

利用延伸 Petri nets 建構校園文書工作流程模型之初探

簡培修

國立臺灣師範大學 資訊中心

phchien@ntnu.edu.tw

摘要

在校園文書處理流程 e 化的過程中，始終欠缺一套很友善直覺的建模及模擬分析工具，往往到了要正式上線時，才發現原先設計不夠完備，或是發生了一些事先未設想到的狀況而產生錯誤。兼具數理基礎及圖形化視覺效果的 Petri nets 很適宜用來作為工作流程建模的工具。本文參考典型 Petri nets 之學理基礎及各種延伸擴充功能的應用，再依照工作流程的特性，先將各種在工作流程上常見的基本型改用 Petri nets 來建模，並闡述這樣做法的優點。然後以數個實際應用於校園的文書處理流程為例，嘗試以典型 PN，以及高階的 PN，包含 Hierarchical PN、Timed PN、Colored PN 與含有 Inhibitor arcs 的 PN，提出一些可能的建模方式，這些對於校園在文書處理 e 化過程與內部工作流程再造，相信能有所助益。

關鍵詞：工作流程、Petri nets

Abstract

We proposed an effective approach, which is based on the theory of Petri nets and the extensions, including hierarchical Petri nets, timed Petri nets and colored Petri nets, to model and analyze the workflow of the official documents in campus. We also illustrated some real cases in campus which benefited from using Petri net extensions as modeling tools.

Keywords: Workflow, Petri nets

1. 前言

工作流程是機關單位處理公務和行政管理的重要程序。在各級政府機關單位中，文書處理又是日常工作中最普及的。利用網路資訊技術對業務流程及公文進行高效有序的電子化處理，是單位資訊化建設的重要組成部分。行政院於民國 88 年成立公文跨部會工作小組，自民國 89 年起推動政府機關間公文電子交換作業，目前政府機關約有四成多的公文均採用電子公文交換服務，每日電子公文交換服務容量已達十萬件，服務成效良好，減少許多人力與時間，提升政府施政效能及決策品質[2]。各機關也隨即展開內部文書處理 e 化，包

含文書流程電子化作業與線上簽核。然而在進行工作流程 e 化之初，首先面對的難題即是如何很明確、有系統地將各種業務流程清楚地訂定。

因此，即便各種文書處理流程電子化有關的資訊應用系統，在資訊軟體系統業界可說是百家爭鳴，但是，卻少有一次導入即成功之案例。以我們服務的學校為例，在導入請假單線上簽核時，即使在系統分析之初，即由資訊人員參與，不斷與人事單位及委外廠商開會溝通確認流程，等到完成後正式上線時，仍然需要近半年的系統調適期，方能運作順暢。因此我們認為，為因應機關內部各種多變性與多樣性的作業流程，必須要有一套建模工具來輔助流程的設計。這套的工具必須能分析預先定義好的流程，協助指出錯誤或是不合理之處。

在文獻探討過程中，我們發現已經有多年發展歷史，被廣泛應用在導覽學習、資訊傳遞、製造業生產流程、企業流程、生物資訊等各領域的 Petri nets，足以勝任這樣的工作。Petri nets 學理十分穩固紮實，視覺化的圖形特性，也使人容易理解流程的實際運作，加上已經被研發出來的各種延伸功能，強烈吸引我們利用 Petri nets 的特性，由基本的工作流程建模開始，最終希望能藉此建構一套完整校園文書流程的資料模型。

本篇文章結構，第 2 節先對工作流程的特性及以 UML 建模現況進行背景瞭解，第 3 節則對 Petri nets 的特性與相關的延伸功能進行定義，討論 Petri nets 分析工具，並嘗試以 Petri nets 對工作流程基本型建模，在第 4 節我們舉出幾個在校園常見的文書流程，並探究以 Petri nets 來建模之結果，第 5 節則提出我們對未來進一步研究的探討，最後是結語。

2. 背景研究

首先，我們必須瞭解「工作流程 (workflow)」的定義。「工作流程管理聯盟」(Workflow Management Coalition, WfMC)對於工作流程的定義，若翻譯成中文為「根據一系列的程序運作規則，在全部或部分的作業程序自動化的過程中，將文件、資訊或是任務從

