

建構一種多媒體派翠西網路系統之 延遲處理模型

戴建耘、饒達欽、邱清文、陳世旺 *

國立台灣師範大學工業教育學系

* 國立台灣師範大學資訊教育學系

摘 要

本研究提出一種以派翠西網路(petri-net)作為分析多媒體延遲與細分部份不易模擬的改善處理方式。此種方式，可輔助作為快速計算出多媒體派翠西網路的可達樹及可達矩陣。有鑑於同一多媒體派翠西網路不同的初始值會造成不同的可達樹及可達矩陣，研究中提出一種和初始值無關的運算方法。文中並在可達矩陣處運用判斷網路品質的複雜度計算公式，供系統架構者做為評估網路的品質。而本研究提供之處理模型亦可作為多媒體系統與工具軟體在使用派翠西網路作分析時，結合個體導向觀念做設計的支援技術之一。

關鍵字：派翠西網路(Petri nets)，P-T表，T-P表，多媒體(Multimedia)

壹、前言

一般常用的派翠西網路架構和基本符號如圖一所示 [4][5][7]。本研究亦採用之，但此四種符號的內涵定義用途如表一所示。表一所使用的名詞乃引用個體導向技術中的名詞以及 OPNets[1][8]的觀念，以支援在本研究中提出的 Bottom- Up 個體導向式多媒體軟體分析與發展整體架構，作為系統發展方法中的一部份。圖一的 (a) 代表位置，表示由接收 transition 傳來的 data buffer 或 message buffer，圖一的 (b) 轉換是細分的點，亦代表個體導向技術中的個體或者是個體內的方法。圖一的 (c) 代表有向弧，在本文中代表傳遞的訊息或者是資料流。圖一的 (d) 代表權杖，代表系統模擬的狀況。



圖一 派曲網路基本符號

表一、在 Petri Nets 與 OPNets 符號所代表的意義 (請參考圖(1))

基本元件 定義 網路	(a)Place	(b)Transition	(c)Arc	(d)Token
Petri Nets	位 置	轉 換	有 向 弧	權 杖
本研究內涵與用途	Data Member or message buffer	Member Function	Message or data of flow	狀 態

依據派翠西網路基本工作定義為：訊息經由 Arc 傳遞到 Place 中儲存，等待 Transition 上的條件滿足後，訊息經由 Transition 處理過後，傳遞下一個訊息或資料到下一個 Place。[4][5]一般分析的方式不考慮節點間的延遲，但是建構多媒體派翠網路時，媒

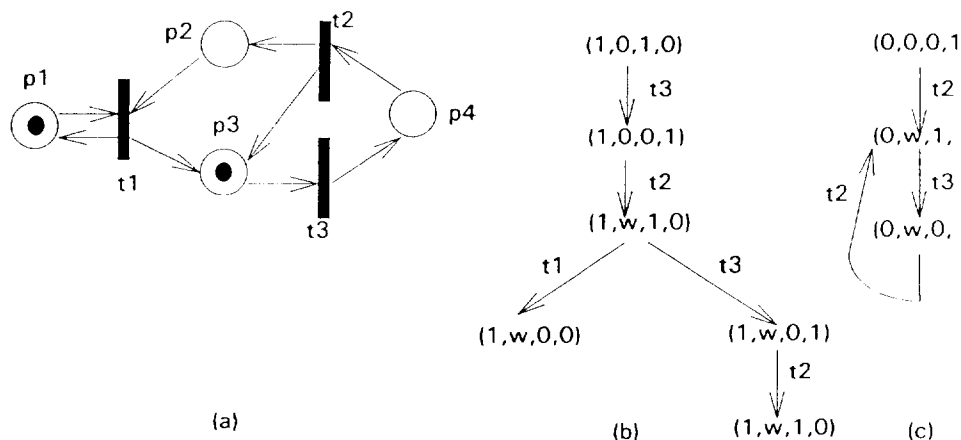
體與媒體間處理經常需要同步、延遲或異步等不同方式處理，所以節點間的延遲將成爲系統重要的參數。傳統的派翠西網路輔助的工具具有可達樹及可達矩陣 [1][4][5]，本研究將可達樹及可達矩陣擴充成爲 P-T 表、T-P 表、T-T 表、P-P 表，可輔助電腦快速計算不同初始值的可達樹及可達矩陣。最後將可達樹用一簡單的複雜度公式計算，所得的結果可以輔助觀察網路的品質。作爲交談式多媒體系統的初始值設定參考。

