上線為止。本研究的功能開發階段程序方法如圖 12。

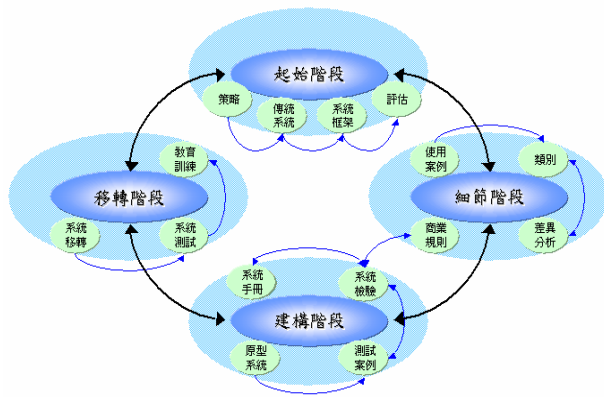


圖12. 功能開發階段的程序

本文將 SDF 的三個階段：策略分析階段、資料移轉階段及功能開發階段，以 UML 類別圖將各階段的執行程序作設計上的說明，並以虛擬碼的方式來描述執行程序。在策略分析階段，首先由系統移轉目標、企業競爭資訊需求、體系間系統連線需求三個方面進行分析，分別得到系統與技術目標、資訊收集與決策分析需求、系統連線技術與資料交換格式，而形成系統移轉策略，再導出關鍵績效指標與策略行動方案，方案由各種相對於策略的工作所組成，各工作將依不同性質分類到相關工作活動，並指派工作的角色，而工作的描述則經由標準工作說明來解釋工作的內容，再得到每項工作應該達到的工作產出。策略分析階段的類別圖如圖 13 所示。再依據類別圖的設計將策略分析階段的執行程序以虛擬碼的方式表達，如表 1。

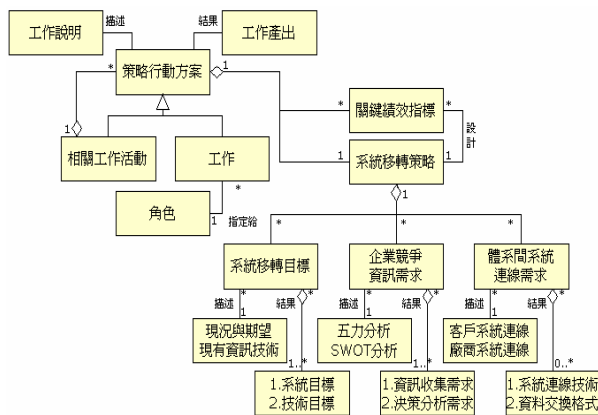


圖13. SDF 策略分析階段設計圖

表1. SDF 策略分析階段虛擬碼

```
//策略行動方案
public class strategyActionProcess {
    public static string main(String args[]) {
        systemTranStrategy stsobj = new
        systemTranStrategy();
        //系統與技術目標
        string targetStrategy =
        stsobj.systemTranTarget();
        //資訊收集與決策分析需求
        string compStrategy =
        stsobj.BusinessCompetition();
```

```
//系統連線技術與資料交換格式
string linkStrategy =
stsobj.DomainLinkage();
//設定關鍵績效指標
string kpi =
setupKeyPerformanceIndex(targetStrategy,
compStrategy, linkStrategy);
//由策略轉換為相對應的工作
string task =
tranStrategyTask(targetStrategy,
compStrategy, linkStrategy);
//將工作分類為相關的工作活動及角色
string groupTaskRole =
summaryGroupTaskRole(task);
return groupTaskRole, kpi;
}
}
//系統移轉策略
public class systemTranStrategy {
    //系統移轉目標
    public string systemTranTarget {
        //分析傳統系統現況
        string asis = analysisLegacy(傳統系統現況);
        //訪談使用者期望
        string tobe = interviewUser(系統使用者);
        string targetResult =
summaryTargetAnalysis(asis, tobe);
return targetResult;
}
//企業競爭資訊需求
public string businessCompetition {
    //五力分析&SWOT 分析
    string bussAny = analysisProcess(企業資訊
現況);
    //彙總分析結果資料收集與決策分析需求
    string compResult =
summaryCompAnalysis(bussAny);
return compResult;
}
//體系間系統連線需求
public string domainLinkage {
    //客戶端系統連線需求之分析
    string linkCus = analysisCustom(客戶及其
資訊架構);
    //廠商端系統連線需求之分析
    string linkVen = analysisVendor(廠商及其
資訊架構);
    //委外端系統連線需求之分析
    string linkSup = analysisSupply(委外及其資
訊架構);
    //彙總系統連線技術&資料交換格式需求
    string linkResult =
summaryLinkAnalysis(linkCus, linkVen,
linkSup);
return linkResult;
}
}
```

