

行政機關建立資訊系統

第一年前半期報告——辦公室作業流程模型

郭德盛	洪永常 ¹
何正信 ²	何鈺威 ³
楊光中	程光遠 ⁴

贊助單位：行政院研究發展考核委員會
 執行單位：中央研究院資訊科學研究所

- 1 中央研究院資訊科學研究所
- 2 國立台灣大學電機研究所研究生
- 3 國立交通大學計算機研究所研究生
- 4 中央研究院總辦事處

中研院資訊所圖書室



3 0330 03 00006 6

中華民國 七十 年 九 月

0006

FOR REFERENCE

 NOT TO BE TAKEN FROM THIS ROOM

書	考	參
借	外	不

1 導 言

1.1 辦公室自動化的緣起

大幅增加辦公室的生產力，一直是機關或企業機構追求的目標。工業革命帶給工廠及農場的機械化，曾大幅增加了工廠及農場的生產力。惟對辦公室的影響，却薄弱未名。究其原因，不外工廠與農場的作業大致爲“不變性”極高，較易施行機械化的例行工作，而且大部份是體力勞動，較易以不需智慧或智慧較低的機具來替代，而辦公室作業型態則偏向腦力的運作，其“變性”成份較高，因此高智慧型機具（如計算機）之運用似乎無可避免。

傳統上，增加辦公室的效率，大致有下列方法：

純人工時代：

- 1 提高職員的工作情緒（如獎金、休假等）以增加生產力。
- 2 設計規格化表格，簡化作業程序，簡化資訊流程，以增加作業處理的速度，提高效率。

機械時代：

- 3 添置資料處理機具（或稱工作機具），諸如打字機、影印機等，以增加資訊處理速度。
- 4 添置傳輸設備，諸如電話、電報等，取代信差遞送資訊，以縮短資訊傳輸時間。

此等方法，對提供辦公室效率，確實有所貢獻。惟自計算機問世至應用於工廠與農場後，工廠生產力急劇增加，相形之下，辦公室生產力落後愈甚，致有成爲整個生產體系的瓶頸的趨勢。美國在 1965 至 1975，十年之內，電子業的工廠生產力年增加率爲 3.6%，而“白領階級”（亦即辦公室工作者）則僅及 2.9% [2]，便是最明顯的例子。處此工廠生產力因計算機的應用而長足進展之際，適當運用計算機以提高辦公室的效率，乃是刻不容緩的事。

1.2 辦公室自動化的意義

辦公室自動化乃是指適當運用計算機，使辦公室之作業執行更具效率，亦即利用計算機以提高辦公室之生產力。其涵義如下：

- 1 汰換辦公室之工作機具，提高其文書處理速度。諸如打字機汰換成

電傳打字機，或文字處理器 (Word Processor) ，以及計算機之引用等。

- 2 汰換辦公室之傳輸機具，提高其文書傳輸速度。諸如電報汰換成電報交換，或電郵系統 (Electronic Mail system) 等。
- 3 更新人機界面，使人更易於使用機械，以處理文書（如編輯文書、表格、歸檔、填表等）及輸送文書。如全錄的 Office-Talk-Zero 界面 [5]，其電視畫面與鍵盤設計，甚適人工操作。再如 OBE (Office-Procedure By Example) [12] 語言及 BDL (Business Definition Language) [6] 語言均是朝此方向設計的例子。
- 4 隨著企業機構的擴展，辦公室逐漸漸分居各處。原來在同一辦公室內可完成之作業，此時或需流經數個“分處”方得完成。若此數個分處散居範圍不甚廣闊（如 10 公里以內），則彼此間常需設“區域網路” (Local Network) 藉資通訊。建立區域網路亦是自動化的部份。
- 5 跨越區域網路（大於 10 公里者）之外，辦公室之間端賴“全區網路” (Remote Network) 聯繫。有效運用全區網路（一般而言，由電信單位提供），亦是自動化的一環。
- 6 辦公室作業流程自動化。亦即利用計算機將辦公室之作業流程，予以自動化。一般而言，大部份辦公室作業流程皆有一定的程序。此“既定”程序，正可提供我們設計自動化作業流程的依據。
- 7 辦公室管理自動化：辦公室運轉期間，常因辦公室本身業務之更動擴展或外界情況之改變而需作人員重組、機具更新以及作業內涵之更動等行爲。其是否變動以及變動之大小，則須研判辦公室現狀以及手頭資料後作一決定。辦公室管理自動化即提供辦公室管理人員此等決策能力，並且作實際上之辦公室重組（如作業更動、人員重組、機具更新）的工作。因此，平時管理自動化須能監理辦公室現有人員、機具與作業之運轉現狀；並提供足夠的資訊給管理人員作決定變更辦公室結構與否的依據。於決定更動後，管理自動化將執行人員重組、機具重配、以及作業更動的工作，諸如新作業之界定，人員之指派、作業時間之安排，作業之時序或並進編排等。此步驟，有賴於智慧型系統 (Knowledge-based System) 的協助，實是進入辦公室全自動化的一大要素。

從上所述，我們可將辦公室的自動化分成下列三個層面：

- 1 作業機具之添置與汰換（即上列 1-5），以增加資訊處理的數量與速度，並提高資訊傳輸之速度和正確性。統稱之為作業機具之自動化。
- 2 利用計算機作作業流程自動化（即上列 6），使作業流程循既定程序進行，以減少人工干預。一則使錯誤減至最低，再則因作業機具之更新獲取作業流程時間之縮短。
- 3 利用計算機作辦公室管理自動化（即上列 7），使辦公室資源之管理（包括人事管理、機具管理、作業管理以及決策管理。）適當地反映出辦公室的運轉品質與成長情況。

1.3 辦公室自動化的方法

從傳統的辦公室進展到電子化辦公室（Electronic office）之過程即為辦公室自動化，這是一套“漸進”（evolutionary）的過程。此過程應該考慮電子化辦公室所帶給社會結構、人際關係、以及職員心理的可能衝擊，即應儘量設法減低此種衝擊的不利層面。另外還須以“全自動化”辦公室為目標。可謂“全自動化”辦公室，最主要的特色是決策管理自動化。決策管理在諸辦公室資源管理中最缺乏固定結構，亦即較難以計算機來加以取代。利用智慧型系統來提供決策管理所需的資訊似屬必要。因此在可預見的將來智慧型系統在辦公室自動化的舞台上，將扮演一個要角。

在全自動化辦公室的社會裡，人的價值將更形突出，也更形重要。因為藉著全自動化辦公室提供的資源，個人“創造力”的發揮將可達於至善境地。於是人將可盡力從事創造力的工作，而把低創造力或者例行公事全交給機器處理。

自動化漸進步驟如下：

步驟 1 為作業機具自動化。即逐漸汰換與添置作業機具，包括資料處理、機具之汰換與傳輸設備之添置等。此步驟使人免於單調繁瑣的工作。這些工作有些需要密集勞力的（如大量資料的處理），有些則根本缺乏生產價值（如資料遞送所佔時間）。

步驟 2 為作業流程自動化。對於固定的作業型態，辦公室的作業流程往往

是固定的。這些固定流程時常是資訊傳輸與資訊處理兩事交互組合並間以人工干預。然而人工干預最明顯的弱點却是易於犯錯，以及公文積壓。本步驟即在去除此不必要之人工干預，以使錯誤減至最低，並使作業流程時間縮至最短。

步驟3 辦公室管理自動化。此步驟最需要智慧型系統的協助。因為辦公室管理所需資料，往往需從各辦公室，甚至於其他企業機構獲取 [8]；更甚者，資料的來源亦不僅限於資料庫，舉凡報章、雜誌，飯後談話資料，均可能刺激新的觀念與新的決定。因此，如何有效利用自動化辦公室來獲取這些資料，並用以決定政策，變更辦公室結構，使辦公室適當地成長，均是此步驟的目標。職是之故，對於固定作業流程的作業型態而言，辦公室管理提供與時俱移的流程更新能力。對特別用途之作業型態，辦公室管理則提供作業界定分配與品質管理等能力。此步驟實是進入全自動化辦公室的關鍵棋子。

辦公室自動化與辦公室的型態息息相關。不同的辦公室型態，其自動化將着重在不同的步驟。舉例而言，商業辦公室作業一般偏重於大量的資料處理，以及決策的分析與決定。因此自動化時便偏重在資料處理機具的汰換與資源管理的自動化。一般行政機關的作業則偏於繁複的資料傳遞流程，因此自動化時較偏重於資料傳輸設備的添置與其繁複流程的自動進行。因此進行自動化的三步驟前，須先對辦公室的特性、型態，做一番分析與瞭解。

其次針對這些特性，我們須得找出一個適當的模型來模塑此一辦公室。此一模型至少需能正確地反映出該辦公室型態的特性，並能適用於自動化的任一步驟。

一個良好的模型，除需能忠實地反映出既存辦公室的特性外，尚需簡單易懂，並協和不矛盾 (consistency)。如此在設計辦公室資訊系統時，才不致因不忠實而使結果與現實大相逕庭；因不協合而使設計錯誤百出；及因不簡易而事倍功半。再者，模型須有能力呈現既存辦公室內不合理與不適當的作業特性，並藉以修改辦公室的某些特性，以達成高生產力的目標。如果模型無法有效地適用於各類型辦公室時 [9]，亦應針對各目的辦公室型態，提出最適當的模型。

最後，根據該模型，並藉用一套設計方法，我們便可依各型辦公室之

特性，設計其辦公室自動化系統（或稱辦公室資訊系統）。該系統將反映出自動化的三步驟，並且有效提高辦公室的生產力，而辦公室的自動化亦於焉底成。

本文在詳述行政機關的特性（第二節）後，將提出一模型以模塑此一型態的辦公室（第三節），並對此模型作一簡要分析（第四節）。第五節為本文總結。

2 辦公室的特性

2.1 辦公室的結構

辦公室型態的不同，在於其所負任務之不同。如會計室司會計、運輸處司運輸工作。不同任務，由不同組合的人員在不同的工作環境下來完成 [2]。任一任務大抵可分成兩部份：其一為任務本身作業的執行，另一則為任務的管理。人員的配置也因而大略分為兩層：一為職員（真正執行任務者），一為管理人員（任管理職責者）。管理人員與職員的關係一般而言是縱向的，或說是樹狀結構的。職員在執行任務時的工作環境，至少需有資料庫（或儲存資料檔案的地方），及工作機具來完成作業（如紙筆、計算器、計算機等），並須有傳輸機具（如電話、擴音器等）來聯繫辦公室內外之其他成員。這些成分（任務，人員及工作環境）即構成一辦公室的雛型。圖一為其簡圖。