貳、派翠西網路之可達樹及可達矩陣

假設已知的典型多媒體以派翠西網路分析之表示結果如圖一 (a)，其可達樹的關係圖分別如圖二 (a) 與 (b) 及表二與表三 (其中 $w = \infty$)，此種表示方式有以下幾項問題值得探討。

一、不同的初始值會造成不同的可達樹或可達矩陣

例如：圖二 (b) 及 (c) 針對相同的網路結構，卻有兩種不同的可達樹結果。因此，不同的網路初始值可能會影響網路本身的工作狀態轉移。在不同的初始值中，可能有些初始值會造成死結或無限迴圈，而系統設計者必須先經由重新計算可達樹後才可能知道。



圖二 派翠西網路與可達樹的例子 [1]

表二、圖二 (b) 初始值為 (1,0,1,0) 之可達矩陣

$(p1, p2, p3, p4)_{n+1}$	$1,0,1,0$	$1,0,0,1$	$1,w,1,0$	$1,w,0,0$	$1,w,0,1$	$1,w,1,0$
$(p1, p2, p3, p4)_n$						
$1,0,1,0$		1,t3				
$1,0,0,1$			1,t2			
$1,w,1,0$				1,t1	1,t3	
$1,w,0,0$						
$1,w,0,1$						1,t2
$1,w,1,0$						

表三、圖二 (c) 初始值為 (0,0,0,1) 之可達矩陣

$(p1, p2, p3, p4)_{n+1}$	$0,0,0,1$	$0,w,1,0$	$0,w,0,1$
reachable $(p1, p2, p3, p4)_n$			
$0,0,0,1$		1,t2	
$0,w,1,0$			1,t3
$0,w,0,1$			

二、不考慮節點間的延遲

在實際作多媒體系統塑模分析時，延遲的問題實際上是存在的。例如：以 Place 做細分時，延遲將發生在 Place 上；以 Transition 為細分時，延遲將發生在 Transition 上。即使以狀態轉移圖也不易表示在同時間的 Place Enable 或 Transition Enable，所發生的不同狀況。多媒體系統可能牽涉到不同系統間的同步、中斷、和執行指令的時間差異，所以同一個系統軟體也可能會因為不同的系統環境而影響到執行的結果。

三、系統運作時對於不同的初始值會有不同反應

初始設計並不能考慮所有狀況，或許某些狀況僅需更改部份初始值便可達到更改系統目的，但可達樹卻需要重新繪製。（可達矩陣可以解決這部份的問題，但是無法考慮延遲問題）。

這些缺點使得多媒體派翠西網路在系統模擬時，並不能真實反應系統狀況。使用本文所建立的表格後，這些缺點可望改善。

叁、P-T 表、T-P 表、T-T 表、 P-P 表製作技術

P-T(place state to transition state)及 T-P(transition state to place state)表主要的特性可考慮所有狀態，故與網路初始值無關。P-T 及 T-P 表的製作方式相近，使用 P-T 表可將現在的 Place 狀態推演到下一個 Transition Enable 的狀態。 $(P1,P2) = \{(0,0),(0,1),(1,1),(1,0)\}$ 是指在 Place 上所需 Transition Enable 的最低條件。所有製表時，皆不考慮 place 間時間延遲的狀態，將所有延遲當作零：有 Enable 時填入 1；沒有時則填 0。（請參考表四的 P-T 轉換實列表）