在資料移轉階段，首先進行傳統系統資料結構的反向工程，取得傳統系統資料概念描述後，參考策略行動方案中有關於資料結構改良的部份工作，重新設計新目標系統資料概念描述，再根據這

個結果設計新目標系統的資料實體描述，即新目標系統的資料結構。在這個過程中整理出資料移轉的規則，做為日後資料移轉執行時的重要依據。接著進行實體資料的移轉，依據資料移轉規則設計程式碼進行自動的移轉過程，但是傳統與新目標系統的資料結構勢必有所不同，因此需依據資料移轉規則設計必要的資料修補介面，讓系統以自動修補(或是手工)的方式將新目標系統的資料庫內容補齊，再進行資料輸出測試，以確保資料移轉的過程無誤，若測試結果有問題，再回到資料修補介面的程序重新考慮設計方式，資料移轉階段的類別圖如圖 14 所示。再依據類別圖的設計將資料移轉階段的執行程序以虛擬碼的方式表達，如表 2。

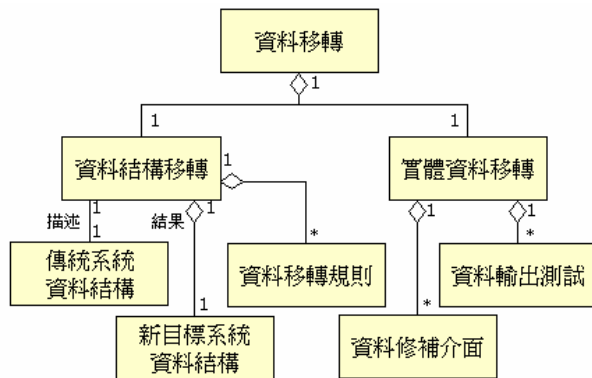


圖 14. SDF 資料移轉階段設計圖

表 2. SDF 資料移轉階段虛擬碼

```
//資料移轉
public class dataTransferProcess {
    public static void main(String args[]) {
        //進行資料結構移轉
        dataFormatTransfer dftobj = new
        dataFormatTransfer();
        dftobj.refineDataFormat();
        //進行實體資料移轉
        dataContentTransfer dctobj = new
        dataContentTransfer();
        dctobj.main();
    }
}
//資料結構移轉
public class dataFormatTransfer {
    public string refineDataFormat {
        //策略行動方案有關資料結構改良的部份
        strategyActionProcess sapobj = new
        strategyActionProcess();
        string actionProcess = sapobj.main();
        string dataActionProcess;
        while (actionProcess.hasMoreElements()) {
            actionProcess =
            actionProcess.nextElement();
            if (actionProcess related to data format)
            {
                dataActionProcess =
                dataActionProcess + actionProcess;
            }
        }
    }
}
//分析傳統系統資料結構設計新資料結構
```

```
string legacyFormat = 傳統系統資料結構;
string legacyLogical =
analysisLegacyStructure(legacyFormat);
string legacyConceptual =
legacyLogicalConceptualize(legacyLogical);
string newConceptual =
refineLegacyConceptual(legacyConceptual,
dataActionProcess);
string newLogical =
designNewLogical(newConceptual);
string targetFormat =
designNewFormat(newLogical);
return targetFormat;
}
//取得資料移轉規則
public string dataTransferRule {
    string legacyFormat = 傳統系統資料結構;
    string targetFormat = 新目標系統資料結構;
    string transferRule =
compareDataFormat(legacyFormat,
targetFormat);
return transferRule;
}
//實體資料移轉
public class dataContentTransfer {
    public static void main(String args[]) {
        //創建新目標系統資料庫結構
        dataFormatTransfer dftobj = new
        dataFormatTransfer();
        string newDataFormat =
dftobj.RefineDataFormat();
        create new database from newDataFormat;
        //依據資料移轉規則將傳統系統資料移轉
        進新目標系統的資料庫
        string newTransferRule =
dftobj.dataTransferRule();
        transferData();
        //依據資料移轉規則設計資料修補介面
        //修補後的資料作輸出的測試直到正常
        while (output test result irregular) {
            design interfaces related to
            newTransferRule to replenish the new
            data structure;
            string legacyOutput =
legacyOutputTestRequire(傳統系統);
            string TargetOutput =
TargetOutputTestRequire(新目標系統
            資料庫);
            if (TargetOutput== legacyOutput) {
                test result regular;
            }
        }
    }
}
}
```