一個參與者傳到另一個的程序」[8]，簡單說，就是將一個專案，拆解成數個子步驟，按照既定的程序，透過電腦處理逐步實施完成的整個過程。

至於談到 workflows 的管理，Aalst 指出不論是哪一種類型，不外乎是在正確的時間內，對於指定的人，完成適切的活動[9]。Aalst 進一步闡述，工作流程是以個案為基礎 (case-based)，建立一套工作流程處理程序，即是為了要將類似的個案按既定的程序逐步處理。在個案的處理過程中，包含了諸多會影響流程是否可以完成的條件，例如某個先遣任務 (task) 是否已完成，是否擁有某些必要的資源 (resource)，是否具備執行該流程的角色等，因此，影響流程運行的「路徑選擇」(routing)，是否能被正確的描述，是該工作流程設計的重點。

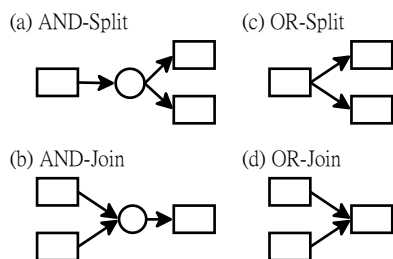


圖 1 四種流程的邏輯基本結構

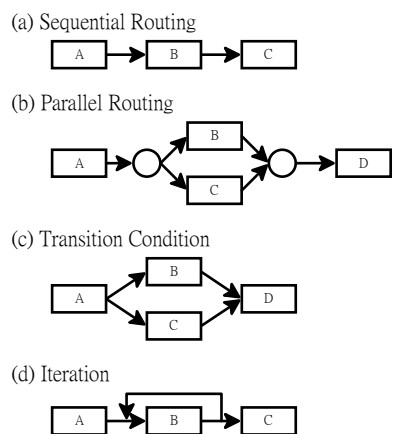


圖 2 四種路徑選擇的結構

圖 1 是 workflows 的邏輯基本結構圖。圖 2 則是四種基本流程路徑。其中，圖 2 (a) Sequential Routing 顧名思義為循序運行，故依照任務 A、B、C 的順序執行；圖 2 (b) 為 Parallel Routing，任務 A 與任務 B 和 C 以 AND-Split 連接，故當任務 A 完成時，任務 B 與任務 C 接著進行，而任務 B 和 C 與任務 D 以 AND-Join 連接，故必須兩者皆完成時，才進入任務 D；圖 2 (c) Transition Condition 結構，與 Parallel Routing 不同之處在於 AND 邏輯結構換成了

OR，所以當任務 A 完成時，依照某種特定的條件，可能進行任務 B 或 C，一旦兩者有一個完成，隨即進入任務 D；最後一個是 Iteration，與 Sequential Routing 類似，只是任務 B 必須重複執行數次，直到達到某種條件後才會進入任務 C。

Gronmo 等人使用在商務流程被廣泛應用的 UML (Unified Modeling Language) 來描述圖 2 之四種路徑基本架構[6]，不過在 Iteration (Loop) 的處理上顯然不夠直覺。另外，Baeyens 也提到使用 UML 在圖形表示上，最大的缺點就是無法區分「狀態」(state) 與「動作」(action)，[7]。且文章最後還建議使用 Petri nets 來作為表達共通的工作流程建模。

對於使用 Petri Nets 於 workflows 的建模而言，Aalst 提出了三點理由[10]：

- (1) 業務邏輯 (business logic) 可以以一種正規且保有圖形化的語言來描述。不論是傳統 Petri net 或是其各種延伸，如：Timed Petri nets, Colored Petri nets, Hierarchical Petri nets 等都已经具有正規的定義與數理證明。因此，像圖 1 四種流程邏輯的基本結構，就可以很容易用 Petri nets 來建模，如圖 3。