T-P 表則較特殊，是由現在 Transition 觸發的狀態導出下一個 Place 的狀態，可由下列表示式做說明：（請參考表五。）

以 $(t1,t2,t3)_n = (0,1,0)$ 時為例，

則 $(P1,P2,P3,P4)_{n+1} = (0,1,1,0) + (P1,P2,P3,P4)_n - I(T2)_n$

其中， n 表示現在狀況， $n+1$ 表示下一狀態。在上式的 $I(T_n)$ 請參考表六，其他的表示法之通式請參考下列表示式：

$$I(T1,T2,\dots,Tn) = I(T1) \cup I(T2) \cup \dots \cup I(Tn)$$

只再需依照圖三的處理流程，充分運用上述的表格，即可快速將不同的初始值，製作可達樹。其處理流程如圖三，演算方式說明如下：

表四、P-T轉換實例表

$(p1, p2)_n$	$(t1, t2, t3)_{n+1}$	0,0	0,1	1,1	1,0
	$(p3, p4)_n$				
	0,0	(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)
	0,1	(0,1,0)	(0,1,0)	(0,1,0)	(0,1,0)
	1,1	(0,1,0)	(0,1,1)	(1,1,1)	(0,1,0)
	1,0	(0,0,1)	(0,0,1)	(1,0,1)	(0,0,1)

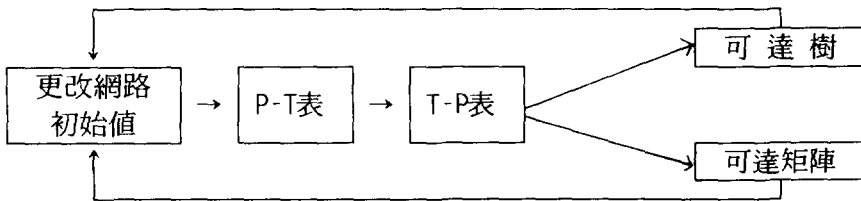
表五、T-P轉換實例

$(t1, t2)_n$	$(p1, p2, p3, p4)_{n+1}$	P1	P2	P3	P4
	$t3_n$				
	0	$(0,0,0,0)+$ $(P1, P2, P3, P4)_n$	$(0,1,1,0)+$ $(P1, P2, P3, P4)_n$ $-I(T2)_n$	$(1,1,2,0)+$ $(P1, P2, P3, P4)_n$ $-I(T1, T2)_n$	$(1,0,1,0)+$ $(P1, P2, P3, P4)_n$ $-I(T1)_n$
	1	$(0,0,0,1)+$ $(P1, P2, P3, P4)_n$ $-I(T3)_n$	$(0,1,1,1)+$ $(P1, P2, P3, P4)_n$ $-I(T2, T3)_n$	$(1,1,2,1)+$ $(P1, P2, P3, P4)_n$ $-I(T1, T2, T3)_n$	$(1,0,0,1)+$ $(P1, P2, P3, P4)_n$ $-I(T1, T3)_n$
	Sum of token	$M'(P)=M(P)-W(P,T)+W(T,P)$			

表六、Transition的輸入與Place關係實例

Place	Token number	P1	P2	P3	P4
	input of Transition				
	I(T1)	1	1	0	0
	I(T2)	0	0	0	1
	I(T3)	0	0	1	0

1. 先決定網路初始值。
2. 查 P-T 表決定下一狀態中的 Transition 的 Enable 狀況。
3. 若查到的 $(t1,t2,t3)_{n+1}$ 有兩個以上的 Transition enable，則進行分支；若僅一個 Transition enable 則直接進行下一步驟。
4. 由上步驟查到的 $(t1,t2,t3)_{n+1}$ 做為 T-P 表的輸入，求得 $(P1,P2,P3,P4)_{n+1}$ 。
5. 記錄該狀態，重複步驟 2。
6. 有重複的 $(P1,P2,P3,P4)_{n+1}$ 出現時，即結束。



圖三 可達樹及可達矩陣處理流程

依照上述步驟，即可產生可達樹。由 P-T 表和 T-P 表可以推導出 $P(n)-P(n+1)$ 表及 $T(n)-T(n+1)$ 表，如表四與表五。透過這些表格同樣可以導出可達樹及可達矩陣。但是推導出可達樹及可達矩陣卻無法如圖四的時序圖可清楚表示時間延遲圖。