在功能開發階段，首先在起始階段，依據策略行動方案，規劃新目標系統的框架，並進行評估後，得到新目標系統的架構。在細節階段，使用 UML 的各分析圖來描述新目標系統，再由傳統系統取出商業規則，兩者進行差異分析，分析結果可



和起始階段的評估及策略行動方案互為驗證。在建構階段，依據 UML 分析圖進行原型系統的開發，並依據傳統系統的商業規則及策略行動方案進行測試案例的設計，以及系統相關文件的製作。在移轉階段，由原型系統進行系統移轉，由測試案例進行系統測試，由系統文件進行教育訓練。以上四個階段可重覆執行直到系統完整移轉為止。由於資料移轉階段和功能開發階段可以同步進行，當系統進行測試及移轉時，傳統系統實際的資料已經移轉到新目標系統的資料庫中，因此可以立即檢驗系統移轉專案的成功率。功能開發階段的類別圖如圖 15 所示。再依據類別圖的設計將功能開發階段的執行程序以虛擬碼的方式表達，如表 3。

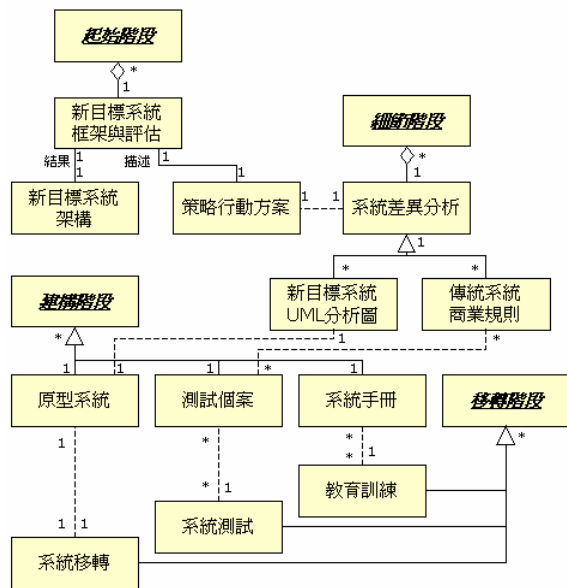


圖 15. SDF 功能開發階段設計圖

表 3. SDF 功能開發階段虛擬碼

```
//功能開發
public class dataTransferProcess {
    public static void main(String args[]) {
        //取得策略行動方案及績效關鍵指標
        strategyActionProcess sapobj = new
        strategyActionProcess();
        string actionProcess, kpi = sapobj.main();
        functionDevelop fdobj = new
        functionDevelop();
        //分析與設計新目標系統
        //並進行差異分析直到無誤
        while (compare result irregular) {
            //取得新目標系統架構及評估結果
            string targetSystem, evaluateResult =
            functionDevelop.inceptionPhase(actionP
            rocess);
            string umlDiagram, businessRule,
            designResult =
            functionDevelop.elaborationPhase(target
            System);
            compare evaluateResult, designResult
            and actionProcess;
        }
        //進行系統設計相關程序
    }
}
```

```
string prototypeSystem, testCase,
systemDocument =
functionDevelop.constructionPhase(umlDiag
ram, businessRule);
//進行系統測試及移轉程序
string transitionSystem =
functionDevelop.transitionPhase(prototypeS
ystem, testCase, systemDocument);
//比較與分析績效關鍵指標
if (compare transitionSystem not match the
kpi) {
    re-execute this process until the result is
    regular;
}
}
}
//功能開發各階段
public class functionDevelop {
    //起始階段
    public string inceptionPhase (string
    actionProcess) {
        //規劃新目標系統的框架與評估
        while (target system framework irregular) {
            string legacySystem = 傳統系統;
            string targetSystem =
            designFramework(legacySystem,
            actionProcess);
            string evaluateResult = evaluate target
            system framework regular or not;