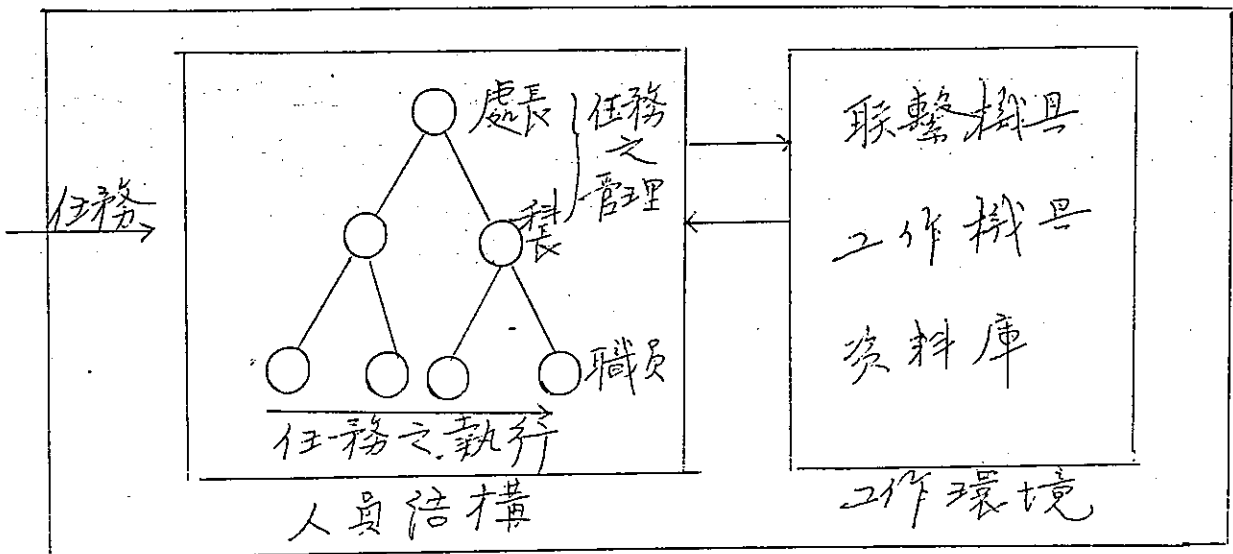


圖1 辦公室結構

圖 1 中的人員配置顯示其縱向階層的樹狀結構。其中處長及科長為管理人員。

分層負責的觀念使得作業在辦公室內有一定的流向。譬如任務的到來，一般而言由收發室簽收，交由辦公室最高人員（如處長）審閱，並由其簽註請某個職員（或其幾個職員）辦理，此稱作業指派。職員在處理作業

時，有些部份往往須請其直接主管（譬如科長）批核簽章，以示分層負責的精神。此等流程，有的純與人員結構有關而與作業型態無關，有的則與人員結構和作業型態均有關係。觀念上，我們稱只與人員結構有關的作業流動方向為辦公室程序 (office procedure) 而稱與作業型態有關的作業流動方向為作業流程 (workflow) (詳見 2.3節)。另職員在執行作業時，須用及之工作環境內的作業機具（稱機具派用），則包含工作機具及傳輸機具。

2.2 作業時特性

首先我們分析作業的執行情況，發現作業有依“分頭並進法”進行，也有依“依序前進法”進行。詳言之，一作業中可劃分出數個與時序無關的動作 (activity) 時，這等動作，即可由一職員派用多個機具同時處理，或分由多個職員使用不同機具一齊處理，此稱分頭並進法。前者稱為“單人多機並進法”，後者則稱為“多人多機並進法”。如一作業中須劃分出某數個與時序有關的動作時，這些動作的執行即是“依序前進法”。所謂時序關係意指時序在後的動作須俟時序在前的動作完成後，方可執行。動作之劃分，分頭並進及依序前進等即構成作業的特性。圖 2 說明作業的特性。其中的單人單機動作一般指從辦公室的資料庫內，存、取資料，或對資料作更換、膾改、計算（利用計算機），或傳送（利用傳輸機具）等處理。其中 a, b, c, d 表職員。

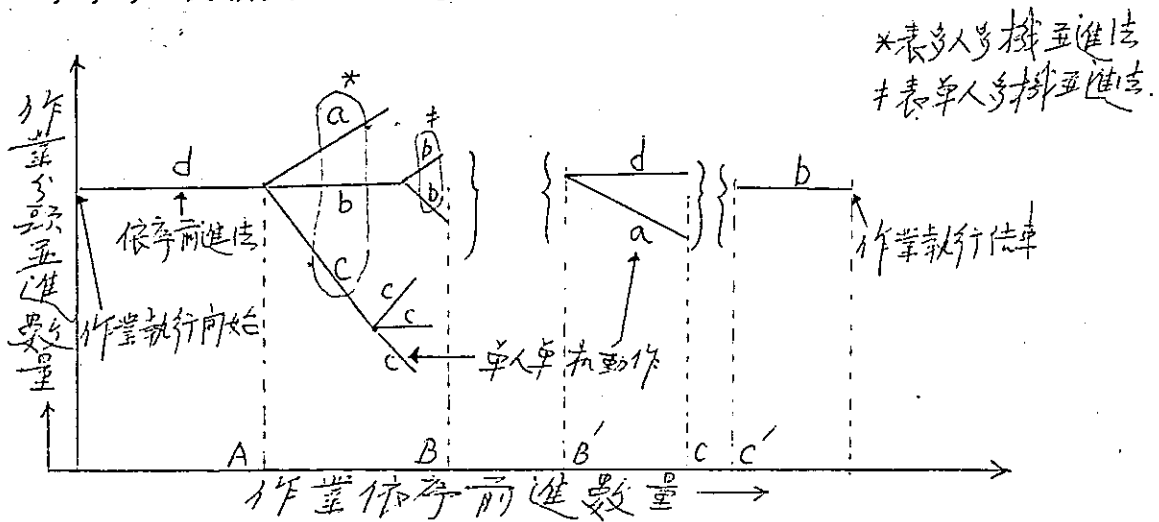


圖 2: 作業特性

2.3 作業的流程

上小節，尚未探討作業放到一個辦公室裡頭，尤其是具有縱向人員配置的辦公室結構裡頭的執行情形。本小節即論述此種狀況。我們會說作業的真正執行者均為職員，而職員是在縱向人員配置的葉端，因此作業的執行可說是橫向的。然而誠如 2.1 節所言，辦公室流程規定了部份與人員結構有關的作業流動方向。如此一來，本來極單純的作業執行順序，就需加入可能有的縱向流動，因此就構成了所謂的與作業型態有關的流程，亦即作業流程。這作業流程不只說明了作業真正執行的順序，並且也涵蓋了作業批核、簽呈等等的辦公室程序。

在 2.1 節中說及作業指派時，曾假設處長直接指派作業給職員。今若考慮人員結構時，則應是處長先視各科職責之不同，指派予科長，該科長再詳視作業的內容，並依各職員所任工作性質的不同，分解該作業成若干“分頭並進”的動作，及若干“依序前進”的動作，以之指派予各職員去執行。作業真正執行時，我們將發現資訊會在動作與動作間活動，這種現象稱為資訊流程。從作業的觀點來看動作與資訊之此種關係則稱“作業輾進”。作業或動作的完成均與時間有關（圖 1），亦即與時推移而漸趨於完成，此種與時推移的現象，稱為作業的時間分配。

至此，辦公室內的作業流程已相當清楚。隨著科長所交予之作業型態的不同，作業的分解與指派，職員的指派，資訊的流程和時間的分配，必然不同。但某些工作（如作業分析、指派、時間分配、辦公室流程等）却都是固定存在的。這些提昇出來的工作屬於部份的辦公室管理工作。辦公室管理中尚有一種作業品質管理，亦即藉著隨時監裡偵查各動作的執行結果與預定目標間的差距（稱作業偏差），來校正正在進行中的作業。

把作業的執行放到辦公室內，我們發現科長的地位相當突出。因為他不只負責作業的指派，還需相當瞭解作業的內容和各職員的工作性質，並配合辦公室流程作作業的分解；除對整個作業的輾進與作業時間的分配有一定的認識外，尚需能控制作業品質的水準。職員則祇對各動作的時間分配有其概念即可。這麼一來科長的工作就很像部份辦公室管理的工作了。

基於一般辦公室的作業型態 (work type) 是固定而且有限的，因此在自動化時，我們考慮以作業型態為基礎來完成自動化的第二步驟與第三步驟（部份）。換言之，將辦公室程序與部份辦公室管理容成兩部份：其一視為與作業型態無關的部份，包含作業指派、品質管理等，提昇為一共同

監理裝置，以處理全部作業。另一則視為與作業型態有關的部份，即將作業的分解、輾進、時間的分配、機具的派用、職員之指派，以及辦公室的程序等，合成隨不同作業型態而有不同內容的作業流程。若然，對辦公室內每一固定之作業型態，將有一固定之作業流程與之相對。一旦有例外之某一作業型態出現時，則需利用「共同監理裝置」與「辦公室程序」，定出一組臨時使用之作業流程。俟此種作業多至納入固定作業時，方予以標準化，亦即方有一標準作業流程與之相對也。

本文旨在提出進行辦公室自動化第二步驟的方法，其第三步驟（含上述之「監理裝置」）亟須智慧型系統的協助，則暫不在論述之內。

3. 作業流程模型

上節我們分析了辦公室及其作業的特性，本節擬提出一模型來模塑辦公室之作業流程。

3.1 模型定義

如前節所述，一種作業之流程係一群動作依某種特定關係組合而成。模塑作業流程即是將各動作間之關係用某種方式表示出來。

參考圖 2，作業在 A 點是以“分頭並進”方式進行；在 B 與 B' 點（或 C 與 C' 點）之作業需要“協調”相互獨立之動作後，再進行時序在後之動作。B' 之起動，顯然須俟各分頭並進動作全部完成之後才發生。動作之起動（開始執行）與否，除了需達到某種狀態（如 B' 之起動須各分頭並進之動作達於“完成”狀態）外，尚須等到某一時機才發生。例如 B' 動作本可在作業流程達 B 點後，立即開始，但若吾人設定 B' 動作須俟流程達 B 點後 2 小時方開始起動，則這 2 小時之設定條件即是 B' 動作起動之“前提”。另外，“決策”亦是辦公室作業執行時常見的動作之一：設在某些狀態下，A 動作與 B 動作均可起動，但囿於某種因素，祇能選擇其中一個執行。則動作 A 和 B 間，即存在著因決策所致之“互斥”現象。最後，動作的執行常與資料檔發生關係。例如作業執行時可能用及某資料（稱輸入資料），亦可能產生新資料（稱輸出資料）。因此，模塑一作業流程，事實上即是將上面所述之關係調理出一個頭緒來，並且用某種工具適當地表現出來。

職是之故，作業流程之模塑工具，勢需有效並能簡易地反映上述現象才行。Petri 網路 [10] 提供了大部份的模塑能力。以下我們將先簡述 Petri 網路之特性，然後對 Petri 網路作某些增修，以期能適當地描述辦公室的作業流程。