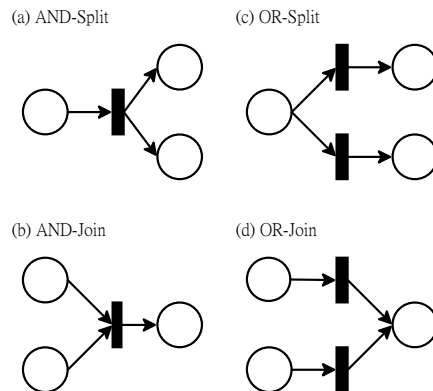


圖 3 將圖 1 改以 Petri nets 來建模

- (2) 以狀態 (state) 為基礎而非事件 (event)。對每個個案而言，以 Petri nets 建模，所有流程運行過程中的狀態，都可以被 Petri nets 明確地表達出來。這對於其他以事件為基礎的建模工具而言，是難以做到的。再者，對於有競爭性質 (選擇性)，如圖 2(c)，很難以表達究竟如何區別 B 與 C，但是 Petri nets 的表達方式就可以很明確。
- (3) Petri nets 具有十分豐富的分析技術。因此，以 Petri nets 對 workflow 建模，

這些相關的分析技術，例如：安全性 (safety)、可達性 (reachability)、死結 (deadlock) 以及不變性 (invariance) 等。結合時間因素，延伸為 timed Petri nets，亦可用來分析執行效能、等候時間、反應時間等。

除了以上這些優點外，我們認為應用 Petri nets 來建模，在工作流程設計之初，除了可以很清楚區分各關卡所需資源外，圖形介面也有助於在設計時與業務需求單位溝通，增進相互對該工作流程的理解，對於流程管理系統而言，使用 Petri nets 還可以在流程實際運行時，隨時提供該流程個案的狀態，估算簽核完成預計時間等，這些優點我們都將在下一節正式定義 Petri nets 後逐一探討，所以在這些因素的考量下，我們選擇 Petri nets 作為校園工作流程建模的工具。

3. 以 Petri nets 為工作流程建模

Petri net 於 1962 年由 Carl Adam Petri 所提出 [3]，基本上是一個有向圖 (directed graph)。除了應用於工作流程外，在生產線、製造業、教育學習、交通運輸、生物資訊等都被應用作為建模與模擬工具。Peterson 提到 Petri nets 用來描述與分析系統之資訊流與控制流是最自然、簡單且有效的方法，尤其是系統中若同時有同步與非同步的活動 [4]。其最主要的基本圖形結構，分別是以圓形來表示狀態，稱為 place，以長條形 (矩形) 表示動作，稱為 transition，再使用有向的 arc (弧線)，一一將它們串連起來。Petri net 的瞬間狀態是由 places 上的 tokens (黑色圓點) 來決定。我們若將工作流程的基本路徑以 Petri nets 來表達，結果如圖 4 所示。為了進一步瞭解 Petri nets 的運作，必須先對 Petri nets 進行正式的定義。

定義 1: Petri net 是一個 4-tuple, $PN=(P,T,I,O)$ ，其中 P 是 places 的集合， $P=\{p_1, p_2, \dots, p_m\}$ ， T 是 transitions 的集合， $T=\{t_1, t_2, \dots, t_n\}$ ， $P \cap T = \emptyset$ ； I 與 O 分別是 T 的 input 與 output 函數， $I(t_j)$ 就是 transition t_j 的 input places 的集合， $O(t_j)$ 就是 transition t_j 的 output places 的集合。

定義 2: 向量 $\mu=(\mu_1, \mu_2, \dots, \mu_m)$ 稱為 Petri nets 的一個「marking」，其中 μ_j 是 place p_j 中 token 的個數。Marking μ 即為 Petri nets 在當時的一個「狀態」。 μ_0 是 Petri nets 一開始 token 分佈的狀態，稱為初始 marking (initial

marking)。

定義 3: Petri nets 執行方式為 transition 的 firing。如果 transition t_j 的每個 input place p_i 都至少有一個 token，則 t_j 稱為 enabled，此時， t_j 就能夠 firing。 t_j fired 之後， t_j 每個 input place p_i 必須消耗一個 token，而 t_j 每個 output place p_o 則被放入一個 token。