        }
        return targetSystem, evaluateResult;
    }
    //細節階段
    public string elaborationPhase (string
    targetSystem) {
        //規劃新目標系統的 UML 分析圖
        string umlDiagram =
        analysisTargetSystem(targetSystem);
        //分析傳統系統的商業規則
        string businessRule =
        analysisLegacySystem(傳統系統);
        //進行差異分析直到兩者符合
        while (uml diagram not match legacy
        business rule) {
            umlDiagram =
            refineTargetSystem(umlDiagram);
            designResult =
            compareDesignSystem(umlDiagram,
            businessRule);
        }
        return umlDiagram, businessRule,
        designResult;
    }
    //建構階段
    public string constructionPhase (string
    umlDiagram, businessRule) {
        //設計原型系統
        string prototypeSystem =
        designPrototypeSystem(umlDiagram);
        //設計測試案例
        string testCase =
        designTestCase(businessRule);
        //撰寫系統相關文件
    }
}
```

```

string systemDocument =
produceSystemDocument(prototypeSystem,
umlDiagram, testCase);
return prototypeSystem, testCase,
systemDocument;
}
//移轉階段
public string transitionPhase (string
prototypeSystem, testCase, systemDocument) {
//由原型系統進行系統移轉
string transitionSystem =
transitionProcess(prototypeSystem);
//由測試案例進行系統測試
systemTest(transitionSystem, testCase);
//由系統文件進行教育訓練
systemTraining(transitionSystem,
systemDocument);
return transitionSystem;
}
}

```

#### 四、 績效評估與比較分析

Bisbal 等人對於一般系統移轉的程序提出了 5 個階段的分析分別為：整體判斷階段、了解傳統系統階段、開發新目標系統階段、測試階段、系統移轉階段。Bisbal 等人以這 5 個階段及第二章中所提及的各系統移轉方法提出了分析表[11]。在這份分析表中，明顯的以蝴蝶法佔有較優勢的評比，而蝴蝶法所提到的資料移轉相關技術也和 SDF 方法有類似的部分。因此，本文將以 SDF 方法和蝴蝶法作一 KPI 評估比較並加以分析，以證實 SDF 方法的優勢。

KPI 的設定採用 Bisbal 等人評估各移轉方法所提出研究的議題，提供可比較的項目分別為：移轉方法論內容、傳統系統的了解程序、新目標系統的開發、資料移轉程序、移轉的工具等五大議題。因此各項指標將依這五大議題分類，分別比較 SDF 方法及蝴蝶法的優缺點。SDF 方法和蝴蝶法的評估比較結果如表 4。在評估比較表中，可以明顯的看出 SDF 方法在各 KPI 指標項目都優於蝴蝶法。由於蝴蝶法著重於資料移轉的程序及移轉時間控制的演算法，其他的系統移轉議題並沒有太多的研究。尤其在方法論主軸的 KPI 項目中，更指出了蝴蝶法的實作結果只可能是傳統系統的更新版本。因新版本的開發過程中，各項功能的需求可能會因企業或使用者的需求而逐漸的進化，如果這個變數沒有考慮進來，就算系統順利的移轉後，新目標系統還是無法滿足改版後的需求。而 SDF 方法則將這個重要的議題列為最優先的考量，因此新目標系統能同時滿足使用者及企業未來營運的需求。