3.1.1 Petri 網路簡介

• Petri 網路是模塑資訊流程的一種有力工具 [10]，其特有的特性，觀念與技巧，使其在描述與分析系統之資訊或控制流程上，顯現自然，簡易與強有力的特色，尤其是在具有“分頭”與“並進”事件之系統上，更具威力。

圖 3 為一 Petri 網路的簡單圖示。圓圈（或節點）稱為處所 (place)，

橫槓稱為轉化 (transition)。兩者間用實箭頭 (arc) 相連，以表示兩者之關係。圖 3 中， P_1 是 t_1 的一個輸入處所，而 P_2 與 P_3 則是 t_1 的輸出處所。

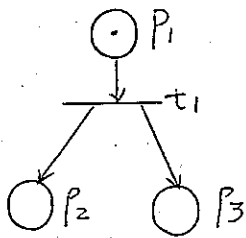


圖 3 (a)

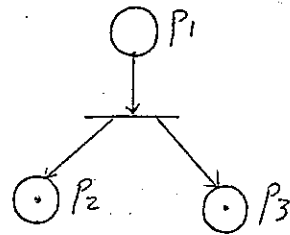


圖 3 (b)

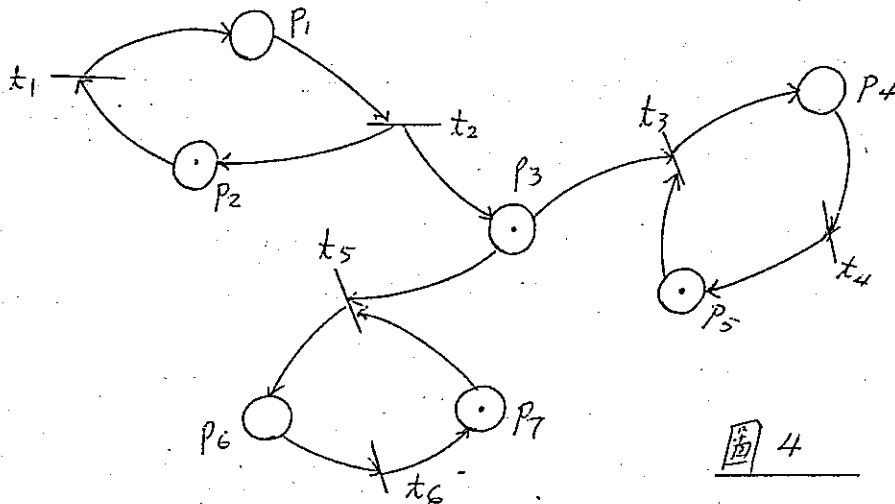


圖 4

Petri 網路之執行由標記 (token) 之移動表示之。圖 3 中以小黑點表標記。含標記之 Petri 網路稱標記 Petri 網路。轉化的起動 (fire) 促使標記移動，當一轉化之所有輸入處所均含標記時，則稱此轉化呈待動狀態。轉化起動後，即移開所有輸入處所之標記，另產生新標記於其所有輸出處所。例如圖 3 (a) 中， P_1 擁有標記所以 t_1 呈待動狀態。若 t_1 起動，則圖 3 (a) 之 Petri 網路呈如圖 3 (b) 之狀態。

標記在 Petri 網路內各處所之分佈情況，稱為 " 狀態 (marking) "。狀態之改變端賴轉化之起動。在不同狀態下 Petri 網路應有不同之轉化呈待動狀態。如圖 4 中之狀態為 $(P_1, P_2, P_3, P_4, P_5, P_6, P_7) = (0, 1, 1, 0, 1, 0, 1)$ ，而此時 t_1, t_3, t_5 均呈待動狀態。因此，下一個起動之轉化，將是 t_1 和 t_3 或 t_1 和 t_5 。

歷來 Petri 網路用之於模塑分頭並進事件系統時，主要在模塑其事件與條件兩者之輾進關係 [1, 7, 10]。通常在某一時刻，系統上某些條件的成立，可能引起某些事件之發生。此些事件之發生又可能改變系統之狀態而引起其他條件成立也。今欲用於模塑作業流程時，有以下兩種方法：

第一種模型建立方法是以轉化表示動作或事件，以處所表示條件。以下說明時序、分頭並進、決策、協調、互斥等作業流程關係之 Petri 網路表示法：

a. 時序：兩個有時序關係之動作 A_1 和 A_2 (A_1 先於 A_2)，可以圖 5(a) 表示。

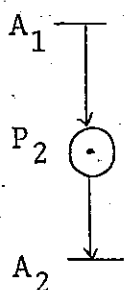


圖 5(a) 時序關係表示法

此一流程圖說明動作 A_1 執行完成後，便產生一標記於其輸出處所 P_2 內。在 P_2 內之標記隨即使動作 A_2 呈待動狀態。假使 A_2 不含任何“前提”條件，則一旦其呈待動狀態，即可起動，直至完成。

b. 分頭並進：在動作 A_1 執行完成後，兩個分頭進行之動作 A_2 和 A_3 ，可以圖 5(b) 表示。

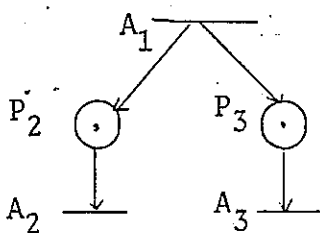


圖 5(b) 分頭並進關係表示法

上圖說明當 A_1 執行完後，即在處所 P_2 與 P_3 內各產生一標記。在 P_2 和 P_3 內之標記隨即使動作 A_2 和 A_3 呈待動狀態。假使 A_2 和 A_3 不含任何前提條件，則當 A_1 執行完後， A_2 和 A_3 即可同時起動。因 A_2 和 A_3 可附屬不同之前提條件，因而它們的起動一般而言是依分頭，非同步方式進行。

c. 決策：若動作 A_1 起動後，動作 A_2 和 A_3 僅能擇其中之一進行。此一關係可以圖 5(c) 表示。

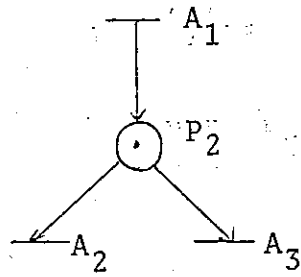


圖 5(c) 兩者擇一進行表示

上圖說明動作 A_1 執行完後產生在 P_2 內之標記使動作 A_2 和 A_3 呈待動狀態。然而僅其中之一可起動。這是因任一動作起動後，將移開在 P_2 內之標記，使得另一動作無法起動。（註：若 A_2 和 A_3 內含有前提條件時，則此兩條件必亦是互斥關係。）

d. 協調：若動作 A_3 須等動作 A_1 和 A_2 均完成後，方可開始執行。此一關係可以圖 5(d) 表示。

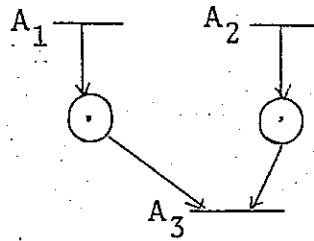


圖 5(d) 協調關係表示

協調工作祇在轉化上為之。

e. 互斥：當一處所之標記來自兩個或兩個以上的轉化時，我們定義此等轉化之起動不能同時發生。圖 5(e) 說明動作 A_1 和 A_2 間有互斥關係。

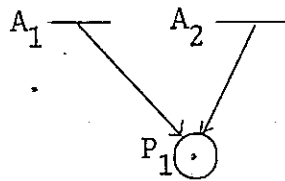


圖 5(e) 互斥關係表示

亦即在任一時刻，最多祇能有一動作可起動。

第二種模型建立方法是以處所表示動作與條件，以轉化表示動作之間的連繫關係。兩個有時序關係之動作 A_1 和 A_2 (A_1 在先， A_2 在後)，可以圖 6(a) 表示。

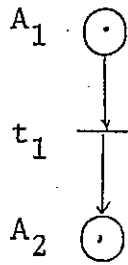


圖 6(a) 時序關係表示法

此一流程說明 A_1 一執行完成後，即將轉化 t_1 起動，把在處所 A_1 之標記移開，然後產生一標記於 t_1 之輸出處所內，此標記隨即將動作 A_2 起動執行。此種模型建立方法是將動作之起動執行與條件之成立混在一起，使得表示出來之流程較不易清楚。例如圖 6(b) 中，必須解釋成動作 A_1 和 A_2 均在執行中，同時

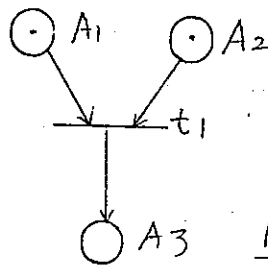


圖 6(b) 協調關係表示法

t_1 之起動須俟動作 A_1 和 A_2 皆完成後才發生。再則，此種模塑方法較不易描述如圖 5 (c) 之關係，因此，本文採用第一種方法。

3.1.2 Petri 網路之增修

爲了更能忠實地將作業流程表示出來，我們將 Petri 網路作了如下之增修：

a. 因爲動作執行需要時間，因此，我們定義轉化之起動需耗費時間，即轉化一起動，即移去所有輸入處所之標記，俟轉化之動作完成後，才產生新標記於其所有輸出處所。如此一來，Petri 網路之轉化與處所更能適切地反應出作業的流程關係。

b. 前提之附屬：如前節所述，動作之起動，除了需達到某種狀態外，有時尚須等到某一前提條件成立才發生。因此我們定義任一轉化上均可附有前提條件。當轉化呈待動狀態時，定義惟當其前提成立時，該轉化方可起動。另設一轉化之前提一動作表，以說明轉化在何種前提下起動，及其所

代表之動作的內容。(此表旨在描述各動作之內容)。前提之設立，旨在描述前提與動作間之“因果關係”。其中有二種前提最值得一提：

其一為時限監視 [7]，即在設定時限到達時(為一前提)，轉化方起動。

其二為資料比對。即在輸入資料與前提設定之資料值相同時，轉化方起動。

c. 在任一轉化上均可訂定起始處所，並立表說明該起始處所擁有標記之意義。其如何擁有，則視系統之實際製作而定。此舉旨在模塑辦公室間部份的傳輸協定 (communication protocol)