表七、T-T 表實例

$(t1, t2, t3)_{n+1} \backslash (t1, t2)_n$	0,0	0,1	1,1	1,0
0	0,0,0	0,0,1	1,0,1	0,0,1
1	0,1,0	0,1,1	(1,1,1)	0,1,0

表八、P-P 表實例

$(p1, p2, p3, p4)_{n+1} \backslash (p1, p2)_n$	0,0	0,1	1,1	1,0
0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0
0,1	0,1,1,0	0,1,1,0	0,1,1,0	0,1,1,0
1,1	0,1,1,1	0,1,1,1	1,1,2,1	0,1,1,1
1,0	0,0,0,1	0,0,0,1	1,0,0,1	0,0,0,1

本研究採用以 Transition 細分的技術，所以主要延遲將發生在 Transition 上。如以圖五為例，當 Token 傳遞到細分的 Transition 上時，會進入副程序中執行，或者可以使用延遲來代表，當延遲發生時，可能其他的 Transition 正好在 Wait 或 Enable。所以，Transition 上的狀況可歸納為四種，process、wait、enable 和 disable。透過上述的四類表格可以很快的導出時序圖實例，如圖四所示。該圖中的 3 個 transition 分別有不同的延遲， $t1(w)=2$ 、 $t2(w)=1$ 、 $t3(w)=2$ ，其中的數字表示延遲的時間（單位可自定）。觸發的先後依序為 t1、t2、t3。在時序圖上：

1. 0 表示 transition 在 wait 狀態。
2. 1 表示正在處理中。
3. 0 → 1 表示 enable。
4. 1 → 0 表示 disable。

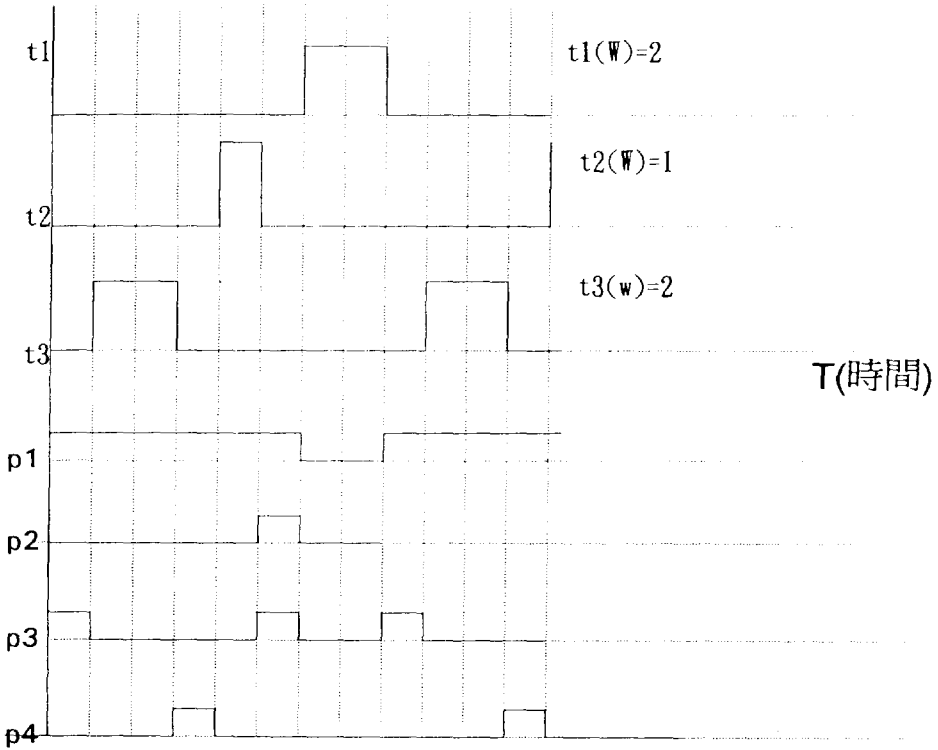
具有延遲特性的網路處理特點為：

1. 在 wait 狀態下的 transition 可接受輸入 token。
2. 在 process 狀態下的 transition 可接受輸入 token。
3. 在 enable 狀態下，將輸入的 token 移入 transition 中。
4. 在 disable 狀態下，將處理過後的 token 釋出。