表4. SDF 方法和蝴蝶法的評估比較表

關鍵績效指標	SDF 方法評估說明	蝴蝶法評估說明
1.1 方法論的主軸	以資料及企業營運策略為導向，重製新	以資料為導向，著重在資料移轉的過

	目標系統，資料移轉及開發程序可以同時進行。新目標系統能符合企業營運的未來需求。	程，保持不間斷的同步移轉程序，直到所有資料移轉完成。新目標系統為傳統系統的改版。
1.2 方法論的 I&I 特性	資料移轉過程及開發程序符合 I&I 特性，並設定 KPI 以檢驗執行的成果。	資料移轉過程符合 I&I 的特性，沒有 KPI 以驗證執行成果。
1.3 方法論的可行性	DMP 及 DOT 可透過現有的軟體工具組合成資料移轉平台。	DAA&Data-Transformer 可透過軟體元件的方式開發實作。
1.4 方法論的相關技術性	整合策略分析、資料、系統開發、網際網路及系統連線等相關技術，以重製新目標系統。	開發資料移轉平台所用到的相關技術，至於新目標系統開發的相關技術並無提及。
2.1 應用程式的了解程度	依據策略行動方案選擇性的了解傳統系統程式，轉換其商業規則為開發新目標系統的依據。	傳統系統的應用程式必須要完全的了解，否則無法找到資料模型間的映對。
2.2 資料的了解程度	傳統系統的資料依策略行動方案選擇性了解，並依需求及目標重製結構。	傳統系統的資料需完全了解，以製作資料移轉平台。
3.1 功能的滿足性	傳統系統及新目標系統的功能不一定相同，依企業營運需求重新調整各相關功能。	傳統系統及新目標系統的功能大致上相同，為其更新版本。
3.2 元件的可重覆使用性	新目標系統的開發程序以物件導向設計方式，元件可重覆使用。	沒有相關的議題。
4.1 資料的滿足性	新目標系統的資料除了能滿足傳統系統的需要外，還能滿足企業未來營運的需求。	新目標系統的資料滿足傳統系統的需要。
4.2 有效且及時的移轉引擎	以現有的軟體工具組合成資料移轉平台，能及時的移轉資料。	以開發軟體元件方式建置資料移轉平台，能及時的移轉資料。
4.3 移轉有效資料的判斷力	由 DMP 中資料移轉規則元件及資料補修介面以移轉有效資料。	傳統系統資料完整的移轉，沒有區分資料的有效性。
5.1 工具導向的平台及開發特性	以現有的軟體工具組合成資料移轉平台，可隨不同資料移轉需求選擇不同軟體工具來整合。	以開發軟體元件方式建置資料移轉平台，開發完成後只能使用於單一的移轉專案中。

#### 五、 結論

既有的系統移轉方法雖有明確的執行程序，

但最終的結果也只是將傳統系統的功能原封不動的移轉到新目標系統，至於新目標系統是否能符合企業日後的發展及競爭的要求，則不在移轉程序中考量，因此新目標系統的實用性對企業而言就和傳統系統是一樣的，系統移轉的結果看不出明顯的效益。而 SDF 的方法則參照既有的系統移轉方法加以改良，加入企業資訊策略分析及策略行動方案的目標引導，使得系統移轉的程序能符合企業日後的需求，同時也考量了傳統系統中資料的重要性，依策略行動方案重新設計新目標系統的資料結構，並規劃二種資料結構中間的轉換程序。當資料轉換程序確認後，傳統系統的資料即可非常便利的轉換到新目標系統，而最後運用遞增及反覆式的開發方式，依新目標系統的資料結構重新設計使用者介面，當設計完成且經由測試無誤後，即可進行移轉系統的步驟。此完整的 SDF 方法較既有的系統移轉方法其轉換的程序更為順暢，而且新目標系統功能符合企業的需求，同時也保存了企業最重要的營運資料。

