

一個交易系統稽核軌跡建構之 OTO 資料集模式

顧添利

實踐大學會計學系

(收稿日期：87 年 4 月 29 日；第一次修正：87 年 6 月 22 日；
接受刊登日期：87 年 8 月 17 日)

摘要

交易系統清楚且完整的稽核軌跡或審計軌跡 (Audit Trails) 對於稽核、管理與控制皆非常重要。在人工作業的環境裡，交易活動所產生的資料都記載於文件與帳簿上，稽核軌跡是由一組文件憑證、帳簿與報表上的參照編號 (Reference Codes) 所構成，整個交易處理軌跡都可用肉眼在文件與帳簿上查看。但在電腦化作業的環境下，稽核軌跡不但無法用肉眼直接在電子形式的資料庫裡查對，而且也因為資料庫設計原理的特性，而使得其更為模糊不清。本研究發現反函數的理論，可以用來建立兩組資料集元素間一對一的映射關係，而得到兩組資料集間雙向的資料轉換軌跡。利用反函數一對一的映射特性，本研究提出一個 OTO 資料集模式，來連結關連式資料庫裡相關 Table 間的關係，以做為交易系統稽核軌跡建構之依據。

關鍵詞彙：稽核軌跡 (審計軌跡)，一對一資料集模式，交易系統，資訊系統，關連式資料庫

壹 緒論

稽核軌跡又稱審計軌跡 (Audit Trails)，其是由記載著一連串交易處理過程的一組文件與帳簿或電子資料，依據其文件與帳簿或資料記錄的參照編號 (Reference Codes)，而與處理程序的步驟連結在一起的資料軌跡 (Arens and Loebbecke, 1995；Gelinas, et. al., 1993)。透過這些軌跡，可以追查從交易的開始，至整個處理過程，而到最後的資訊結果；相反地，也可從最後的資訊結果逆查到整個處理過程，而至交易開始的源頭。以總帳會計交易系統為例，稽核軌跡是指由交易發生之原始憑證與記帳傳票至記錄分錄之日記帳簿，再過入總分類帳或明細分類帳，最後產生財務報表之一連串有跡可循之過程；相反地，也可指由最後的財務報表上的資訊，逆查到明細分類帳或是總分類帳，再到記錄分錄之日記帳簿，最後到交易發生之記帳傳票與原始憑證之反向處理軌跡 (Wilkinson and Cerullo, 1997)。

任何系統的處理過程皆必須要有跡可循，才能夠迅速的找出問題並加以控制 (吳琮璠, 1997；Gallegos and Bieber, 1987)。交易系統清楚且完整的稽核軌跡對營運控制、會計控制、查核工作及遵循政府法令規定等均極為重要

(Wilkinson and Cerullo, 1997)。在交易系統發生問題時，管理人員可以藉稽核軌跡發現問題所在，並用來斷定產生問題的原因與後果。而企業組織的內外部稽核人員也可用稽核軌跡來做為證實測試之基礎，以查出系統的錯誤與不法 (Crump, 1981 ; Hinde, 1979 ; Mcfadden, 1997)。稽核軌跡模糊不清的交易系統，其處理過程的錯誤與舞弊將因為無跡可循而難以查出，而其稽核工作也將因為無法獲得清楚的處理軌跡而滯礙難行。所以一個安全可靠的交易系統必須要有著清楚且完整的稽核軌跡 (Cushing and Romney, 1993 ; Hinde, 1984 ; Skudrna, 1982)。

在人工作業的環境裡，交易處理的活動都記載於文件與帳簿上，稽核軌跡是由一組記載於文件憑證、帳簿與報表上的參照編號所構成，整個交易處理過程都可用肉眼直接在文件與帳簿上查看。但資料處理的環境從人工作業轉變到電腦作業後，交易系統的稽核軌跡不但無法用肉眼直接在電子資料庫裡查對，而且也因為資料庫設計原理的特性，而使其變得更加模糊不清。由於資訊科技的進步，企業組織的各個部門或業務活動皆已電腦化，交易系統的業務活動已經與其資訊系統緊密地相結合。目前通訊網路、電子資料交換的應用與跨組織系統已經相當地盛行，企業組織不同部門間或企業組織之間交易文件與資料的傳遞與處理，已大部份經由通訊網路來進行，因此無紙化的企業環境即將來臨。在這無紙化的環境下，由於網路資訊系統的高度複雜化，因而使得交易系統的處理軌跡更難以追查 (Watne and Turney, 1990)。

在資訊技術 (Information Technology, 簡稱 IT) 的環境下，交易活動所產生的資料都記載於其資訊系統的資料庫裡，因此要建構交易系統的稽核軌跡，就必須從如何建立資料庫之資料軌跡來著手。本研究發現反函數 (Inverse Function) 的理論，可以用來建立兩組資料集的元素間一對一的映射關係 (One To One Mapping Relation)，而得到兩組資料間雙向的資料轉換 (Transformation) 軌跡。利用反函數一對一的映射特性，本研究提出一個 OTO 資料集模式，來連結關連式資料庫 (Relational Data Base) 裡相關表格 (Table) 間記錄資料 (Record) 之關係，以做為交易系統稽核軌跡建構之依據。在後面，本論文將會以一個簡單的總帳會計交易系統為例子，以探討如何利用 OTO 資料集模式，來建構系統清楚且完整的雙向稽核軌跡。

貳 文獻探討

有關提到稽核軌跡重要性之文獻非常多 (葉誌崇, 1997 ; Bieber, 1987 ; Brown, 1983 ; Cushing & Romney, 1993 ; Davis, 1996 ; Gallegos and Basica,

1986), 但是有關建構稽核軌跡方法之文獻卻不多見。Weber (1982) 認為要設計交易系統完善的稽核軌跡, 必須要先瞭解稽核軌跡的本質, 因此其提出要建構先進電腦會計系統的稽核軌跡, 必須要先確認稽核軌跡之目的, 然後再來建構所需要之稽核軌跡。其認為電腦會計系統稽核軌跡之主要目的有: 回答詢問 (answering inquiries)、更正錯誤 (correction of errors)、確認錯誤的原因 (determining the consequences of errors)、弊端的防止 (fraud deterrence)、安全的監督 (security monitoring)、備份與回復 (backup and recovery) 及績效監督 (performance monitoring) 等七個項目。

Cerullo (1983) 認為在一個即時連線 (online real time) 的電腦系統裡, 為了保存稽核軌跡, 必須採取一些特殊的安全與控制措施, 其包括說明 (instructions)、回讀 (read back)、完整檢查 (completeness check)、冗餘檢查 (redundancy check)、交易序號 (serial numbering of transactions)、時間控制總數 (time control totals)、資料傾倒 (dumps)、硬本文件單元 (hard copy units)、交易日誌 (transaction log)、重複資料檔案 (duplicate data files) 及持續的監督與稽核 (continuous monitoring and auditing) 等措施。

Menkus (1990) 提出稽核軌跡是銷售點終端資料處理系統最有效的控制工具, 可利用稽核軌跡來幫忙維護零售業資訊的一致性, 並用來考核與控制銷售員之績效。稽核軌跡必須設計在系統發展之時, 如此才能產生最大的效用。保存銷售點終端處理系統稽核軌跡的記錄應包括一個終端機的辨識碼 (a terminal identifier)、一個銷售人員的辨識碼 (a salesperson identifier)、一個交易記錄號碼 (a transaction record number)、一個交易日期及當天的序號 (a transaction date and interday sequence)、一個交易類別 (a transaction type)、交易明細 (transaction details) 及一個檔案變動的資料 (a file change image) 等七項資料。