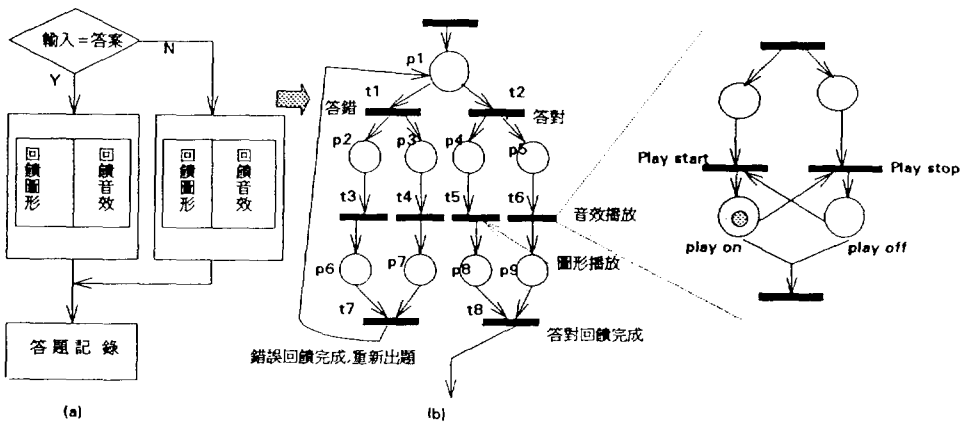
得出時序圖的步驟如下：

1. 將網路初始值輸入網路中。
2. 並由 P-T 表中查出下一時間決定那些 transition enable。
3. 將 enable transition 在時序圖中標示。
4. 將 enable transition 的輸入端移走。
5. 重複步驟 2，但處理中若有 transition disable，則將 token 移出至 transition 的輸出端。

依據上述步驟即可得出具有多媒體特性中延遲效果的圖四媒體時序圖。



圖四 多媒體系統 Transition 及 Place 時序圖例



圖五 多媒體互動式輸入派翠西網路圖

肆、複雜度分析

可達樹的複雜度一般可用的複雜度計算公式如下：[6]

$$CS = \frac{(\text{Max}-\text{CD})}{(\text{Max} - \text{Min})}$$

$$\text{Max} = (n^2-n)C$$

$$\text{Min} = (n^2-n)$$

其中，Max 代表網路中節點可能的最長連接總和，Min 表示網路節點連接的最小連接和，C 代表節點數。運用此一公式計算如下：以圖五為例，其可達樹如圖六所示。因此，可達樹複雜度計算過程為：