本文中 SDF 方法的各階段以 UML 類別圖的設計方式說明之，再以虛擬碼來描述執行程序。並和傳統較常使用的方法依功能作關鍵績效指標評估比較並加以分析，以證實 SDF 方法的效果與效率。由於目前系統移轉相關的研究並未提出更新的作法，因此 SDF 方法對企業所迫切需要的系統移轉專案提供一個更好的選擇。未來，SDF 方法於企業資訊策略分析中的策略行動方案中可再深入的探討，以提出更有效率的系統移轉程序。另外，可考慮進行傳統系統及新目標系統之間的 DMP 資料移轉器之細部設計方式，以提高資料移轉的完整性及一致性。

## 六、參考文獻

- [1] 李妙福，「國內企業 e-化發展關鍵成功因素之研究」，國立台北大學，碩士論文，1999。
- [2] 周樹林，「我國資訊軟體市場發展現況與趨勢」，資策會 MIC，2004。
- [3] 蔡春智，「使用者導向之企業資源規劃系統移轉方法」，東海大學，碩士論文，2000。
- [4] 經濟部技術處示範性資訊應用開發計畫，<http://doit.moea.gov.tw>，2005。
- [5] A. Bateman, J. Murphy, "Migration of Legacy Systems", School of Computer Applications, Dublin City University, Working Paper A-2894, <http://www.compapp.dcu.ie>, 1994.
- [6] Aebi, D., "A Process Model for Re-Engineering, Migration and Multi-Use of business Data", In *Proceedings of 1st Euromicro Working Conference on Software Maintenance and Reengineering (CSMR '97)*, Berlin, GERMANY, March, 1997.
- [7] Amjad Umar, "Application Reengineering: Building Web-Based Applications and Dealing with Legacies", Prentice Hall, New Jersey, 1997.
- [8] B. Wu, et al., "Legacy Systems Migration - A Method and its Tool-kit Framework", IEEE Computer Society, pp. 312-320, 1997.
- [9] B. Wu, et al., "The Butterfly Methodology: A Gateway-free Approach for Migrating Legacy Information Systems", In *Proceedings of the 3rd IEEE Conference on Engineering of Complex Computer Systems (ICECCS97)*, pp. 200-205, Villa Olmo, Italy, September 1997.
- [10] Grady Booch, James Rumbaugh, Ivar Jacobson, "The Unified Modeling Language User Guide", Addison Wesley, New York, 1998.
- [11] J. Bisbal, et al., "A Survey of Research into Legacy System Migration", Technical Report TCD-CS-1997-01, Computer Science Department, Trinity College Dublin. January 1997.
- [12] Jean HENRARD, 「Program Understanding in Database Reverse Engineering」, University of Namur, Belgium, Thesis for the degree of Doctor, 2003.
- [13] Maria Istela Cagnin, et al., "PARFAIT: Towards a Framework-based Agile Reengineering Process", In *Proceedings of Agile Development Conference*, pp. 22-31, Utah, America, 2003.
- [14] M. Brodie, M. Stonebraker, "DARWIN: On the Incremental Migration of Legacy Information Systems", TR-022-10-92-165 GTE Labs Inc., <http://info.gte.com/ftp/doc/tech-reports/tech-reports.html>, Mar. 1993.
- [15] M. Brodie and M. Stonebraker, "Migrating Legacy Systems: Gateways, Interfaces and the Incremental Approach", Morgan Kaufmann, USA, 1995.
- [16] M. Mecella and B. Pernici, "Designing Wrapper Components for e-Services in Integrating Heterogeneous Systems", pp. 2-15, *VLDB Journal*, 10, 1, 2001.
- [17] Porter, Michael E., "Competitive Strategy", Free Press, New York, 1998.
- [18] Robert C. Seacord, et al., "Legacy System Modernization Strategies", Technical Report CMU/SEI-2001-TR-025, Software Engineering Institute, Carnegie Mellon University, July 2001.
- [19] Teta Stamati, et al., "Legacy Migration as Planned Organizational Change", In *Proceedings of the 6th International Conference on Enterprise Information Systems*, pp. 501-508, Porto, Portugal, April 2004.
- [20] Ying Zou and Kostas A. Kontogiannis, "Web-based Legacy System Migration and Integration", In *Proceedings of the 4th International Conference in Systematic, Cybernetics and Informatics (SCI)*, Orlando, USA, July 2000