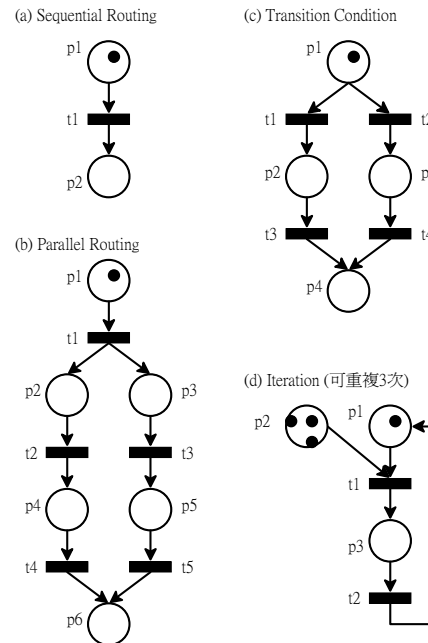


圖 4 以 Petri net 來呈現流程基本路徑

如圖 4(a)是一個簡單的 Petri net，對 transition t_1 而言， p_1 是唯一的輸入 place， p_2 則是唯一的輸出 place。在 place p_1 有一個 token，所以初始 marking 為 $(1, 0)$ 。又因為 place p_1 是 transition t_1 的唯一輸入，故此時的 transition t_1 為 enabled 狀態，一旦 transition t_1 「啟動」(firing)，則將消耗掉 p_1 中的一個 token，同時在 p_2 產生一個 token。此時，Petri net 的狀態變為 $(0, 1)$ 。因為 p_1 中已經沒有 token 了，故 t_1 不再是 enable 狀態，所以整個流程就結束。

依照 Petri nets 三個基本定義，我們在第 3 節所提到，在其他工作流程建模工具不易表達的幾個問題，就可以很容易以 Petri nets 來解決，而且「狀態」(places)與「動作」(transitions)分開以不同的圖形表示，更容易讓人理解。圖 4 就是我們把圖 2 以 Petri nets 來建模的例子。其中，圖 4(b)用來表達需要同步處理的流程；圖 4(c)為表達具有選擇性的流程路徑，相較於圖 2(c)不夠明確的表述方式，對於圖 4(c)而言，只要在 transition t_1 與 t_2 的部分註明 firing

條件，比圖 2(c)的呈現方式容易理解。在下一節我們將回到此圖再進一步討論。

同樣地，圖 4(d)是以 Petri nets 來表達 iteration 結構。在上一節我們曾經提到，圖 2(d)的表達方式很難描述需要重複幾次，因此，我們特別在圖 4(d)增加了一個 place p_2 。 p_2 是用來控制整個迴圈要進行多少次，關鍵就在 p_2 上的 token 數目。以本例而言，初始 marking 為 $(1,3,0)$ 每進行一次迴圈，會用掉 p_2 一個 token，而 p_1 的 token 數不會改變，所以迴圈只能進行 3 次，流程就必須停止。

使用 Petri nets，除了能夠適當的釐清狀態與動作的不同，並且在同步執行、條件選擇與重複執行作明確的表達外，另一個要點的就是要利用 Petri nets 數理基礎來對工作流程模型進行分析。

定義 4： Petri nets 的某個狀態 marking M_n 若可由初始 marking M_0 經過一系列的 transition firing 到達，則稱由 $M_0 \rightarrow M_n$ 為可達 (Reachable)。

可達性 (Reachability) 的驗證，可以用來確保想要的狀態可以到達，相反的，如果不應該出現的狀態在驗證過程中出現了，也表示該流程的設計有問題，這就有助於提早發現問題，不需要等到實際上線時才發現。

定義 5： 給一個 PN 跟起始 marking M_0 對於所有的可達之狀態集合 (Reachability set) 元素 M_i ，都能存在啟動某一 transition t_k 到達下一個狀態 M_j 。則稱 PN 具有存活性 (Liveness)。

PN 具有存活性，可以避免死結 (Deadlock)，而且確保每一個 transition 都可以被 enabled，如果存在有不能被 enabled 的 transition，表示該動作不會被執行到，是很奇怪的事，除非是依照不同起始狀態將 enable 不同的 transition，否則可能是設計有誤。

定義 6： 在 PN 中所有可達的 marking M_j ，其中每個 place 的 token 數都固定小於 k ，則稱該 PN 為 k -bounded。如果 $k=1$ ，亦即在 place 中的 token 數最多是 1，則稱 PN 為 safe。

Bounded 及 safe 特性，是為了確保 token 數不會無限制增加，會造成系統溢位 (overflow)，而且所代表的意義會變成很奇怪。

定義 7： 若 PN 中從任一可達的狀態，最後都會導到最終點，而且結束時除了最終點的 place 以外，其他的 places 完全沒有 token 留下，則該 PN 稱為 sound。