例如圖 7，表當 M_1 情報到來時，本辦公室內此流程之 P_{00} 擁有標記，若此時 P_{n1} 亦擁有標記，則 A_n 即呈待動狀態。

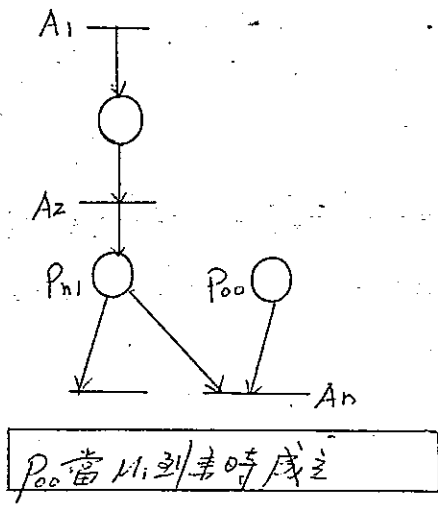


圖 7

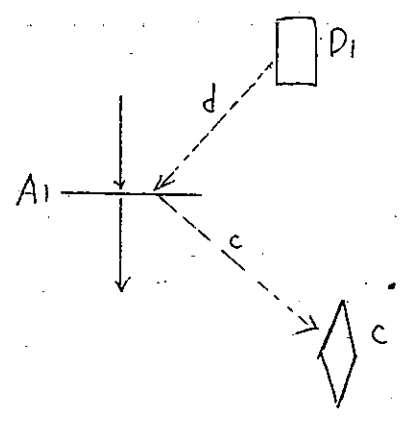


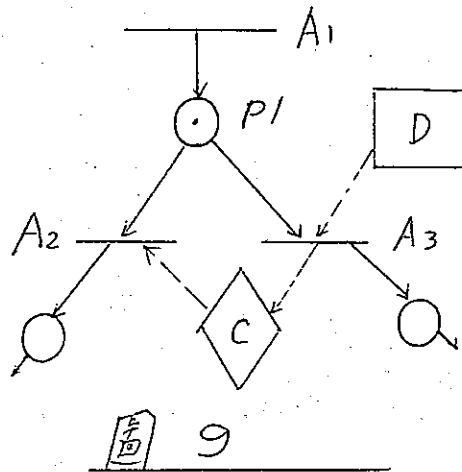
圖 8

d. 將資料流程圖 (data flow graph) 之資料運算子 (operator) 與轉化對映起來 [11]，圖 8 表此種對映。其中方塊“□”表資料庫 (global database)， D_1 為其中一資料檔 (file)， d 為 D_1 之一資料錄 (record)。菱形“◇”表工作區 (local area)， c 為其中一資料錄， A_1 為資料運算子。

我們定義所有的資料儲存區 (方塊或菱形) 均與 A_1 之呈待動狀態與否無關 (僅與其前提或動作之內容有關)，亦即方塊與菱形均不可能擁有標記。

轉化與資料運算子之對映，旨在利用資料流程解決控制流程之某些歧

異 (conflict) 現象 [1,4]。圖 9 中控制流程表示 A_2, A_3 當 P_1 成立時均呈



待動狀態，但 A_2, A_3 有互斥關係。設此時 A_2 起動，則 A_3 不可能起動。但 A_2 卻須用及來自動作 A_3 之輸出資料，顯然資料流程與控制流程間互為抵觸。因為就資料流程而言， A_2 與 A_3 呈時序關係，而非互斥關係。因此資料圖之加入提供吾人偵測歧異作業流程之能力。

另外整個 Petri 網路上之資料流程，亦可由此等資料圖表現出來，亦即將作業之輾進現象直接表現在 Petri 網路上。

e. 我們採用 Petri 網路的 "轉化階層結構" (transition hierarchical structure)，不擬採用其 "處所階層結構" (place hierarchical structure) 亦即從辦公室之觀點看，每一作業均是一個作業槓，經細分 (refine) 則可分成各職員所負責之各動作槓，如欲詳細瞭解各動作的內容，則可再細分為小動作槓的組合。其階層結構如圖 10。

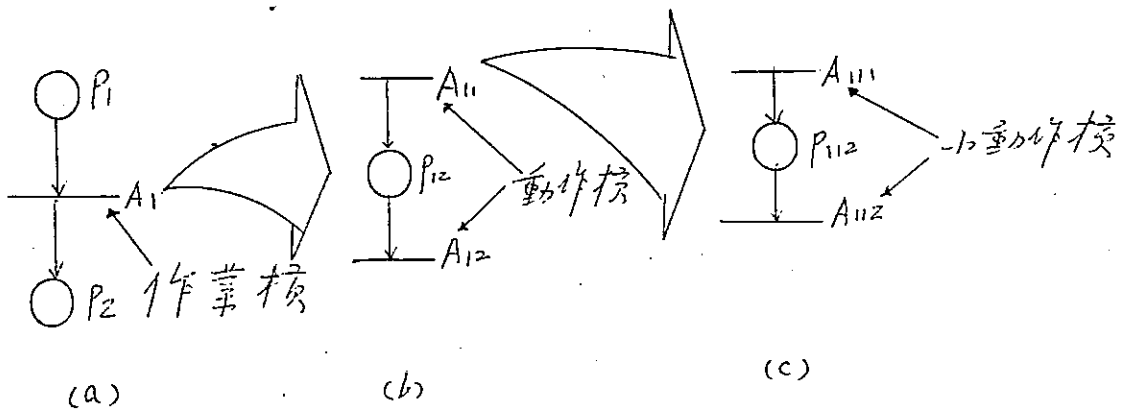


圖 10 轉化之階層結構

f. Petri 網路模塑作業流程之方法，主要是借用其分析能力。其模塑作業流程之難易性與可讀性，則暫不考慮，因 Petri 網路之結構本適應於計算機之執行，並非給人使用的。如欲遷就上兩項因素時，則可用 EFC(Extended Flow Chart) 來作模塑工具 [見附錄]。EFC 與 Petri 網路有一對一之對應，因此在分析上不會引起困難。應用時，則因與傳統程式流程圖相近，在模型建立與瞭解上，反較有利。

3.1.3 討論

在尋找模型的過程中，我們曾經想把前提視為動作來處理，以將前提無法表示在模型上之缺點去除，圖 11(a) 為轉化附有前提之情形，圖 11(b) 則將前提視為動作之情形。

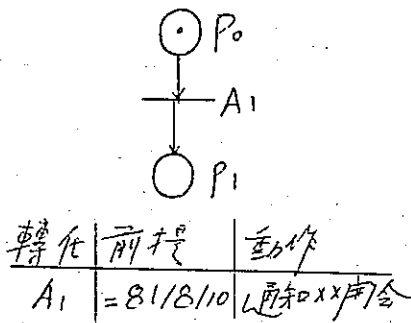


圖 11 (a)

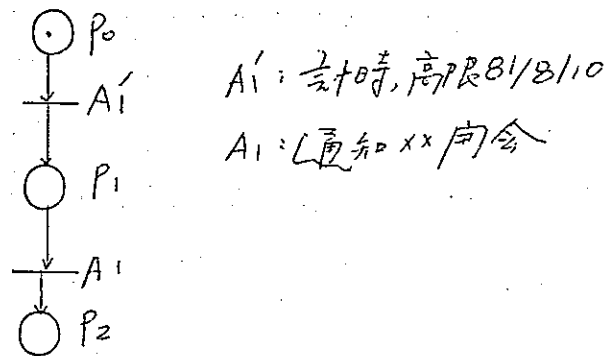


圖 11 (b)

如此，則祇在 Petri 網路上便能看出各轉化起動之條件（此時動作便無呈待動與起動狀態之區別）。但 Petri 網路之執行則必須如下定義：轉化在其輸入處所成立時起動，俟動作完成時，移開其輸入處所之標記，然後產生標記於其輸出處所上。因為惟有如此圖 11(b) 之 A_1' 才會在計時到 81/8/80 時，將標記由 P_0 移到 P_1 ，以備 A_1 之執行。

然而我們的想法，卻遇了下列的問題：

a. 轉化於動作結束後，方移開標記所引起之問題。圖 12 中，設 A_2' 計時達 2 小時時，移開標記，以備執行 A_3 。而 P_1 若在 2 小時內成立時，則可執行 A_3 之動作，而令 A_2' 無法呈待動狀態。今若當 A_3 動作正在執行，然尚未完成，亦即尚未移去 P_0 與 P_1 內之標記前， A_2' 之計時到達時（其移去 P_0 之標記），將迫使 A_3 之執行停止（因此時 P_0 內無標記）。如此則令作業之流程與所

欲順序相違。

克服此缺點之一法如圖 13。圖中我們另加了一個純轉化（不附動作） A_3' 使 P_1 一成立，即移去 P_0, P_1 之標記，令 A_2' 不呈待動狀態，然後才執行 A_3 之動作。如此則可避免圖 12 之困難。

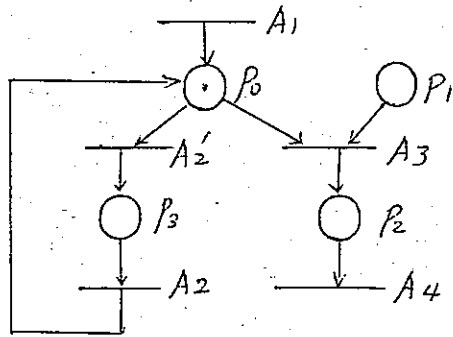


圖 12

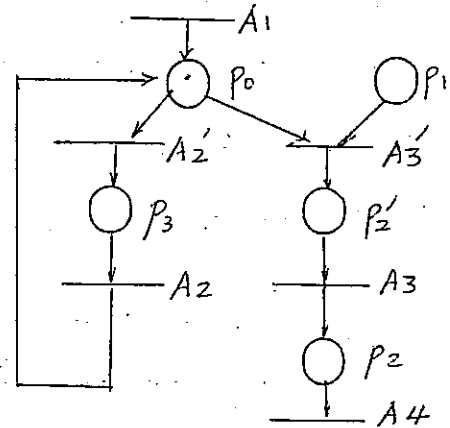
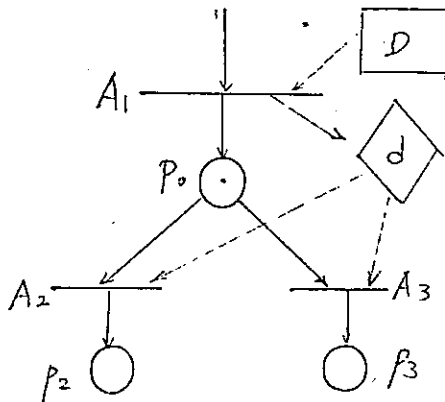


圖 13

b. 視前提為動作之問題

如前所述，將時限前提視為動作是可行的。然問題出在視資料比對為前提上。圖 14 中，設 A_1 將 d 放在暫存區， A_2, A_3 則取 d 與其前提比對，設 A_2 之前提為 $d=3$ ，而 A_3 之前提為 $d=2$ 。今若欲將 A_2 （或 A_3 ）之前提視為動作，則 A_2 之動作勢須多加一項：比較 d 與 3 之大小。但於此動作結束後，不管 d 等不等於 3， A_2 均移開 P_0 之標記，同時產生一標記於 P_2 內。很明顯地，此種流程非吾人所要。