Cerullo 與 Menkus 所提出建構系統稽核軌跡之方法有一個共通點, 就是在交易或資料的處理過程中, 將一些關鍵點所產生的資料做記載保存與控制檢查的處理, 以做為事後追查交易活動之依據。因此, 本研究亦認為要建立交易系統之稽核軌跡, 必須從如何記載與保存交易活動所產生的資料來著手。

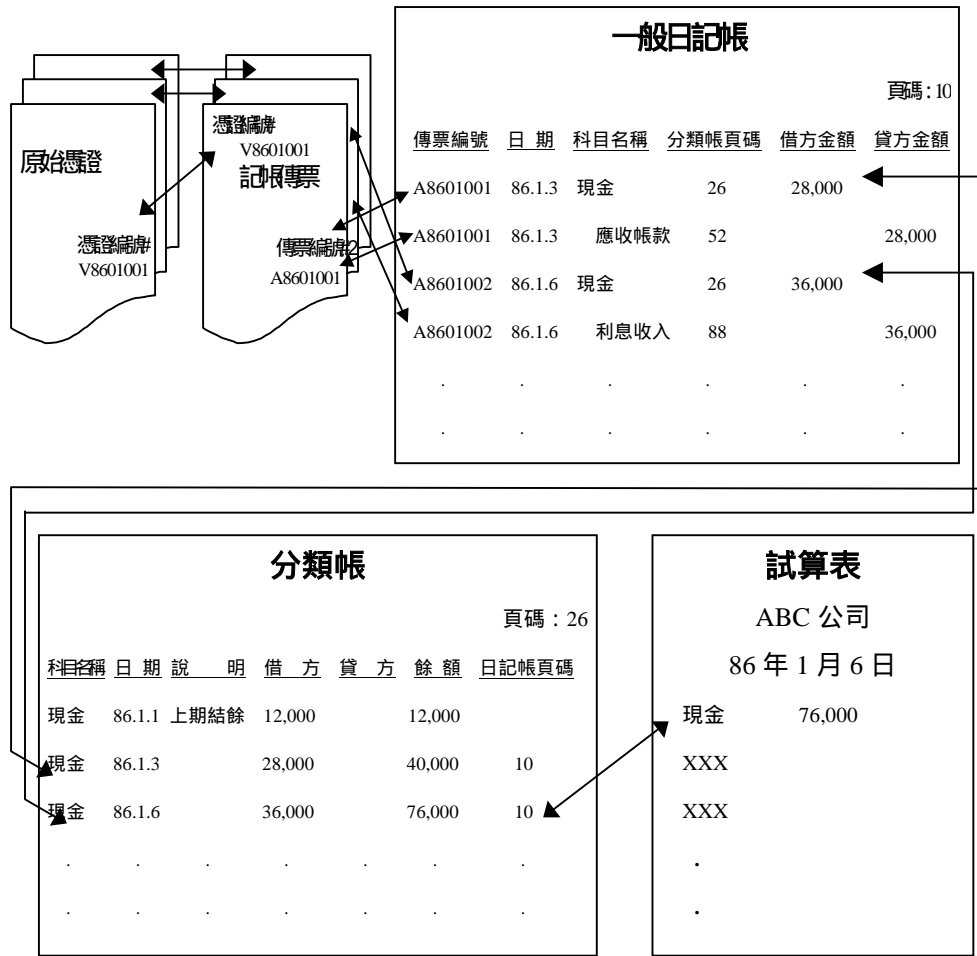
參 稽核軌跡之消失與再造之途徑

一、稽核軌跡之消失

在人工處理系統之下，稽核軌跡是記載於肉眼可見的 (Visible) 憑證與文件上，因此可以十分清楚地來查看；而在資訊化後，原始文件或硬本文件 (Hard Copy) 逐漸減少，甚至於消失，而使得可見的稽核軌跡減少 (吳琮璠, 1997；Waples, 1989)。雖然硬本文件的減少而使得肉眼可見的稽核軌跡漸次地消失，而且網路系統的普遍使用也使得稽核軌跡的找尋不易，但其並不意謂著稽核軌跡消失或是變得模糊，只是稽核軌跡不再記錄於肉眼可見的文件帳簿上，而是以另外一種肉眼無法直接查對的電子形式保存在可能分散於各地的電子媒體裡。

交易系統稽核軌跡真正變得模糊或消失，其實是交易活動所產生的資料之組織方法的改變。在人工作業的資料處理環境裡，記載在原始文件、帳簿及報表等紙張媒體間的交易活動資料，由於是人工肉眼可見的處理資料，其資料並無特殊的結構限制，祇要能透過參照編號來產生交易資料的關連與軌跡，則不須特殊考量到記載於文件與帳簿上資料之組織。以圖一的簡單會計總帳交易系統為例，其在文件帳簿媒體間所儲存的交易資料，都沒有任何特殊的結構限制，但卻可經由設計好的參照編號來產生雙向且完整的資料軌跡。在圖一裡，可以從交易發生的原始憑證之編號追查到其記帳傳票，從傳票的編號追查到日記帳，從日記帳的分類帳頁碼追查到分類帳，再從分類帳的科目名稱追查到最後試算表上的資訊，而由相關的參照編號組成一連串的資料軌跡；相反地，也可透過參照編號，從試算表的科目名稱逆查到分類帳，從分類帳的日記帳頁碼逆查到日記帳，從日記帳的傳票編號逆查到記帳傳票，再從記帳傳票的憑證編號而逆查到交易發生的原始憑證等一連串的反向資料軌跡。

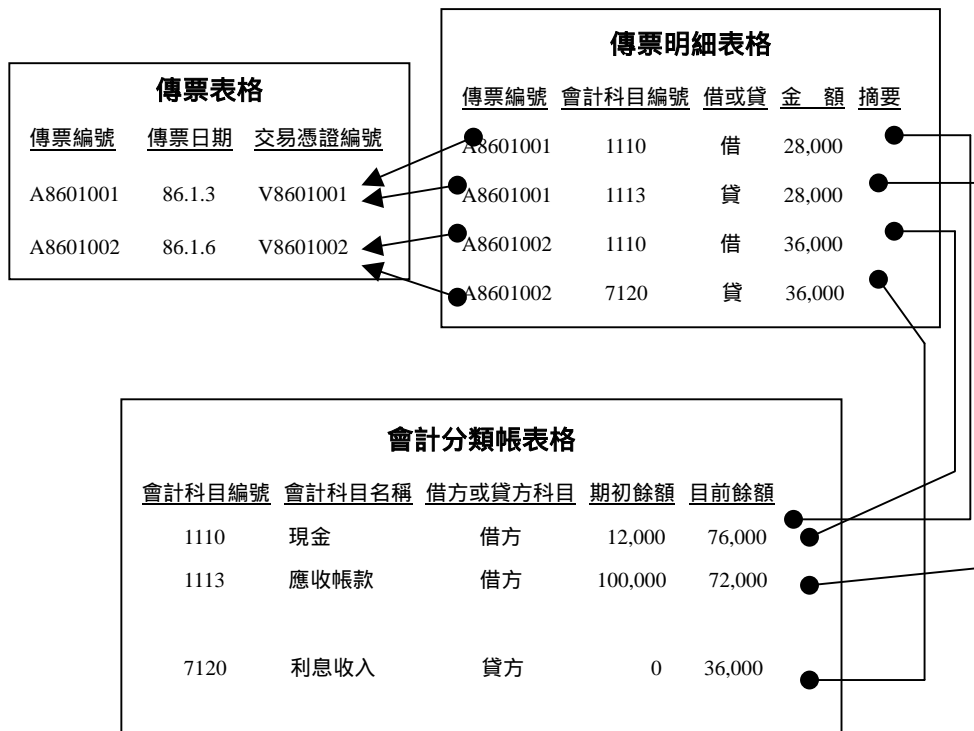
但資料處理電腦化後，交易活動所產生的資料都保存在經過嚴謹設計的資料庫裡，資料庫設計時必須考量到資料的邏輯結構，而疏忽了資料軌跡的保存。以目前使用最普遍的關連式資料庫為例，其資料庫設計時必須考量到資料正規化 (Normalization) 的問題，正規化的過程最主要的目的是要去除資料的重覆，及增加、刪除與修改資料時所可能產生的異常，其關注點乃是在表格檔案的組織及資料的存取技術，以致於使得其資料結構無法有效地保存交易活動的處理軌跡。