Soundness 的特性，可以確保該 PN 無死結 (deadlocks) 及活結 (livelocks)。

除了以上與工作流程相關的分析技術外，為了讓 Petri nets 能夠更有彈性，更直覺

地成為工作流程建模工具，三種 Petri nets 的延伸應用也將被我們所引用：

- (1) Hierarchical Petri nets (HPN): 雖然傳統典型的 Petri nets 在很多方面已經足以用來為工作流程建模，然而因為流程本身涉及的資源與動作很多，將導致 Petri nets 複雜度過高，HPN 的提出，就是為了解決這個問題。基本上的作法，是讓某些 sub Petri nets 用 places 來取代，一來增加可讀性，減少複雜度，二來適當的設計還可以讓整個建模的過程變成模組化設計。
- (2) Colored Petri nets (CPN): 由於典型的 Petri nets 很難呈現某些物件的屬性，例如要描述一部電腦的使用年限、配備等，因此導入 CPN 的目的，就是要讓這些物件的一些屬性能被描述。在 CPN 中，每一個 token 都有標記 (或顏色)，用以代表不同屬性，而 transition 就可以依照輸入不同顏色的 tokens，決定不同輸出顏色的 tokens，這也意味著要可以訂定一些特別的 transition firing 先決條件。
- (3) Timed Petri nets (TPN): 將時間導入典型的 Petri nets，一方面可以允許系統對時間的要求，例如一般公文都有處理時間的限制，加入時間的觀念後，就可以做到超過一定時間則進行稽催流程，另一方面，在分析系統時，也可以對效能作一番評估。時間可以加在 places、transitions、tokens。

以上這三種延伸，我們都將引用到校園的文書處理流程，另外我們還要導入「抑制弧」(inhibitor arcs)的觀念於 Petri nets，如圖 5。

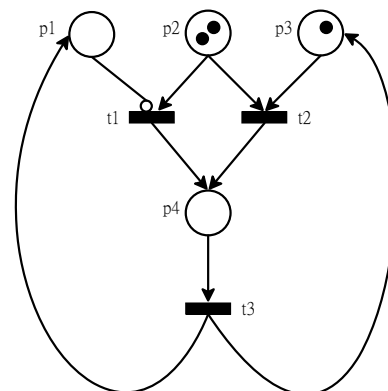


圖 5 具有 inhibitor 的 Petri net (iteration)

定義 8： 如果 transition t_j 連接輸入的 place p_i ($p_i \rightarrow t_j$) 是抑制弧，且 p_i 存在至少一個

token，該 transition t_j 即不能被 enabled [7]。

圖 5 一開始的起始狀態為(0,2,14)， $p1 \rightarrow t1$ 為 inhibitor arc，但是沒有 token，所以 $t1$ 與 $t2$ 都被 enabled，不論是 $t1$ 或是 $t2$ firing 後，接下來一旦 $t3$ fires，此時有一個 token 會被放在 $p1$ 中，這時候抑制弧就會發生作用， $t1$ 被抑制，只剩下 $t2$ 是 enabled 狀態。抑制弧的功用在於進行分工。當某一工作有兩個人有權限去作，但是一旦某甲作了該類型的工作，就限制往後只能由某甲執行。

4. 實例分析與討論

本篇文章希望藉助 Petri nets 及其相關之高階 Petri nets 來為校園內部的文書處理流程建模，依照研考會「文書處理手冊」[10]對於「文書」之定義：「文書，指處理公務或與公務有關，不論其形式或性質如何之一切資料」。由定義可看出文書所涉及範圍相關廣泛，可以說是包羅萬象。簡而言之，機關文書包含了機關單位內所有的簽辦文件。我們習慣性將之粗略地分成電子表單與電子公文兩類。後續的內容將列舉幾種文書處理的實例來說明 Petri nets 如何被應用於工作流程建模。

首先是最簡單的循序處理，如圖 4(a)，以公文六大類之「公告」為例，place $p1$ 可以表示為「擬公告稿」，為一種狀態，transition $t1$ 表示「呈主管核簽核」，是一種「動作」，place $p2$ 表「主管核准發佈」，同樣是一種狀態，當然中間可再進一步細分各種狀態與動作，基本上是屬於單線式的流程。