轉化名	前提	動作
A_1	----	----
A_2	$d=3$	----
A_3	$d=2$	----

圖 14

綜上所述，我們發覺在原始Petri 網路上視前提為動作似不可行。在[1]所提 EPN內之“互斥或邏輯”似可解決資料比對似為前提之問題。但卻無比取代時限前提。因此本模型乃決定放棄視前提為動作之想法。

3.2 辦公室的模型建立

茲舉一典型之辦公室及其內某典型之動作為例，試用上節之模型加以模塑並藉以分析該辦公室某些固定之基本工作。舉例如下：

設辦公室內科長指定二職員籌辦內部職員××課程訓練班。職員V首先預估費用並簽呈上級單位，請核准費用。為求時效，擬定每逾二週未有回音時，應再簽呈催告。上級單位一般於同意費用支出時，要求將學員及講師名冊呈送備查。因此，上級核准文書到達時，職員V即向出納股請款，職員U則同時在本單位徵求學員及聘請講師。然後簽呈附講師及學員名冊呈報上級單位。並函告各學員上課細節（如時間、地點等）。

我們可用圖15及表1和2來模塑此一辦公室作業型態。圖15為其作業流程，表1為各轉化之前提一動作內容，表2為各起始條件說明。（圖15及表1和2見下兩頁）。

檢視表1，我們發覺 t_7 ， t_8 ， t_{13} 及 t_{20} 之動作較有趣。 t_7 為得知上級不核准時，即時轉入結束流程之動作，本身實際上不執行工作，因此其轉化瞬間起動（表1之動作欄內，未有動作名（如 A_1 等））。 t_8 與 t_{20} 為一配對，可看成多人多機工作之開始與結束，本身動作之分派與收尾的動作，亦皆是瞬間起動轉化。 t_{13} 為分派單人多機動作之開始，亦不耗費時間。此等轉化之動作欄內皆未附動作名，表示其不費時間之特性。

表1之前提欄內，只有 t_5 ， t_7 及 t_8 標出前提，此等轉化之前提，旨在提供決策（即選擇那一轉化起動）之基礎。

表1之動作欄內，甚多動作其性質相近或完全相同。例如 A_3 A_{10} A_{15} 均是“送處長簽章”， A_4 A_{11} A_{16} 均為“發遞文書”，似各可歸為一類。 A_8 A_9 A_6 均可視為“資料取用”一類， A_2 A_7 A_{12} 可視為“產生文書”一類。 A_{13} 及 A_{14} 為“產生函並發函”一類。（實際應用時， A_{13} 類可看成 A_2 及 A_4 兩類之結合，而不須重新歸併一類）。 A_1 可看成“需要計算之工作”的代表。

由上面的觀察，我們發覺辦公室的動作大致不離開下述範圍：

- a. 資料庫之資料處理，如 A_8 類。例：資料之存取、增刪等。

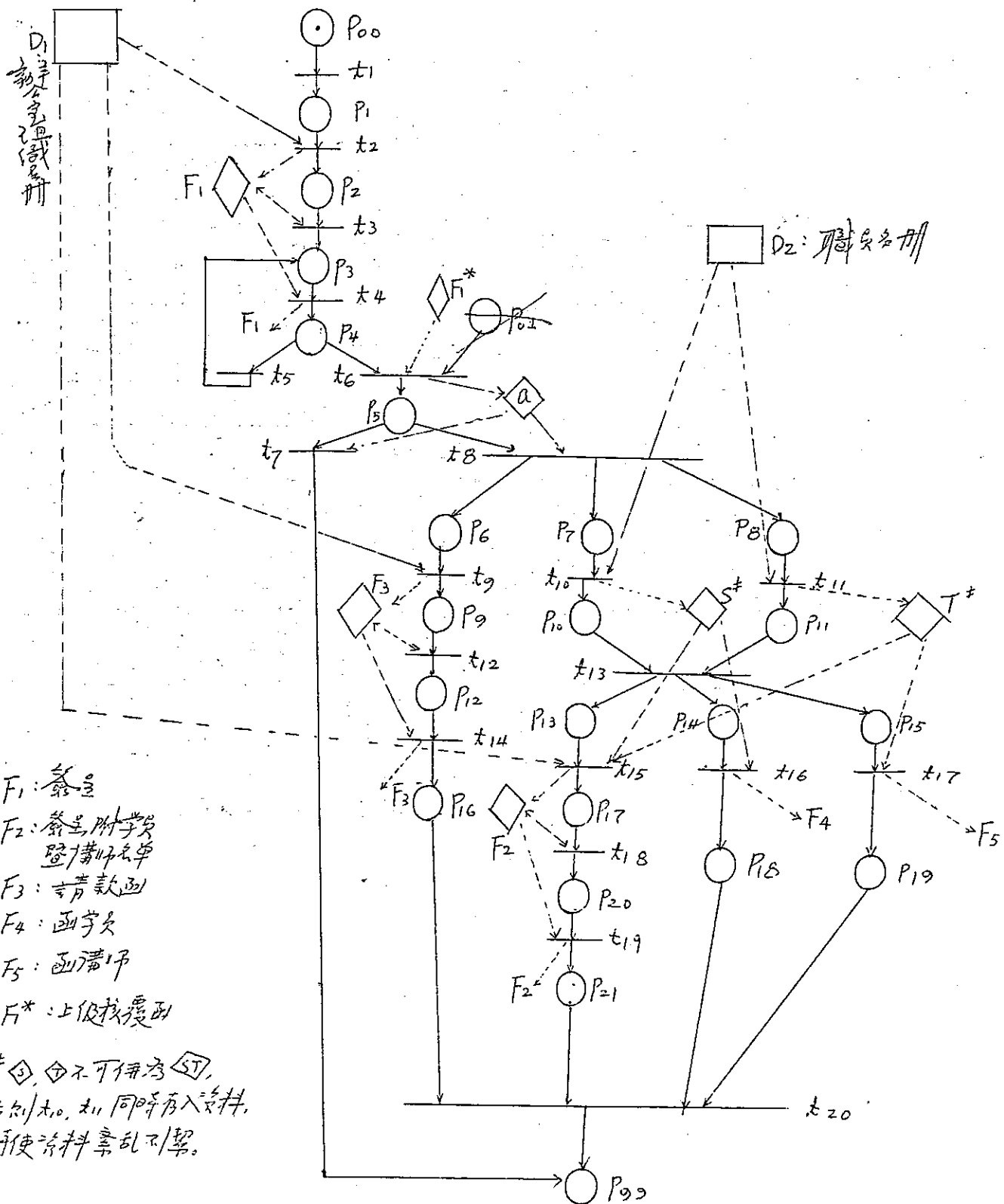


圖15 作業流程

表 1 轉化之前提工作表

轉化名	前 提	動 作
t_1		預估費用 (A_1)
t_2		產生簽呈，擬辦訓練班 (A_2)
t_3		處長簽章 (A_3)
t_4		送簽呈 (A_4)
t_5	? \cong 2 週	發出警報 (A_5)
t_6	? F_1^*	取得上級指示內容 (A_6)
t_7	? 不核准	
t_8	? 核准	
t_9		產生請款函 (A_7)
t_{10}		徵求學員 (A_8)
t_{11}		聘請講師 (A_9)
t_{12}		處長簽章 (A_{10})
t_{13}		分派
t_{14}		發函 (A_{11})
t_{15}		產生簽呈，並附學員及講師名單 (A_{12})
t_{16}		發函學員 (A_{13})
t_{17}		發函講師 (A_{14})
t_{18}		處長簽章 (A_{15})
t_{19}		發函 (A_{16})
t_{20}		糾合

表 2 起始條件成立表

條 件 名	意 義
P_{00}	起始情報到來時，條件成立

- b. 暫存區之資料處理，如 A_1 類。例：C ← A 或 C ← A + B 等敘述。
- c. 檔案或表格處理，如 A_2, A_4 類。例：產生函、發函或在函上填具意見等。
- d. 簽章，如 A_3 類。例：文書簽字等。
- e. 純與流程諸現象有關之動作，如 t_{13} 類。例：作業分頭指派，與作業終結集合等。

因此適當運用此些基本動作項目，將大有裨益於辦公室之自動化過程。

4. 模型分析

本模型基本上是 Petri 網路，除了資料圖須另作分析外，本節之分析將採用 Petri 網路之部份分析理論。4.1 節將介紹有關 Petri 網路之分析理論。4.2 節則針對本模型作分析。4.3 節將本模型與既存諸模型作一簡單比較。

4.1 Petri 網路分析理論：

用 Petri 網路模塑控制流程，旨在藉 Petri 網路理論分析控制流程是否存有「卡死」(deadlock)之現象，模塑之控制流程中，處所表狀況，因此只有狀況成立與不成立兩種情況。若視標記存於一處所與否表狀況之成立與否，則每一處所至多僅能擁有一標記，此種限制處所只能擁有一標記之 Petri 網路為「安全 Petri 網路」，亦即沒有與標記數目有關之界限問題 (boundness problem)。流程的卡死乃指在某一情況下，流程內部無法繼續進行下去的現象。此問題亦稱「活性問題」。一個活的控制流程，表示在任何情況下，該流程內總有轉化成待動狀態。

Petri 網路的活性問題定義如下：若 Petri 網路內任何一個轉化之起動，均無法令某一轉化呈待動狀態時，該轉化稱卡死轉化。一轉化至少可經由一起動順序而呈待動狀態時，稱為活性轉化。一個活性 Petri 網路係指其轉化均為活性轉化者而言。

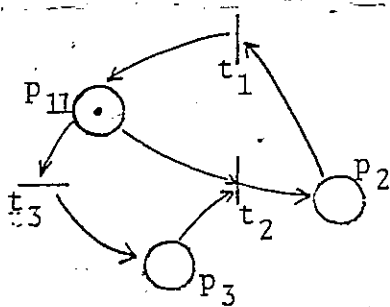


圖 11

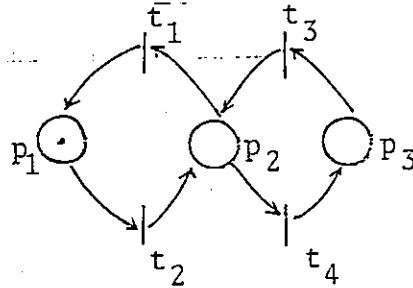


圖 12

圖 11 為非活性 Petri 網路，其 t_2 非活性。因 t_3 起動後，卡死。

圖 12 為一活性 Petri 網路，其所有轉化均屬活性轉化。問題在於如何確認圖 12 為活性 Petri 網路？Hack 曾證明此種活性問題即是 Petri 網路之可及性問題 (Reachability) [10]。可及性問題之定義如下：已知 Petri 網路之起始狀態 M_0 ，試問已給之狀態 M ，可否由 M_0 導出，亦即從 M_0 經一連串起動順序推及。舉圖 12 為例，圖 12 之起