圖一 會計總帳交易系統文件帳簿之資料結構與軌跡圖

依照關連式的模式來設計簡單總帳會計交易系統的資料庫，經過正規化的過程後，一般可以得到如圖二的「傳票」、「傳票明細」與「會計分類帳」等三個表格 (Table)。如果一筆傳票的記錄 (Record) 有兩個會計科目，則在「傳票明細」表格裡就會有兩筆記錄資料，而每個會計分類帳科目在「會計分類帳」表格裡祇會有一筆記錄。「傳票明細」表格的記錄資料則依照會計科目編號過帳到「會計分類帳」表格的相關記錄裡。在圖二裡，從「傳票明細」表格的記錄可以找到「傳票」表格裡相關的傳票記錄，而從「傳票」表格的傳票記錄則無法確認可以找到「傳票明細」表格裡所有相關的明細記錄資料；而「傳票明細」與「會計分類帳」表格之間記錄資料的雙向軌跡也是模糊不清的。「傳票明細」表格的記錄有無正確過帳到「會計分類帳」表格的相關會計科目記錄裡並無法得到確認；相反地，「會計分類帳」表格的會計科目餘額是如何從「傳

票明細」表格的記錄過帳而來，其軌跡也是模糊不清的。因此在電腦化處理總帳會計交易資料後，其清楚的資料軌跡往往不易在關連式的資料庫裡找尋到。

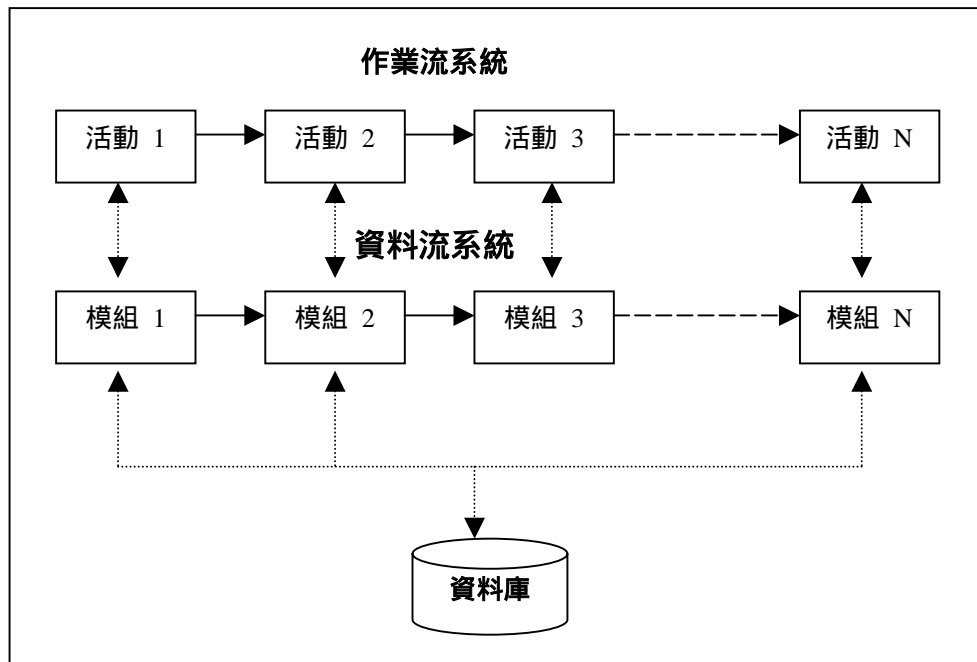


圖二 總帳會計交易系統之關連式資料庫與模糊之資料軌跡圖

二、稽核軌跡再造之途徑

一個企業組織可能有許多之交易系統或企業程序 (Business Process)，一般來說，每個交易系統都可分為作業流與資料流的活動，資料流記載著作業流的詳細活動，祇要作業流活動的資料軌跡保存在資料流裡，就可利用資料流的資訊來稽核、管理與控制作業流的活動。在圖三的關係圖裡，一個交易系統的作業流可以有 N 個作業活動，每一個作業活動在資料流裡都會有一個相對應的處理模組來記載與處理其所產生的資料。所以在作業流活動的進行中，有一個資料流系統正在記載其活動的細節，因而透過資料流的資訊，即可追查到作業活動的處理過程。而每一個處理模組在記載與處理作業活動所產生的資料時，必須在資料庫裡對相關的表格資料做新增、刪除或修改其記錄內容的工作。

一個關連式的資料庫是由許多相關的表格 (Table) 所組成，一個表格一般皆記載著交易系統裡一個資料實體 (Entity) 或實體間關連 (Relation) 的活動資料，並且利用表格間欄位的關連來表示整個交易系統的運轉規則與過程。所以一個關連式的資料庫其實就是儲存著交易系統一連串相關交易活動之影像，其記載與保存了整個系統交易活動之過程。



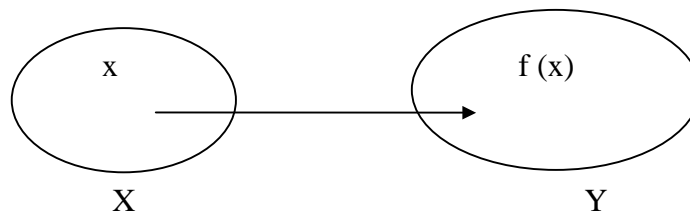
圖三 交易系統之活動與資料關係圖

在人工作業的環境下，交易活動是記錄與保存在文件與帳簿裡，其資料的組成是不受任何特殊結構的限制；而在 IT 的環境下，交易活動的資料是保存在可能分散於各處的電子資料庫裡，由於資料庫的設計必須依據其特定邏輯結構的原理，因而忽略了資料軌跡的保存。因此祇要在資料庫的設計時，在不違反其設計的原理下，加上能產生資料軌跡的資料，即可使得企業組織的交易系統，再度找回其已消失的稽核軌跡。所以要建構一個交易系統的稽核軌跡，其途徑就是從其資料庫裡的資料組織來著手進行。