在這邊我們要作一個 Timed Petri nets 的導入示範。若圖 4(a)改成線上報名電子表單之流程，重新繪製成圖 6。此時，加上限制 $t1$ firing 的時間為上班時間 8:00 至 17:00。雖然 $t1$ 被 enabled，但是若系統時間不是在上班時間，則仍然不允許流程繼續向下走，表示該報名表只能在上班時間被處理。

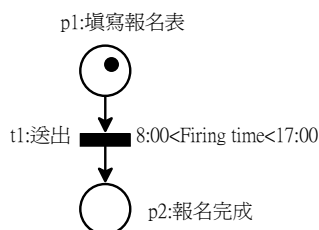


圖 6 以 TPN 呈現線上報名流程

涉及平行同步處理的公文書流程處理，可以公文機關來函的「分會」動作為例，如圖

4(b)，place $p2$ 與 $p3$ 分別表示會辦「資訊中心」與「總務處」兩單位，此時單位承辦擬妥意見後，送各自單位主管核示 (transition $t2$ 與 $t3$ ， $p4$ 與 $p5$ 為主管簽辦意見之狀態)，必須兩單位主管都完成後送出 ($t4$ 與 $t5$ fire)，始能「呈校長核可」($p6$)。在這個例子說明中， $p1$ 其實代表了收文單位內部的簽核過程， $p6$ 則代表分會完成後的後續簽核流程，各自可代表另外一個可能是循序處理的 Petri net，也就是說我們導入了 Hierarchical Petri net 的觀念，很明顯地，這對流程在視覺上的簡化有很大的助益。

具有競爭性質的流程可如圖 4(c)之形式表達，place $p1$ 同時 enable transition $t1$ 與 $t2$ ，形成兩者的競爭。我們以提出「文具需求」的流程為例 (圖 7(a))，place $p1$ 表示「需要文具」， $t1$ 為「向總務單位請領」， $t2$ 為「自行採買」，在這裡，我們再次使用 HPN 的觀念， $p2$ 與 $p3$ 為 HPN，分別代表進入「請領流程」與「採購流程」，這兩個流程可以形成模組，在別的地方也可以重複利用。流程走完後，不論原來是走哪一個流程皆可獲得所需的文具 (進入 $p4$)，而結束流程。

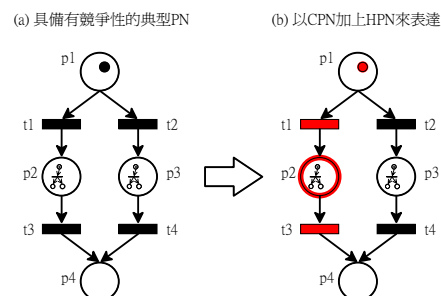


圖 7 以 HPN 與 HCPN 表達文具需求流程

示範 CPN 的應用可將圖 7(a)改以 CPN 表示如圖 7(b)，限制 $t1$ 、 $p2$ 、 $t3$ 只允許紅色的 token 通過，因此當 $p1$ 的起始 token 為紅色時，只有 $t1$ 被 enabled，也就是說若給定起始的 marking 為 (紅 1, 0, 0, 0)，在文具需求的流程處理中，將啟動「向總務處領取」之流程。這是利用不同顏色的 tokens 來決定流程走向。

Iteration 的應用，如果需要限制次數，可使用 multi-tokens 的觀念，例如要限制有限資源的使用。以圖 4(d)模擬派車單流程為例， $p2$ 可以視為可用公務車數，以該圖而言，三個 tokens 表示車輛只有三台， $p1$ 為填寫派車申請單，一旦該流程走三次， $p2$ 上的 tokens 數用完後，在沒有補充新的 tokens 之前 (公務車完成行程返回或採購新車等)， $t1$ 就不能被 enabled，如有新的派車申請單就必須等待。

帶有抑制弧的 Petri nets 的應用，以圖 5

為例，是一個具有一條抑制弧的 iteration 流程。我們以開學時學生申請電腦教室流程為例說明。假設社團可以選擇申請教室 101 一整學期每週使用一次(t1)，或是勾選按需求申請教室 102，但可隨時申請(t2)，如果一開始選擇了教室 102，則就不允許再借教室 101（抑制弧的功能使 t1 不能被 enabled），反之亦然。抑制弧可讓 PN 有更複雜的應用。