始狀態為 $M_0 = (1, 0, 0)$ 此時若 t_2 起動則狀態為 $(0, 1, 0)$ 此狀態稱由 M_0 直接可及，若 t_4 此時起動，則狀態轉為 $(0, 0, 1)$ 此狀態稱為由 M_0 可及。

一般而言，Petri 網路之分析理論，有下列三種：線性控制理論法，"不變質"法以及"可及樹"法[13]。在分析可及性問題方面，似以可及樹較為易懂、便利。以下我們將著重在探討 Petri 網路之可及樹分析法。

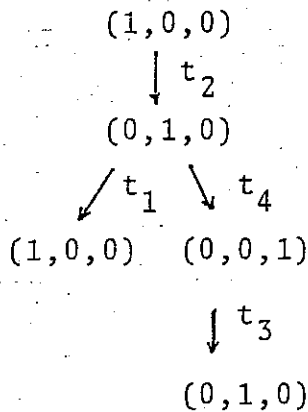


圖 13

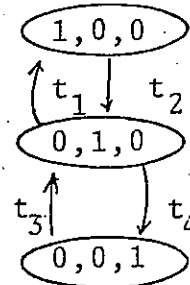


圖 14

圖 13 為圖 12 之可及樹，起首之樹根為起始狀態 $(1, 0, 0)$ ，其下之實箭頭表 t_2 起動後，狀態轉成 $(0, 1, 0)$ 。此狀態下有兩轉化呈待動狀態，但相互抵觸，因此若 t_1 起動，則走左子樹， t_4 起動則走右子樹，以下類推。

我們發現，可及樹事實上是模擬 Petri 網路之動態狀況之空間。而 Petri 網路之動態空間可能是無限的，因此可及樹亦可能是無限的。圖 13 中，葉端 $(1, 0, 0)$ 與起首狀態相同，可再起動演變下去。而葉端 $(0, 1, 0)$ 則與第二狀態相同，亦可繼續伸展下去。圖 13 將其列為葉端以避免繼續伸展在圖示上之困難。實際分析時，可採用標記機 (token machine) [7] 之方式。圖 14 為圖 12 之標記機。比較 13, 14 兩圖，可知圖 14 為一圖形，而非樹。

如果利用有限的圖形，能夠描述出整個 Petri 網路之動態狀況，則其是否活性，便可用觀察法得知。如圖 13 或圖 14 中，各可能之起動順序包含了所有的轉化，因此該 Petri 網路為一活性網路。

4.2 本模型之分析

本模型之分析，其目的在確認任一流程為一"正常流程"。茲定義

“正常流程”為一滿足下二條件之流程：

(i) 流程為活性（見 4.2.1 節）；

(ii) 控制流程與資料流程不發生歧異（見 4.2.1 節）。

4.2.1 控制流程之活性分析

茲定義控制流程之活性如下：

(i) 所有轉化均為“潛在性可起動”，一轉化符合下兩條件時稱“潛在性可起動”；

a) 流程存在一序列之起動，能使該轉化呈待動狀態；

b) 轉化之所有輸出處所均不含標記；

(ii) 流程之標記機內，包含所有轉化；

(iii) 流程之標記機內之各分支必為

a) 可自行由 $(1, 0, \dots, 0, 0)$ 之起始狀態抵達 $(0, 0, \dots, 0, 1)$ 之終結狀態，且其終結狀態必為 $(0, 0, \dots, 0, 1)$ 者，或

b) 成迴圈，但該迴圈非為潛在性無窮迴圈者。

茲舉一例說明之。圖 15a 之流程，其標記機為圖 15b。

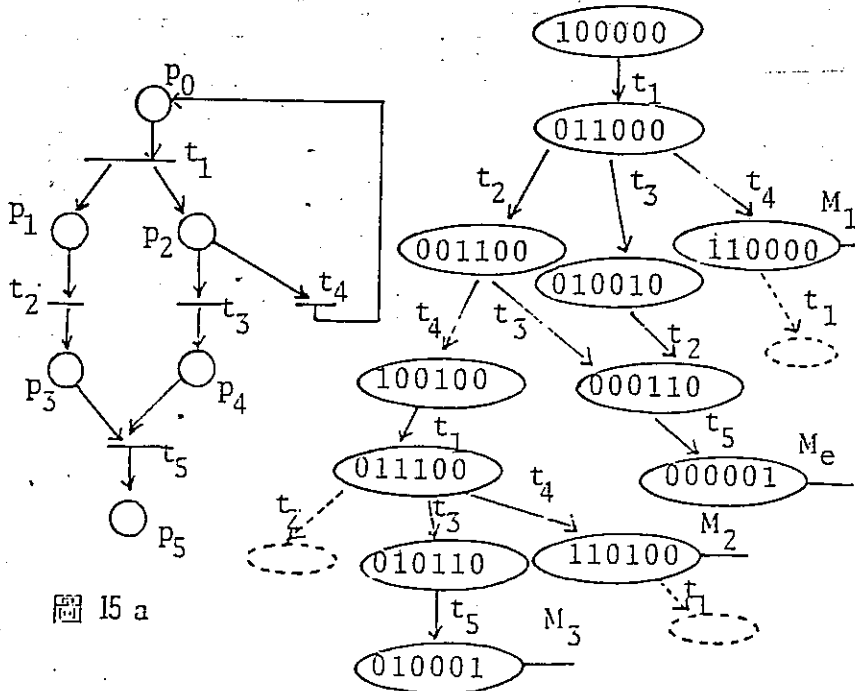


圖 15 a

圖 15 b

檢視圖 15b 發覺其符合 (i) 之條件，其中虛線部份均表受 (i)b 條件限制而無法存在之狀態；此流程亦符合 (ii) 之條件，因各分支包含了所有轉化。但本流程不符合 (iii) 之條件，因為終結狀態應為 $M_e = (0, 0, 0, 0, 0, 1)$ 而各分支之葉端出現有三個狀態 M_1, M_2, M_3 非為正常終結狀態者。此流程因此，應為非活性，亦屬非正常流程。

茲再說明 (iii) 之迴圈現象如下。圖 16b 為圖 16a 之標記機。其中， t_2 之起動構成迴圈，如 t_3 在有限時間內不起動，而 t_2 之事件內容，又沒有要求外界介入或自身有能力終止流程時， t_2 之迴圈將往復不止。似此之迴圈稱為“潛在性無窮迴圈”，一個潛在性無窮迴圈之流程，絕非正常流程。

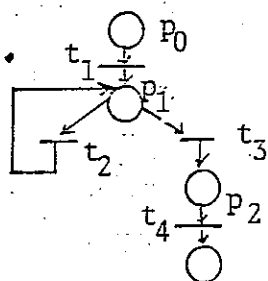


圖 16 a

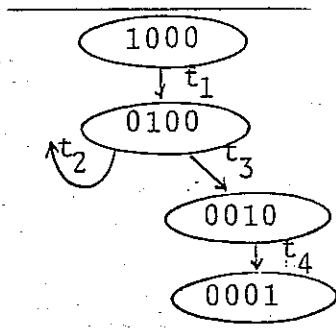


圖 16 b

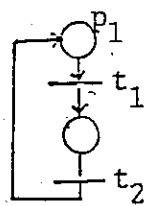


圖 17 a

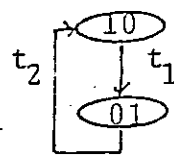


圖 17 b

迴圈是構成潛在性無窮迴圈的必要條件，如圖 17a 之 Petri 網路其圖 17b 之標記機上出現 $(1,0) \xrightarrow{t_1} (0,1) \xrightarrow{t_2} (1,0)$ 之迴圈，因此有可能成為無窮迴圈。但無窮迴圈之發生與否卻與迴圈內轉化所代表的事件內容有關，因此，在分析流程模型時，若標記機上出現迴圈時，就必須檢視該等迴圈所經之各轉化所代表的事件內容，如所有轉化均無要求外界之介入以強迫終止流程，而本身亦無法自行終止流程時，此迴圈即是個“潛在性無窮迴圈”。

非正常流程約有二種可能來源：其一為實際流程本身之不適當，其二為流程模塑時之錯誤。例如：辦公室人員於獲悉流程有“潛在性無窮迴圈”存在時即應與實際流程比較。如係前者，但礙於某些因素而暫時無法修正實際流程時，則該流程執行時，系統須在每次無窮迴圈發生時，設法終止無窮迴圈。例如系統可計數經歷該迴圈之次數，於該數目超越某一限值時終止流程或知會辦公室人員處理。當然，直接修正實際流程以消除無窮迴圈最佳。如係後者，則辦公室人員即應修正模塑流程以消除模塑錯誤所致之無窮迴圈。

4.2.2 資料流程與控制流程之比較分析

首先，定義一 Petri 網路之資料流程為其所有 "資料序組" 之集合。資料序組之定義如下：一資料序組為一 Petri 網路流程之某一輸入資料，經該流程之一組資料運算子運作後，底於消失之過程。資料消失有下列二種情況：一為資料離開本系統，另為資料運算子所吸收。資料運算子吸收資料之情況有二：一為資料運算子取為資料運算元，但資料運算子不再產生新資料；另為資料運算子將資料存入資料庫，且該資料庫不再為所有運算子所使用。圖 18a Petri 網路之資料流程為圖 18b 及 c 之二資料序組之集合。

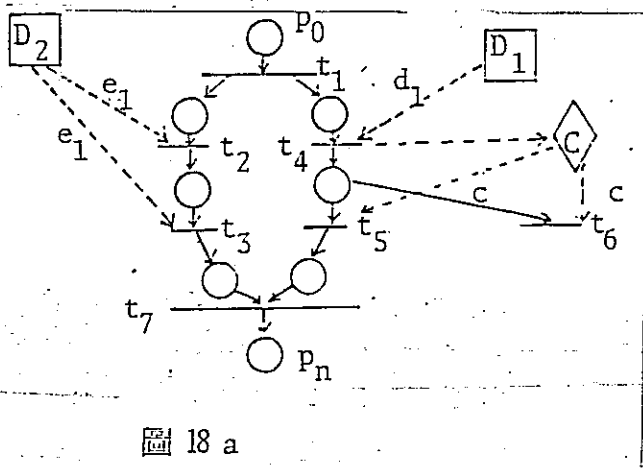


圖 18 a

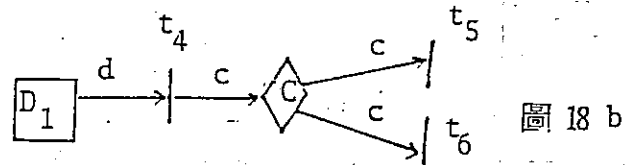


圖 18 b

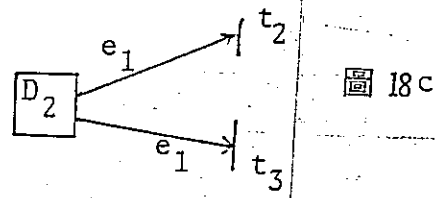


圖 18 c

資料流程分析之目的有二：一為在資料流程上進行 "資料存取效率化" 的工作，另為與控制流程作比較，作解決歧異現象的根據。以下依次說明達此二目的的方法。

資料存取效率化的意義為，資料庫內同一資料檔若為兩個以上之運算子所使用，且此兩運算子具有時序關係時，可在兩運算子間加入 "暫存區"，使資料庫之進出次數減少，以爭取時效。下舉兩例說明之。圖 18c 中 D_2 在資料庫內， t_2 和 t_3 均取用 D_2 之資料錄 e_1 ，而控制流程上顯示 t_2 時序在 t_3 之前，因此圖 18c 可改為圖 18d 之形式，如此，即可減少 D_2 之進出次數。