肆 稽核軌跡建構之資料模式

一、反函數 (Inverse Function) 之特性

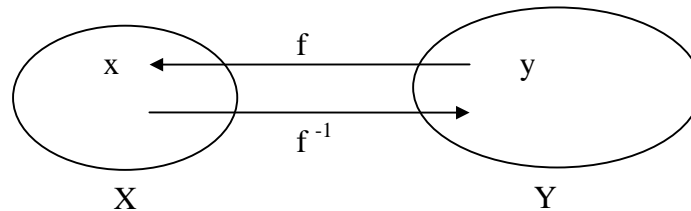
假設 X 與 Y 是兩個資料集合 (Set)，如果任何一個元素 (Element) x 在 X 集合裡，都可以在 Y 集合裡找到一個唯一 (Unique) 的元素值 y ，那麼就可以說有一個函數 (Function) f 從 X 集合到 Y 集合。如圖四所示，可用 $f : X \rightarrow Y$ 或 $f(x)$ 來表示 (Kolman and Busby, 1987 ; Truss, 1991)。



圖四 函數 $f : X \rightarrow Y$

簡單地說，函數就是一個定義好的運算式或處理過程，用其任何一個定義範圍內的元素 (Element) 當成函數的參數 (Parameter)，再利用其已定義的運算式加以處理，則可以得到一個唯一的值 (Value)。

如圖五所示，若函數 f 是從 X 到 Y ，而有另一個函數 f^{-1} 是從 Y 到 X ，假設任何一個元素 y 在 Y 集合裡，都可以在 X 集合裡找到一個唯一的元素值 x ，則稱 f^{-1} 為 f 之反函數，且 $f f^{-1}(y) = y$ 與 $f^{-1} f(x) = x$ 的公式皆成立 (Kolman and Busby, 1987 ; Silverman, 1985 ; Truss, 1991)。



圖五 反函數 f^{-1}

反函數 (Inverse Function) 其實也是一個已定義的運算式，利用其函數處理後所得到的值當成其參數值，然後加以運算，而可得到一個唯一的值，此值即是本來函數的參數值。

兩組資料集合之間如果有反函數的關係，那麼這兩組資料之間的關連也就是一對一的映射 (Mapping) 關係，即可從 X 組資料集的每個元素映射到 Y 組資料集的每個元素；相反地，也可從 Y 組資料集的每個元素反向映射到 X

組資料集的每一個元素，則此兩組資料間元素的轉換與處理過程就有跡可循。因而利用反函數的特性即可用來產生兩組資料集間正向與反向的處理軌跡。

二、一對一資料集模式 (One To One Data Set Model)

在 IT 的環境下，交易系統所產生的資料幾乎是全部存放在可能分散於各處的電子資料庫裡，資料及其處理過程都是肉眼不可見的，因此為了追查交易系統的錯誤與舞弊，其交易系統的資料庫裡必須要有一個完整且清楚的資料處理軌跡，以用來追查正向與反向資料之轉換過程。反函數的特性既然能用來建立兩組資料集間雙向完整之資料轉換軌跡，那麼也可用來建構兩個相關連的表格間一對一的映射關係。因此本研究利用反函數一對一映射之特性，提出一個一對一資料集模式 (One To One Data Set Model, 簡稱 OTO Data Set Model)，以利用此資料集模式來重新定義關連式資料庫裡任何兩個相關表格間之關係，再分別利用其各自欄位所產生的兩個資料集，來建立元素間一對一的映射關係，最後再利用資料集元素之鍵值欄位，來間接建立兩個相關 Table 間記錄的關連，而使得整個交易系統資料的處理過程有著雙向清楚的轉換軌跡。因此，可利用 OTO 資料集模式來強化關連式資料庫設計時對稽核軌跡之疏失。

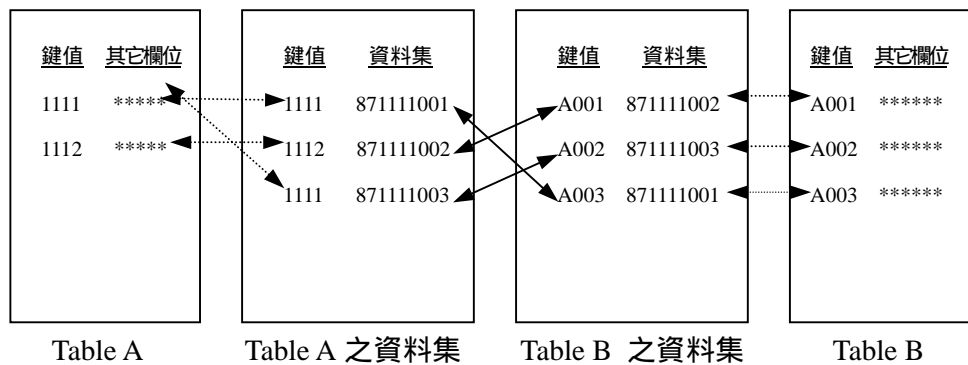
使用 OTO 模式建立資料集之步驟如下：

- (1)重新定義關連式資料庫裡任何兩個相關表格間之關係。在此步驟裡必須調整或新增兩個相關表格間的關連欄位，以利用此關連欄位來求得兩個 Table 間能一對一映射的兩組資料集元素。重新定義兩個相關 Table 間的關係時，其欄位的調整與新增並沒有一定的方式，只要能在兩個相關的 Table 間產生一對 OTO 資料集即可。
- (2)利用兩個相關表格間的關連欄位，產生各自的一組資料集元素。建立資料集元素時，可依據兩個表格間之關係，利用各自的欄位或欄位的運算式來產生。經過正規化後的資料庫，其兩個相關的表格大多為一對多的關係，“一”這邊的表格一般都必須用欄位運算式來產生資料集的元素，而“多”這邊的表格，則可直接用欄位或欄位的連結來產生資料集元素，而且兩組相關資料集元素的數目必須相同。產生資料集元素時，必須同時也保存其記錄之鍵值欄位。
- (3)建立兩組相關資料集元素間一對一的映射關係。在建立兩組資料集元素間一對一的映射關係時，如果兩組資料集裡的元素皆互有相同的值，則可以直接利用相互間的元素值來建立一對一的關連；否則，必須透過函

數運算式的運算後，兩個相關連的元素間才能建立一對一的映射關係，譬如 x 與 y 兩個元素之間的映射是由 $f(x) = y$ 與 $f^{-1}(y) = x$ 之運算所達成。

(4) 間接由資料集元素之鍵值欄位來建立相關表格間記錄資料之關連。而利用相關表格間記錄資料之關連，即可建構整個交易系統清楚且完整的稽核軌跡。

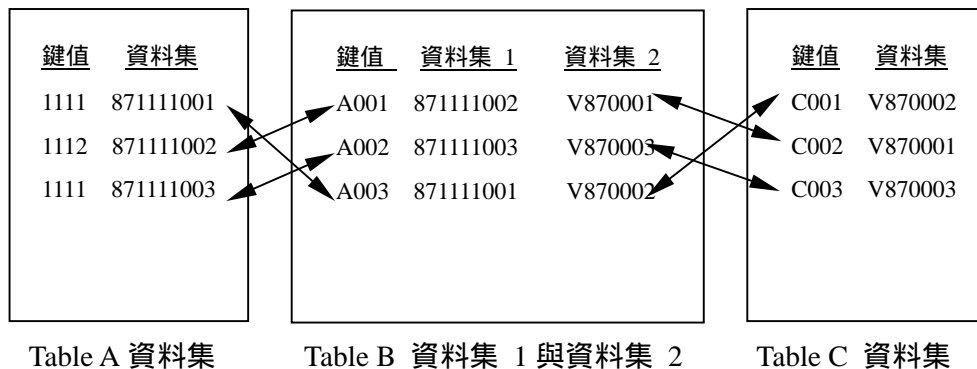
假設如圖六有 A 與 B 兩個相關的表格，經過 OTO 資料集模式重新定義其關係後，即可利用各自表格的欄位來產生一對一相互映射的兩組資料集合。譬如利用兩個資料集皆有的“871111001”資料元素，來直接產生一對一的映射關係，則可間接建立 Table A 鍵值為“1111”之記錄與 Table B 鍵值為“A003”之記錄間的關連。