1. 已知

$$n=9(\text{節點數})$$

$$C=9(\text{節點數})$$

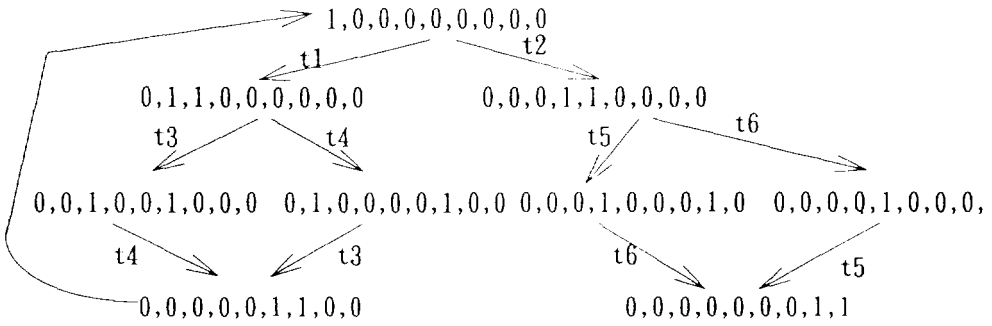
$$CD = 514$$

2. 複雜度計算

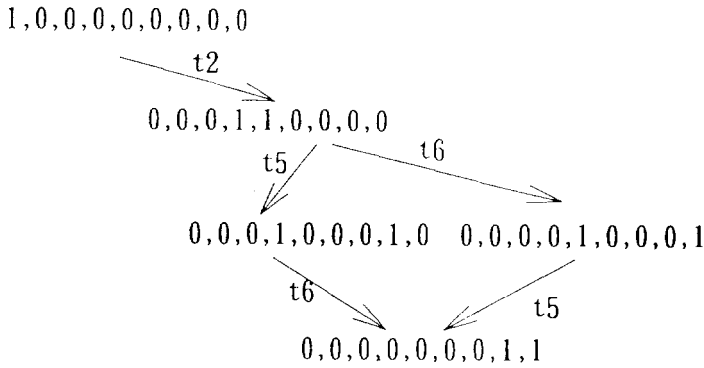
$$CS = \frac{(\text{Max}-\text{CD})}{(\text{Max} - \text{Min})} = \frac{(n^2 - n)C - \text{CD}}{(n^2 - n)C - (n^2 - n)} = \frac{(9^2 - 9)9 - 514}{(9^2 - 9)9 - (9^2 - 9)} = 0.233$$

如果只計算出複雜度並不能直接和不同網路比較，因為影響網路的複雜度有更多的因素，例如：細分的程度、簡化網路的程度、網路初始值、狀態的多寡等。所以單是計算整體的複雜度對於系統架構者的幫助並不大，以下是以配合不同的狀態與路徑來進一步說明網路複雜度的意義。

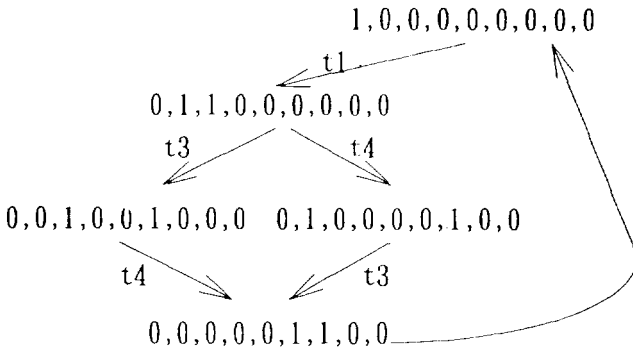
由圖七與圖八，再分別畫出圖五互動式多媒體的答對與答錯的可達樹，由計算複雜度的公式可得複雜度分別為 0.3875（答對時）和 0.6250（答錯時）。這些數字可以明顯看出對於答錯者在本系統內會得到一個較複雜的網路，複雜度計算僅能輔助系統架構者評估不同路徑或不同初始值所造成網路相對複雜的程度，對於其數值並沒有絕對的意義。例如：複雜度 0.6250（答錯時）並不能說成是複雜度 0.3875（答對時）的兩倍。這個數值本身僅可代表當時網路連接的複雜性，不可以用量化的觀念看他。



圖六 圖五 (b) 之可達樹實例



圖七 圖五 (b) 使用者答對後之可達樹



圖八 圖五 (b) 使用者答錯後之可達樹

陸、本模式之應用與發展

由圖九很容易了解本研究所提出的處理方式，在整個系統發展過程中的關係位置。本研究提出的方法僅是完整系統中的一部份，將來是提供一種以 Bottom-Up 發展 OO 系統方法中的一個環節，而這些表格所得的數值，將可作為與個體導向技術結合時的重要關鍵參考資料。