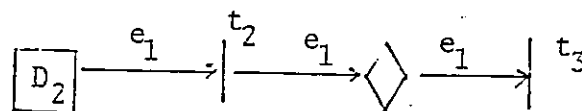


圖 18 d

另例，若圖 18c 更換為圖 18e 之形式，亦即 t_2 改為存入資料錄，而 t_3 取用並存回資料錄時，則可將 t_2 與 t_3 間之 D_2 改為一暫存區，如圖 18f 所示。如此，亦可降低 D_2 之進出次數。

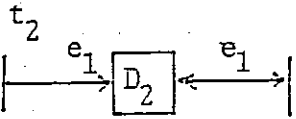


圖 18 e

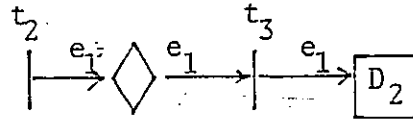


圖 18 f

此等利用暫存區來連貫兩運算子之方法，對於資料存取頻仍之流程，將可省去可觀之資料出入時間，達到資料存取效率化的目的。

3.1.2 節曾提及歧異現象，以下試作深一層之探討。此處只考慮一流程內，因“工作區”的使用不當所導致的控制流程與資料流程不一致的現象。至於資料庫之存取，因牽涉到流程間之複雜執行關係，除非作動態模擬，殊難找出資料與控制流程抵觸之實情。以下試分三種情況說明一些歧異現象。

(i) 當 t_i 與 t_j 為分頭並進時，不可有圖 19a, b, c, d 或 e 之資料序組，否則稱資料流程與控制流程間存有歧異。

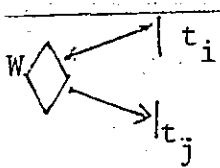


圖 19 a

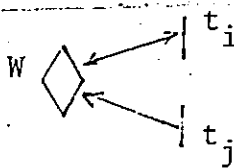


圖 19 b

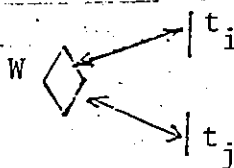


圖 19 c

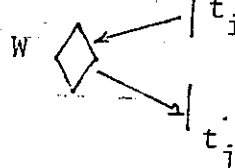


圖 19 d

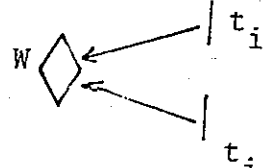
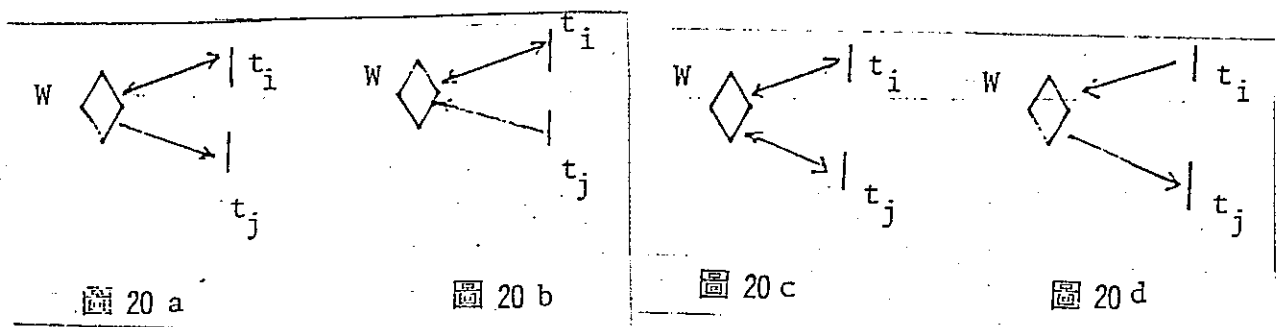


圖 19 e

此種歧異的來源為 $t_i t_j$ 既是分頭並進動作，應是互為獨立才是。但圖 19b 中 t_i 用及 t_j 之資料，則 t_i 與 t_j 之執行次序一有不同，便會有不同之執行結果，此與其獨立進行之含意不合。圖 19e 中， $t_i t_j$ 同時存入資料。對兩個獨立進行的動作，其工作區本應獨立才是，否則存到 W 內之資料就不知是 t_i 抑或是 t_j 之執行結果了。如果如此之資料序組竟不影響全部流程之執行結果時，則 t_i 與 t_j 之分成兩動作，似屬非必要之舉了。(ii) 當 t_i 與 t_j 為決策互斥時，不可有圖 20a, b, c 或 d 之資料序組，否則資料流程與控制流程間亦存有歧異。



此種歧異導緣於互斥兩動作之資料引用，竟互有關係所致。

(iii) 當 t_i 與 t_j 有先後之次序關係，同時不再存有不同時序之轉化使用此工作區，並且 t_i 與 t_j 不同處於同一迴圈時，存在類似圖 21 之各資料序組，表示該工作區之設置不當。(($t_i < t_j$) 表 t_i 時序在 t_j 之前)。

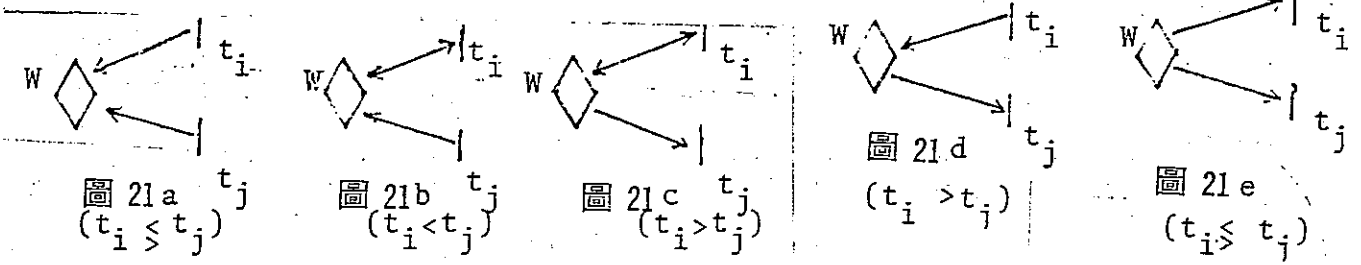


圖 21a, e 表示在 $t_i < t_j$ 或 $t_i > t_j$ 時，工作區 W 之設置不當。
 圖 21b 表示 $t_i < t_j$ 時， t_j 用不上 t_i 之資料而 t_j 存入 W 之資料又無其他轉化使用上，因此 t_j 存入之動作設計不當。
 圖 21c, d 表示 $t_i > t_j$ 時， t_j 取用之設計不當。

圖 18b 中 $t_4 t_5$ 及 t_6 均牽連到同一工作區。其中 $\{t_4 t_5\}$ ， $\{t_4 t_6\}$ 兩序組均具時序關係，但均不在 (iii) 之情況內，而 $\{t_5 t_6\}$ 序組為決策互斥，其資料序組亦不違背 (ii) 之諸項，因此圖 18a 應無歧異存在。

解決歧異的方法，規實際控制流程與資料流程兩者一致的程度而定。於參照實際作業流程後，決定控制流程與資料流程何者較為合理，而作修正流程之根據即可。

4.3 模型比較

現有的辦公室作業流程模型中，以 Zisman 的 Augmented Petri 網路最早 [11]，其貢獻為提出 Petri 網路加入 Production Rules 來模塑一作業流程。但因其著重在「一職員有一 Petri 網路」的想法，因此，作業之整體流程，反而無法由 Petri 網路上得知，必須以 Production rules 之 action 部分作為整體流程之貫連，在流程分析上較不易。其 production rules 之

condition部分亦有值得商榷之處，例如將資料之可用與否列成 condition 即其一（見 3.1.2），其資料流程隱在 Production rules 內難以用之以解決歧異；並且未提及辦公室間之聯繫方法等均是缺點。張系國先生提出的 Alerter System model [3] 與 Zisman 者非常相似。其重點為利用同一資料庫作為各動作的聯繫場所，其想法克服了 Zisman 的部分困難，但因其模型的設計層次較近實作，因此，模型之複雜程度與實作後之效能(performance) 均是不可忽視的問題。Ellis 提出的 ICN (Information Control Net) [5] 事實上是把流程圖作了部分修改，並加入了資料流程，對辦公室的作業輾進提供了表現的方法，並且藉以達成流程整型 (procedure restructuring) 的效果原屬良策，惟該模型亦似只針對同一辦公室內之作業流程，而於辦公室間作業流程之聯繫與時限前提等之描述則付之闕如。Tschritizis 的 Form flow model 在工作量重配 (workload redistribution) 與流程整型上有其優點[3] 惟該模型較有利於經模塑後之作業流程的分析，如欲以之模塑一辦公室，則似不易。本模型草創之時，曾設法揉合各模型之長，並且以詳細分析辦公室之作業流程為基礎。因此在作業整體流程上，辦公室間聯繫上，資料流程上，模型層次上與模塑難易上均作了適當的取捨，應為一較完整之模型。以之模塑一辦公室當更簡易，明瞭與準確有效。

5 結語

本文已對辦公室之特性作了詳細分析，並根據其特性提出一套模型，供設計師描述辦公室的各種作業流程關係。所提之模型主要為 Petri 網路，另附適當的增刪。此一模型並將資料圖明顯地表示出來，使得在模塑辦公室時，更能忠實地反應出各作業流程與資料庫間之關係。

本文所提之模型分析方法，僅是現階段之研究結果，距發展出一套完整的分析方法，尚有一段距離。更深一層之分析方法，則有待日後的探討。

6. 參考文獻：

1. Baer, J. L., "Model, Design and Evaluation of A Compiler for A Parallel Processing Environment," IEEE Trans. on Software Engineering, SE-3, No.6, Nov. 1977.
2. Baumann, L. S., and Coop, R. D., "Automated Workflow Control : A Key to Office Productivity," Proc. AFIPS Office Automation Conference, March 1980.
3. Chang, S. K., "Knowledge-Based System," Chapter 14 (to be published).
4. Ellis, C. A., "Office Information Systems and Computer Science." ACM Computing Surveys, 12, 1, 3, 1980.
5. Ellis, C. A., "Information Control Nets : A Mathematical Model of Office Information Flow," Conference on Simulation, Measurement and Modeling of Computer System, 1979.
6. Hammer, M., Howe, W. G., Krushal, V. I., and Wlandawsky, I., "A Very-High Level Programming Language for Data Processing Applications," CACM 20, 11, 4, 1977.
7. Merlin, P. M., "A Methodology for the Design and Implementation of Communication Protocols," IEEE Trans. on Communication, COM-24, June 1976.
8. Naffah, N. "Integrated Office Systems Protocols," Intégrated Office Systems, North-Holland, IFIP 1980.
9. Neuman, W. M., "Office Models and Office System Design," Integrated Office Systems, North-Holland, IFIP 1980.
10. Peterson, J. L., "Petri Nets," ACM Computing Surveys 9, 3, 3, 1977.
11. Zisman, M. D., "Representation, Specification and Automation of Office Procedures," Ph.D. Dissertation, Wharton School, University of Pennsylvania, 1977.
12. Zloof, M. M., "QBE/OBE : A Language for Office and Business Automation," IEEE Computer, May. 1981.
13. 談肖虎： "派翠網理論在資訊流動系統上之應用 " 交大計研碩士論文， June 1981.