圖六 兩個 Table 之 OTO 資料集

假設某一個交易系統的資料庫裡有三個表格，Table A 與 Table B 有關連，而 Table B 又與 Table C 有關連，那麼經由 OTO 資料集模式的重新定義後，就可產生如圖七之資料集，Table A 的資料集與 Table B 的資料集 1 是相互一對一映射的關係，而 Table B 的資料集 2 與 Table C 的資料集也是相互一對一映射的關係。經由 Table A 資料集與 Table B 資料集 1 的“871111002”資料元素之一對一映射關係，可用來建立 Table A 鍵值為“1112”之記錄與 Table B 鍵值為“A001”之記錄間的關連。而經由 Table B 資料集 2 與 Table C 資料集的“V870001”資料元素之一對一映射關係，則可用來建立 Table B 鍵值為“A001”之記錄與 Table C 鍵值為“C002”之記錄間的關連。如此，Table A 鍵值為“1112”的記錄可經由“871111002”的資料集元素，追查到 Table B 之鍵值為“A001”之記錄，再經由 Table B 資料集 2 的“V870001”之元素追查到 Table C

鍵值為"C002"之記錄。相反地，也可從 Table C 鍵值為"C002"之記錄逆查到 Table A 鍵值為"1112"之記錄。



圖七 多 Table 之 OTO 資料集

伍 稽核軌跡建構之程序

利用 OTO 資料集模式來建構一個交易系統之稽核軌跡，其共有如下之四個步驟：

- (1)設計交易系統之資料庫。
- (2)利用 OTO 資料集模式重新定義相關 Table 之關係。
- (3)產生資料集及建立 OTO 之關係。
- (4)驗證交易系統是否有清楚完整之稽核軌跡。

首先必須利用關連式資料庫的設計原理，來設計一個交易系統之資料庫，然後再利用 OTO 資料集模式來重新定義任何兩個相關 Table 間之關係，最後再產生 OTO 資料集，並驗證交易系統是否保存著清楚完整之稽核軌跡。如果發現稽核軌跡不夠清楚完整，那麼就必須回到步驟 2，再重新定義相關 Table 之關係，以產生新的 OTO 資料集，然後再一次驗證交易系統清楚完整的稽核軌跡是否存在。

下面以簡單的總帳會計交易系統為例子，來探討如何利用 OTO 資料集模式來建構其清楚且完整的稽核軌跡。

一、設計交易系統之資料庫

用關連式的模式來設計總帳會計系統的資料庫，在正規化的過程裡，至少要使表格達到第三階的正規格式 (Third Normal Form)。假設依據交易系統的資訊需求，設計的結果產生了傳票、傳票明細及會計分類帳等三個關連

(Relation) 或表格 (Table)。傳票編號是「傳票」Table 的鍵值欄位，傳票編號與會計科目編號是「傳票明細」Table 的鍵值欄位，而會計科目編號是「會計分類帳」Table 的鍵值欄位。其資料庫的結構如下：

傳票 (傳票編號，傳票日期，交易憑證編號)

傳票明細 (傳票編號，會計科目編號，借或貸，金額，摘要)

會計分類帳 (會計科目編號，會計科目名稱，借方或貸方科目，期初餘額，目前餘額)

二、利用OTO資料集模式重新定義相關Table之關係

OTO 資料集模式就是要在交易系統的資料庫裡，為任何相關的兩個 Table 各自建立一組能相互一對一映射的資料集。因此在這個步驟裡，任何相關的兩個 Table 間都必須依照 OTO 資料集模式重新定義其關係。為了產生資料集必須調整或新增 Table 的欄位，在總帳會計交易系統的資料庫裡，「傳票」與「傳票明細」這兩個相關的 Table 間，以現有的欄位並無法各自產生一對一的資料集，也就是說這兩個相關的 Table 間無法產生雙向的資料處理軌跡。為了建立這兩個 Table 間一對一映射的資料集，可以在「傳票」Table 裡新增一個記錄傳票會計科目數的欄位，如果會計科目數欄位的值為 3，即可計算出代表第一筆、第二筆及第三筆會計科目的 1、2 及 3 的運算值，傳票編號的欄位值連上會計科目數的運算值，即可成為資料集裡唯一的元素值。而「傳票明細」Table 則可新增一個會計科目序號的欄位，將傳票編號與會計科目序號欄位的值相連結，則可成為資料集裡唯一的元素值。如此，兩個相關的 Table 即可利用各自資料集的元素間一對一之映射關係，而產生雙向的資料轉換與處理軌跡。

而「傳票明細」Table 的記錄資料則依會計科目編號過帳到「會計分類帳」Table 裡，因此「傳票明細」與「會計分類帳」兩個 Table 間也有關連，但是以現有的欄位並無法各自產生一對一的資料集。為了建立這兩個 Table 間一對一映射的資料集，可以在「傳票明細」Table 裡新增過帳序號欄位，在「會計分類帳」Table 裡新增起始過帳序號與目前過帳序號兩個欄位。利用「傳票明細」Table 裡過帳序號的欄位產生其資料集，再利用「會計分類帳」Table 裡的起始過帳序號與目前過帳序號兩個欄位間的所有過帳序號組成其資料集，則兩個資料集間的元素就會有一對一之映射關係，而產生雙向資料轉換的軌跡。

利用 OTO 資料集來重新定義兩個相關 Table 間的關係時，其欄位的調整與新增並沒有一定的方式，只要能在兩個相關的 Table 間產生一對 OTO 資料集即可。重新定義後的總帳會計交易系統的資料庫結構如下：

傳票 (傳票編號, 傳票日期, 交易憑證編號, 會計科目數)

傳票明細 (傳票編號, 會計科目編號, 借或貸, 金額, 摘要, 會計科目序號, 過帳序號)

會計分類帳 (會計科目編號, 會計科目名稱, 借或貸方科目, 期初餘額, 目前餘額, 起始過帳序號, 目前過帳序號)

三、產生資料集及建立OTO之關係

OTO 資料集並沒有真正儲存於交易系統的資料庫裡，而是由資料庫裡表格的欄位或欄位的運算式所產生。為了說明如何產生 OTO 資料集，因此必須使用一些測試或虛構的資料來討論其過程。假設有甲分公司與乙分公司，每個分公司都會有一些交易事項，各自依其交易性質與項目，直接從網路的總帳會計交易系統裡產生或輸入傳票資料，而在總公司則可應用 OTO 資料集之模式，隨時來查核分析整個交易系統之活動。表一則為甲與乙兩分公司某個期間所發生之交易事項與所需之會計分錄。