5. 未來研究方向

由於文書種類繁多，本篇文章僅就部分工作流程提出說明，並未全面對校園各種電子表單與公文簽核流程進一步分類，我們認為，應該把各種流程作一番歸納整理，在基本的流程路徑上，針對校園內部文書處理整理一份專屬的「文書建構基本元件」，並以 Petri net 表達，所有的文書流程都以這些基本元件重新建模，形成 HPN 建構模組元件，讓工作流程的建模變得更簡易，也更容易驗證其正確性。

此外，這樣做還有另外一個好處，那就是在文書作業流程的處理過程中，對於經常遇到動態的「加會」、與「加簽」動作，將會有所幫助。利用這些預先定義好的 Petri nets 基本建構元件，在設計流程管理系統時，便可在需要時以動態加入元件的方式處理，不過這部分仍需要未來進一步的規範與定義。

分析各種文書處理流程，加入適當的時間參數，除了能進行效能分析外，也有助於對文書處理的稽核與稽核動作，處理時效的分析對於文書處理流程的改造，也能作為參考依據。

此外，Petri nets 的模擬工具，例如：CPN Tools，可以加入作為對於流程正確性的驗證與效能檢測，對於工作流程再造預期可有相當之效益。

6. 結語

為有效提升校園內行政作業的效率，校園文書處理電子化，是各校普遍的需求。導入一套工作流程管理系統並不難，然而，這其中成敗的關鍵之一，就在於流程的設計。但是，業務需求提出單位與流程設計人員之間，首先就要面對如何溝通的問題，如何能清楚有效地描述整個流程設計，就在於如何選用一套合適的工具來建模。

以數理為基礎的 Petri nets 在六十年代被提出後，從最原始的應用，到包含 Hierarchical PN、Timed PN、Colored PN 以及具有 Inhibitor arcs 的 PN 等由 PN 延伸之高階 Petri nets 的應用，每一項在背景的研究都十分紮實。圖形化

的介面，加上豐富的分析工具與模擬工具，讓它在學術研究領域佔有相當的地位。而以應用在工作流程上的建模而言，將「狀態」與「動作」分離的天性，對於工作流程的建模而言，不再是語意不清，難以理解。Petri nets 簡單的圖形呈現方式，則有助於瞭解整個流程的設計理念；分析工具的導入，能預先發現錯誤，並適時提供流程正確性的驗證；模擬工具的加入，有效地改善流程效能，排除不合理狀態；各種 Petri nets 延伸版本的支援，更是對工作流程建模上提供更多更直覺的應用。

參考文獻

- [1] 行政院研究發展考核委員會。2005。文書處理手冊第三版。
<http://www.ey.gov.tw/public/Attachment/51230152371.pdf>
- [2] 行政院研究發展考核委員會。2006。行政院電子化政府報告書(93-94 年度)。
<http://www.rdec.gov.tw/public/Data/711615155271.pdf>
- [3] C. A. Petri. "Kommunikation mit Automaten," PhD thesis, Institut für instrumentelle Mathematik, Bonn, 1962.
- [4] J. L. Peterson, "Petri Nets," ACM Comput. Surv. (CSUR), Vol. 9, Issue 3, pp. 223-252, Sep. 1977.
- [5] N. Busi, "Analysis issues in Petri nets with inhibitor arcs," Theor. Comput. Sci. 275, 1-2, pp. 127-177, Mar. 2002.
- [6] R. Gronmo, and I. Solheim, "Towards Modeling Web Service Composition In UML", In Proc. WSMAI, pp. 72-86, 2004.
- [7] T. Baeyens, "The state of workflow," TheServerSide, 2004,
<http://www.theserverside.com/tt/articles/article.tss?l=Workflow>
- [8] WfMC, "Workflow Management Coalition Terminology and Glossary," (WfMC – TC - 1011). Technical report, Workflow Management Coalition, Brussels, 1996.
- [9] W.M.P van der Aalst, "The Application of Petri Nets to Workflow Management," J. of Circuits, Systems, and Computers, Vol. 8, (1):pp. 21-66, 1998.
- [10] W.M.P. van der Aalst. "Three Good Reasons for Using a Petri-net-based Workflow Management System," In S. Navathe and T. Wakayama, editors, Proceedings of the International Working Conference on Information and Process Integration in Enterprises (IPIC' 96), pp. 179 – 201, Cambridge, Massachusetts, Nov 1996.