附錄：EFC 模型

爲了提昇本文所述之模型（以下簡稱 PND 模型）的易瞭解度及易模塑性，我們本擬用一般流程圖作爲 PND 模型的“副模型”（亦即與“用者”較接近之模型）。然而一般流程圖用於模塑辦公室作業流程之錯綜複雜的關係時，顯得力不從心，例如分頭並進，協調等關係便無法描述。因此本附錄擬提出一 EFC 模型，亦即擴充一般流程圖之描述能力，同時保持一般流程圖易懂，易描述的特性，作爲 PND 模型之副模型。因其與一般流程圖之差異甚微，“用者”在模塑其深悉之流程時，便不致於因尚不熟悉 PND 模型而分散專注意力，以致於模塑出來之流程與實際流程大相逕庭，或錯誤百出。因 EFC 模型與 PND 模型之間有一對一之對應，因此其轉換相當便利。A-1 將先介紹 EFC 模型，A-2 則提出 EFC 與 PND 間對映轉換的演算法。

A-1 EFC 模型介紹

效用 EFC 模型模塑一電信局取用土地籌建交換機房之作業流程如圖 A-1。

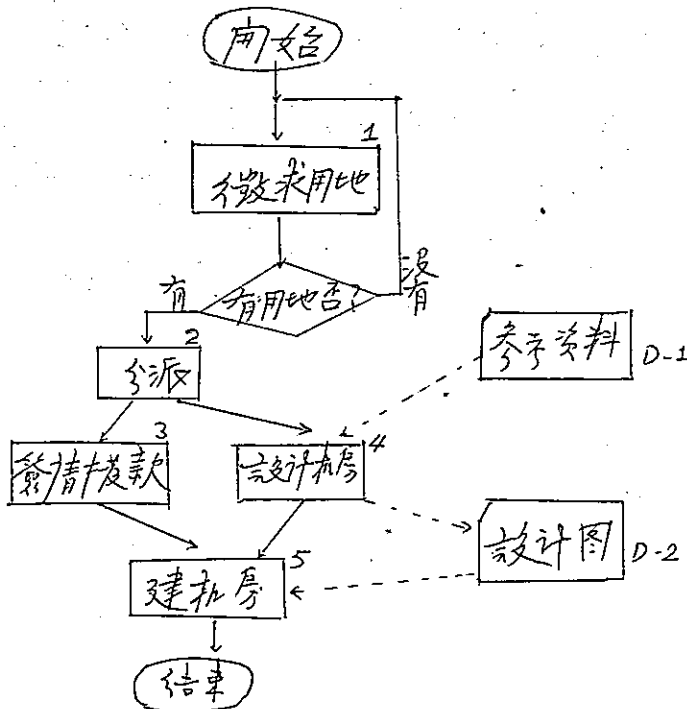


圖 A-1

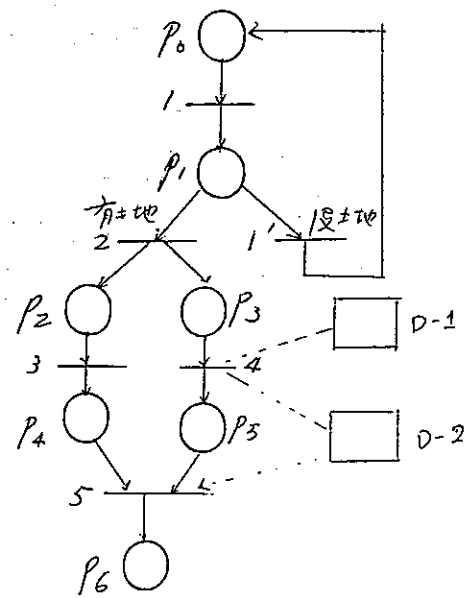
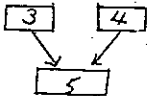
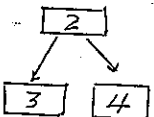
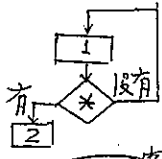
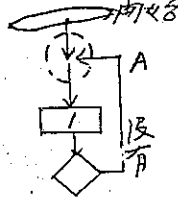

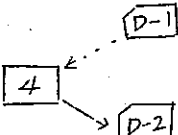




圖 A-2

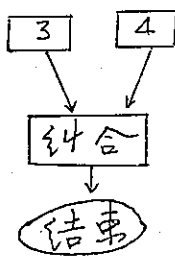
此流程之意義如下：首先，先“徵求機房用地”，俟確“有”用地時，一邊“簽請上級撥款”，一邊則着手“設計機房”（兩邊分頭並進），俟兩者均完成後，再開始“建機房”。以下我們分析此EFC 之各主要元件，為簡略起見，以方塊號碼代表方塊內容：

- a.  : 表協調 3,4 後，再作 5 (AND IN)
- b.  : 表 2 作完後，3,4 分頭並進 (AND OUT)
- c.  : 表決策，"有*"時作 2，"沒有*"時作 1 (OR OUT)
- d.  :  內表 1 之方塊可由“開始”進入，亦可由“A”箭頭進入 (OR IN)
- e.  : 表讀入 D-1 資料檔，並寫入 D-2 資料檔。

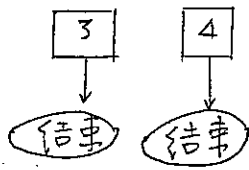
上述分析，顯示EFC 模型與一般流程圖不同處在於其正式定義了分頭並進，協調、決策以及 OR IN 等元件。今為了使協調與並進之模塑更廣泛有效，再定義下二個專為協调用或表分頭並進用之元件：

- f.  : 此方塊表其下之方塊必二個以上，可分頭並進。如圖 A-1 之方塊 2。其本身不作任何實質動作，僅用於作分頭並進動作之領頭而已。
- g.  : 此方塊協調其上二個以上之方塊，但本身不作任何實質動作，亦即為純協調方塊。其用法如下：若圖 A-1 改為須在方塊 3,4 作完後結束流程時，因 3,4 代表分頭並進動作，流程之正常結束點應是 3,4 全部作完。此時便可加入本元件以正確表示並進

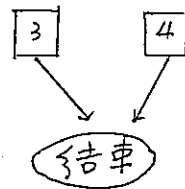
動作之完成才結束流程的特性。如圖A-3 所示。



圖A-3



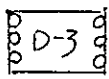
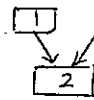
(a)



(b)

圖A-4

圖A-4 均為錯誤表示法。因其意義為3作完"或"4作完後均結束流程也。

此外"暫存"資料檔可用  表示。而  中若 F_1 來自其他辦公室，則可表傳輸情形。

如此則EFC之模塑能力將與PND相等。

A-2：轉換演算法。

EFC 模塑出來之流程可經本演算法而轉換成PND 模型。

1 資料圖轉換：



EFC \longrightarrow PND

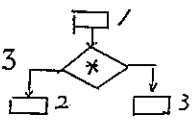
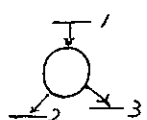
1.1   (資料庫)



1.2   (暫存區)



2 流程圖轉換：


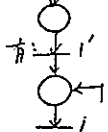
2.1   (處所)

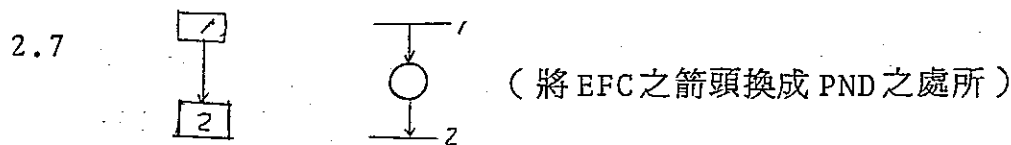
2.2   (決策)

2.3   (轉化)

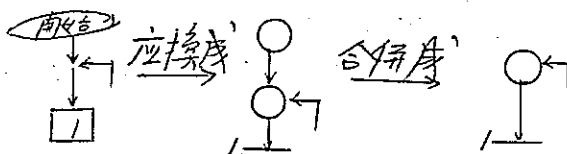
2.4   ("有(*)" 為此轉化之前提)

2.5   (將"有"改爲一轉化，"有(*)"並爲此轉化之前提，此轉化爲純轉化，不作動作)。

2.6a   (將"有"改爲一轉化，"有(*)"並爲此轉化之前提，此轉化爲純轉化，不作動作)。

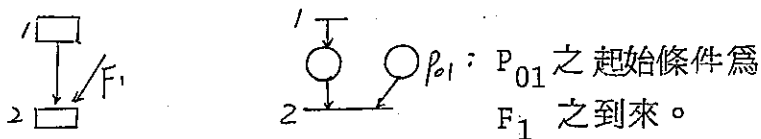


2.8 若轉換結果，在兩轉換之間出現兩處所時，該兩處所須合併為一處所。例如



3. 傳輸轉換：

(含起始條件表之形成)



4. 轉化表之形成：將EFC各方塊之動作，作為相對應之轉化的動作。但如有2.4之情形者，轉化上之前提須列入轉化表。有2.6之情形者須另加一轉化，記入其前提，但不記入任何動作。

應用此演算法，可將任何EFC模型轉換成PND模型。圖A-1經轉換後之流程部分如圖A-2。(轉化表須另列)。