表一 甲與乙分公司交易事項與所需會計分錄表

分公司別	交易憑證編號	交易日期	交易事項	傳票編號	會計分錄
甲分公司	A87010001	87.01.02	賒銷貨品 \$80,000 予周久公司	87010001	借：應收帳款 80,000 貸：銷貨收入 80,000
乙分公司	B87010001	87.01.06	賒銷貨品 \$60,000 予九洲公司	87010002	借：應收帳款 60,000 貸：銷貨收入 60,000
乙分公司	B87010002	87.01.12	現銷貨品 \$10,000 與安和公司	87010003	借：現金 10,000 貸：銷貨收入 10,000
甲分公司	A87010002	87.01.12	支付上個月之水電費 \$7,600 及伙食費 \$2,300	87010004	借：水電費 7,600 伙食費 2,300 貸：現金 9,900
甲分公司	A87010003	87.01.16	收到 87.01.02 賒銷 予周久公司之貨款	87010005	借：現金 80,000 貸：應收帳款 80,000
乙分公司	B87010003	87.01.18	收到 87.01.06 賒銷 予九洲公司之貨款	87010006	借：現金 60,000 貸：應收帳款 60,000

而這些交易資料輸入總帳會計交易系統後，其資料庫裡表格的內容會如表二所示，每一筆交易會有一張傳票，所以「傳票」Table 裡會有六筆記錄，而六張傳票共有十三筆會計科目的資料，所以「傳票明細」Table 裡就會有十三筆記錄，而「會計分類帳」祇列出傳票使用到的五個會計科目。

表二 傳票 Table 之內容

傳票編號	傳票日期	交易憑證編號	會計科目數
87010001	87.01.02	A87010001	2
87010002	87.01.06	B87010001	2
87010003	87.01.12	B87010002	2
87010004	87.01.12	A87010002	3
87010005	87.01.16	A87010003	2
87010006	87.01.18	B87010003	2

表三 傳票明細 Table 之內容

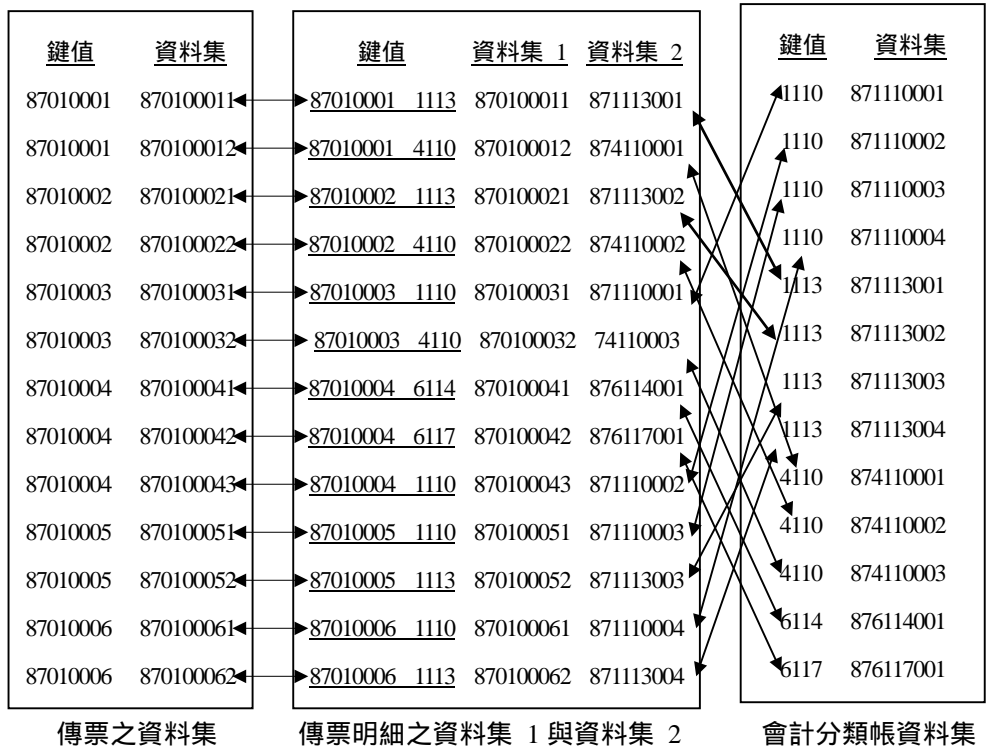
傳票編號	會計科目編號	借或貸	金額	摘要	會計科目序號	過帳序號
87010001	1113	借	80,000	A87010001 交易之應收帳款	1	871113001
87010001	4110	貸	80,000	A87010001 交易之銷貨收入	2	874110001
87010002	1113	借	60,000	B87010001 交易之應收帳款	1	871113002
87010002	4110	貸	60,000	B87010001 交易之銷貨收入	2	874110002
87010003	1110	借	10,000	現金存入彰銀 001567989 帳號	1	871110001
87010003	4110	貸	10,000	B87010002 交易之銷貨收入	2	874110003
87010004	6114	借	7,600	86 年 12 月之水電費	1	876114001
87010004	6117	借	2,300	86 年 12 月之伙食費	2	876117001
87010004	1110	貸	9,900	台銀支存 11287869 號支票	3	871110002
87010005	1110	借	80,000	現金存入交銀 223113799 帳號	1	871110003
87010005	1113	貸	80,000	A87010001 交易之貨款收入	2	871113003
87010006	1110	借	60,000	現金存入彰銀 001567989 帳號	1	871110004
87010006	1113	貸	60,000	B87010001 交易之貨款收入	2	871113004

表四 會計分類帳 Table 之內容

會計科目編號	會計科目名稱	借或貸方科目	期初餘額	目前餘額	起始過帳編號	目前過帳編號
1110	現金	借	888,000	1,028,100	871110001	871110004
1113	應收帳款	借	0	0	871113001	871113004
4110	銷貨收入	貸	0	150,000	874110001	874110003
6114	水電費	借	0	7,600	876114001	876114001
6117	伙食費	借	0	2,300	876117001	876117001

利用資料庫裡三個 Table 的欄位資料，則可產生如圖八之 OTO 資料集。「傳票」Table 之資料集的元素是由傳票編號的欄位連上會計科目數欄位的運算值，譬如傳票編號為 "87010004" 之傳票記錄，其會計科目數的欄位值為 "3"，那麼就會產生 1、2 與 3 三個運算值，連結後則會產生 "870100041"、"870100042" 及 "870100043" 等三個資料元素。「傳票明細」Table 必須要建立兩個資料集，資料集 1 是要與「傳票」Table 之資料集產生一對一的映射關係，其是由傳票編號連上會計科目序號所組成；而資料集 2 是要與「會計分類帳」Table 之資料集產生一對一的映射關係，其是由過帳序號之欄位值所組成。而「會計分類帳」Table 之資料集是由每個會計科目之起始過帳序號與目前過帳序號間之所有過帳序號所組成。

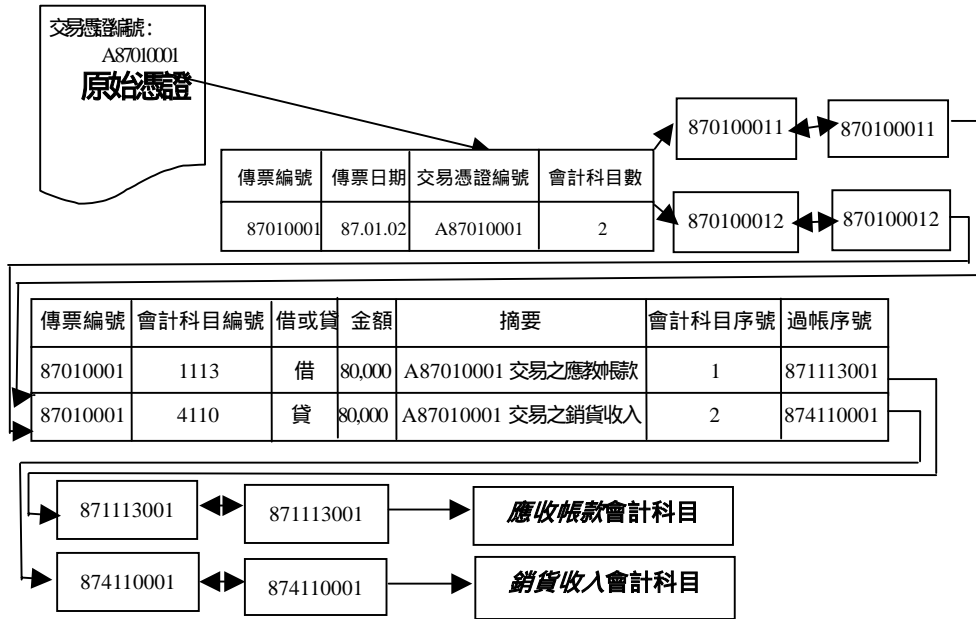
由於「傳票」之資料集與「傳票明細」之資料集 1 間的元素皆互有相同的值，因此可以直接利用相互間的元素值來建立一對一的關連。而「傳票明細」之資料集 2 與「會計分類帳」之資料集間的元素也皆互有相同的值，因此也可以直接利用相互間的元素值來建立一對一的關連。在本例子裡，建立資料集間元素一對一的關係，皆不必經由函數運算式的運算來達成。



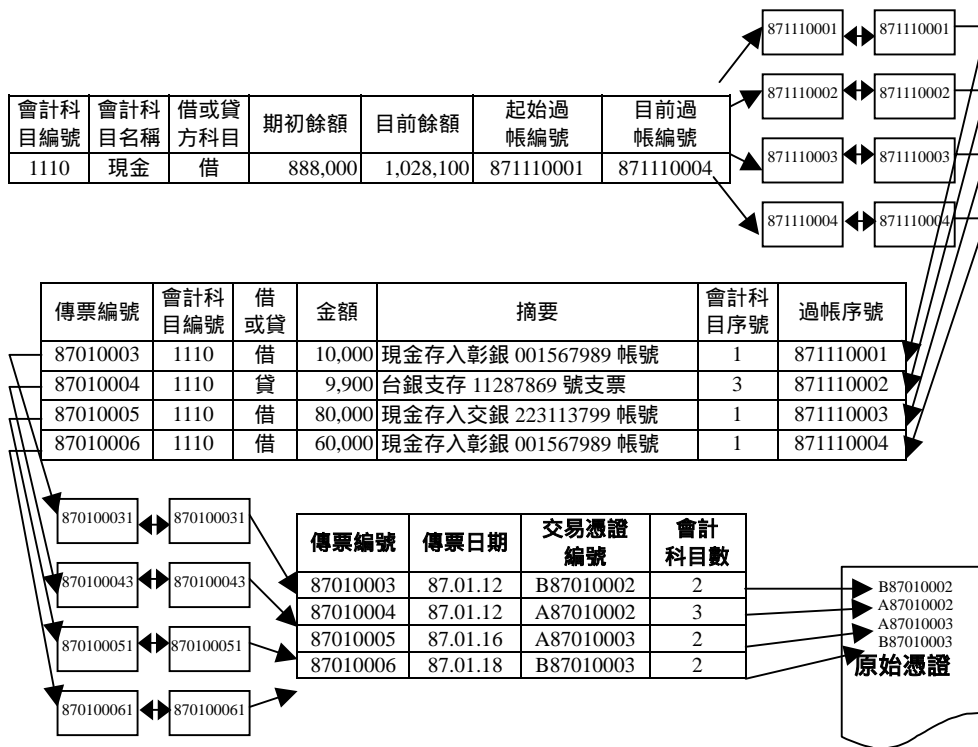
圖八 總帳會計交易系統之 OTO 資料集

四、驗證交易系統是否有清楚完整之稽核軌跡

透過兩個相關的資料集間元素一對一的映射關係，再利用資料元素的鍵值則可間接映射到其 Table 的記錄上。從步驟 3 所產生之 OTO 資料集裡，我們可以找到如圖九從原始憑證到「會計分類帳」科目之正向稽核軌跡。從交易憑證編號為“ A87010001 ”的原始憑證追查到「傳票」Table 裡，再從「傳票」與「傳票明細」兩個 Table 的 OTO 資料集之"870100011"與"870100012"兩個元素追查到兩筆傳票明細記錄，再從「傳票明細」與「會計分類帳」兩個 Table 的 OTO 資料集之"871113001"與"874110001"兩個元素追查到「會計分類帳」Table 的應收帳款及銷貨收入科目。



圖九 從交易憑證到會計分類帳科目之正向軌跡圖



圖十 從現金科目到交易原始憑證之逆向軌跡圖

相反地，從 OTO 資料集裡，我們也可以如圖十找到從「會計分類帳」Table 裡現金科目到原始憑證之逆向稽核軌跡。從「會計分類帳」與「傳票明細」兩個 Table 的 OTO 資料集之"871110001"、"871110002"、"871110003"與"871110004"等四個資料元素逆查到「傳票明細」Table 的四筆明細記錄，再從「傳票明細」與「傳票」兩個 Table 的 OTO 資料集之"870100031"、"870100043"、"870100051"與"870100061"等四個資料元素逆查到「傳票」Table 的四筆傳票記錄，再從每筆傳票記錄的交易憑證編號逆查到交易發生的原始憑證。

從現金科目到交易原始憑證之逆向稽核軌跡裡，我們也可以得到如表五之現金科目交易明細表，而得以查知所有明細交易之詳細活動。

現金科目交易明細表		
	87.01.01	87.01.18
期初餘額 (87.01.01)		888,000
乙分公司現銷收入 (87.01.12)		10,000
甲公司支付電費及伙食費 (87.01.12)		9,900
收到 87.01.02 賒銷予周九公司之貨款	80,000	
收到 87.01.06 賒銷予九洲公司之貨款	<u>60,000</u>	
餘額 (87.01.18)		1,028,100

表五 現金科目交易明細表

陸 OTO資料集模式之施行與適用

一、OTO資料集模式之施行

利用 OTO 模式所產生的資料集是虛擬的資料，其資料元素其實是由資料庫裡表格的欄位或欄位的運算式所產生，並不一定要真正儲存於資料庫裡。不過，在各種類型交易系統的應用上，也可能會因為特殊的情況或需要，而將 OTO 資料集及其相關的表格資料另外儲存在磁碟檔案裡，因而會佔用了大量的磁碟空間。當然，在利用程式模組產生 OTO 資料集來執行查核分析工作時，是會佔用電腦大量的主記憶體空間與 CPU 時間，這是施行 OTO 模式所需負擔的一定成本。但在電腦硬體價格持續下降，而電腦 CPU 執行的速度卻更快的情況下，將施行 OTO 資料集模式所擔負的成本與所產生的效益互相比較之下，其所需的成本就顯得微不足道了。

OTO 模式的施行除了須重新定義資料庫裡任何兩個相關表格間的欄位外，就是撰寫產生 OTO 資料集及所需查核分析之程式模組，其程式模組可以放在支援交易活動的資訊系統裡，也可以自成單獨的一套稽核系統，以便利用其所產生的資料集之資料軌跡，來追查與分析交易系統活動的過程，並做為交易系統控制、考核與管理之用。

由於 OTO 模式應用於交易系統時，其施行的方法與步驟清楚容易，且所需成本不高，因此其實際的施行與應用是可行的。

二、OTO 資料集模式之適用

OTO 模式是利用關連式資料庫裡相關的兩個表格間之欄位，來分別產生其資料元素間能一對一相互映射的資料集，以間接建立兩個表格間資料處理的雙向軌跡。因此任何型態的交易活動所產生的資料祇要記錄到資料庫後，即可依此模式產生虛擬之資料集，而建構了交易系統清楚完整的稽核軌跡。由於 OTO 資料集模式是從整個關連式資料庫裡的資料來建構其交易系統的稽核軌跡，所以 OTO 資料集模式適用於任何利用關連式架構來建立其資料庫之交易系統。

柒 結論與建議

在人工作業的環境裡，稽核軌跡是由一組文件憑證、帳簿與報表上的參照編號所構成。透過這些參照編號，稽核人員可以很容易地發現錯誤與舞弊，資料處理人員可以迅速地更正錯誤的交易資料，而管理人員也可即時地取得所要的資訊。因此祇要妥善地設計文件帳簿上的參照編號，交易活動的軌跡即可清楚地記錄於其文件帳簿上。但資料處理的環境從人工作業到電腦化作業後，交易系統的稽核軌跡不但無法用肉眼直接在可能分散於各處的電子資料裡查看，而且也因為資料庫設計原理的某些特性，而使其變得越來越模糊不清。所以本論文利用反函數的理論，提出一個 OTO 資料集模式，來建立關連式資料庫裡相關表格 (Table) 間雙向的資料軌跡，以增強關連式資料庫設計時對稽核軌跡之疏忽，而為 IT 環境下建構交易系統清楚且完整的稽核軌跡之依據。

OTO 資料集模式最主要的功能，就是在資訊技術的環境下提供交易系統一個清楚且完整的稽核軌跡，而使得查核、控制與管理交易系統的整個活動過程變得更容易與可能。也由於其能產生交易系統清楚且完整的稽核軌跡，而使得交易系統的查核工作能順利地進行，因而降低了昂貴的查核費用。

要建立一套企業組織交易活動的自動稽核系統, 其前題是要能取得交易系統清楚且完整的稽核軌跡。由於 OTO 資料集能即時產生交易系統清楚且完整的稽核軌跡, 因此本研究建議有興趣的學者專家, 可以研究利用 OTO 資料集模式來建構企業組織交易活動之自動稽核系統 (Automated Auditing Systems)。

參考文獻

吳琮璠, 「會計資訊系統與電腦審計」, 台北: 智勝文化公司, 1997 年。

葉誌崇, 「會計資訊系統」, 台南: 自行出版, 1997 年。

Arens, Alvin A. and Loebbecke, *Auditing-An Integrated Approach*, New Jersey: Prentice Hall, 1994.

Bieber, Douglas W., "Home Banking Security and Audit", *Journal of Accounting & EDP*, Vol.3, Spring, 1987, pp.27-35.

Brown, Nander, Jr., "Minicomputers Control, Security, and Audit", *Internal Auditor*, Vol.40, Feb., 1983, pp.39-42.

Cerullo, Michael J., "Audit Trail Controls for Online Real Time Computer Systems", *Cost & Management*, Vol.57, No.6, Nov/Dec, 1983, pp.16-19.

Crump, Deryk, "On the Computer Audit Trail," *Accountancy*, Vol. 92, October, 1981, pp.63-68.

Cushing, Barry E. and Romney Marshall B., *Accounting Information Systems*, New York: Addison-Wesley, 1993.

Davis, Joyce E., "Safeguarding Your Network", *Black Enterprise*, Vol.27, October, 1996, pp. 48-50.

Gallegos, Frederick and Basica, Daniel, "Microcomputer Security: Audit problems and Solutions", *Journal of Accounting & EDP*, Vol.1, Winter 1986, pp.49-56.

Gallegos, Frederick and Bieber, Douglas W., "Emerging Technology and Information Systems Auditing," *Journal of Accounting & EDP*, Vol. 3, Summer, 1987, pp.47-56.

Gelinas,Ulric J., Oram, Allan E. and Wiggins, William P., *Accounting Information Systems*, Cincinnati: South-Western, 1993.

Hinde, Stephen, "Approaching the First-time Computer Audit", *Accountancy*, Vol.90, May, 1979, pp.77-81.

Hinde, Stephen, "The Control Total Approach to Computer Auditing," *Accountancy*, Vol.95, August, 1984, pp.100-104.

Kolman, Bernard and Busby, Robert C., *Discrete Mathematical Structures for Computer Science*, New Jersey: Prentice Hall, 1987.

Mcfadden, Patrick James, "Guarding Computer Data," *Journal of Accountancy*, Vol.184, July, 1997, pp.76-79.

- MenKus, Belden, "How an 'Audit Trail' Aids in Maintaining Information Integrity as Illustrated in Retailing", *Computers & Security*, Vol.9, April, 1990, pp.111-116.
- Silverman, Richard A., *Calculus with Analytic Geometry*, New Jersey: Prentice Hall, 1985
- Skudrna, Vincent J., "The Status of EDP Auditing", *CPA Journal*, Vol.52, March, 1982, pp.26-31.
- Truss, J. K., *Discrete Mathematics for Computer Science*, New York: Addison-Wesley, 1991.
- Waples, Elaine and Norris, Daniel M., "The Impact of Just-In-Time on the Audit of Purchasing", *Journal of Purchasing and Materials Management*, Fall, 1989, pp.26-31.
- Watne, Donald A. and Turney, Peter B. B., *Auditing EDP Systems*, New Jersey: Prentice Hall, 1990.
- Weber, Ron, "Audit Trail System Support in Advanced Computer-Based Accounting Systems", *Accounting Review*, Vol. 57, April, 1982, pp.311-325.
- Wilkinson, Joseph W. and Cerullo, Michael J., *Accounting Information Systems*, New York: John Wiley & Sons, 1997.

An OTO Data Set Model for Audit Trails Construction of Transaction Systems

TIAN-LIH KOO

Department of Accounting, Shih Chien University

ABSTRACT

The clear and complete audit trails of a transaction system are important to auditing, management and control. In the environment of manual operation, the data created during the transaction process are all written down on papers and account books, in other words, the audit trails are composed of a group of reference codes on the original documents, books and reports. Hence, all the processing trails are visible and checkable. However, after the data processing environment has been changed from manual operation to computerized operation, the audit trails have become not only invisible due to the electronic data, but also more ambiguous due to the special characteristics of database design principle. This research proposes that the one-to-one mapping relation of the elements of two sets of data can be built upon the theory of inverse function; and from this relation, we can get, thus, the two-way transformation trails from these two sets of data. By using the one-to-one mapping characteristics of inverse functions, this research presents an OTO data set model to build a relation among related tables in the relational database. The OTO data set model will function as the basis for audit trails construction of transaction systems in the environment of information technology.

Keywords: audit trails, OTO data set model, transaction systems, information systems, relational database