傳統程式設計師撰寫程式語言，通常為 Function oriented[2][3]。若是系統一開始便使用 Object oriented 分析設計，系統就先以 Object 為考量，但是一般系統若是物件不明顯，或者處理的程式不知道分給哪一個 Object 時，Function oriented 這時就有存在的價值。況且一般顯而易見的是 Instance，並不是 Class，因此若是直接建立 Class library，可能有困難。如果能結合 Function oriented 和 Object oriented 的優點，先期使用 Function oriented 找出 Instance，再由 Instance「自動」連結成 Class library，系統發展者將會在不知不覺中已經使用到 Object oriented 的優點了。

派翠西網路對於動態行為描述的能力很強，但發展時缺乏物件化的觀念，不容易得到個體導向的再使用、資料封裝、繼承、同名異式等優點。若結合個體導向的軟體發展方法與派翠西網路描述系統的特性，這將會在軟體工程上有很大的效益 [1][8]。本研究提出的 Bottom-Up 的方法將有助於實現此一目標。

圖九 轉換 petri nets 成為 class library 的自動化系統架構圖

柒、結論與建議

派翠西網路對於動態性網路的模擬能力是受肯定的。但是和個體導向技術或者細分技術結合時，網路的延遲、不同的初始值等，都影響到系統建模的工作，本論文針對考慮網路延遲和不同初始值提出建議。其重點如下：

1. 定義 Object-oriented 和 Petri nets 之間的對應關係。
2. 定義 P-T 表、T-P 表、P-P 表、T-T 表。
3. 快速輔助製作可達樹即可達矩陣。
4. 提供複雜度計算公式。

在未來研究方向可專注於派翠西網路與個體導向技術的充分結合，使得整個 **Bottom-Up** 的發展方法能有所突破。

參考文獻

1. D. C. Rau, C. Y. Dai , M. S. Lin, and S. W. Chen, "Applying Petri Nets to Multimedia Flow Analysis," *LASTED International Conference Modelling and Simulation*, 1994
2. E. Yourdon and L. L. Constantine, structured design: fundamentals of a discipline of computer program and systems, *Prentice Hall, Englewood Cliffs, N.J.*, 1979.
3. E. Yourdon, Modern structured analysis, *Prentice Hall, Englewood Cliffs, N.J.*, 1989
4. J.L.Peterson, Petri net theory and the modeling of systems, *Prentice-Hall, Englewood Cliffs, N.J.*, 1981
5. L. Anneberg and H. Singh, "Petri net approach for requirement analysis phase of software engineering," *Proceedings of the Twenty-First Annual Pittsburgh Conference, Modeling and Simulation*, vol.21, part 3, May 1990, pp 1015-1019
6. R. A., E. Botafogo, And Shneiderman, B. Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. *ACM Trans. on Information System* 10(2), April 1992, pp.142-180
7. W. Reisig, Petri nets : an introduction., New York: SpringerVerlag, 1985
8. Y. K. Lee and S. J. Park, "OPNets:an object-oriented high-level Petri net model for real-time system modeling", *J. Systems software*, vol.20, no.99, pp69-86, 1993.
9. Chien-Yun Dai, Dar-Chin Rau and Ching-Wen Chiou, "*Virtual Classroom Petri Nets: Constructing the Model of Virtual Classroom With Petri Nets*", Proceedings, LASTED International Conference Modelling and Simulation - MS'94, Pittsburgh, Pennsylvania, April 17-29, 1995
10. Chien-Yun Dai, Dar-Chin Rau and Ching-Wen Chiou, "*Electronic-Shops Virtual Classroom (EVC): The Model of Virtual Classroom in Electronic Shops with Petri Nets*", Proceedings of the 1995 International Conference on Engineering Education-Taipei, May 18-20, 1995

Construct a Process-model for Multimedia Petri-nets in Delaying and Multi-layer Simulation

C. Y. Dai 、 D. C. Rau 、 C. W. Chiou 、 S. W. Chen*

Department of Industrial Education

*Department of Information and Computer Education

National Taiwan Normal University

Abstract

This paper provides a method to improve the weakness of multimedia Petri nets in both the delay of system process and multi-layer simulation. The paper also presents an assistant method which quickly calculates the reachable tree and reachable matrix of the multimedia Petri nets. Although there are different reachable trees and reachable matrixes in the same multimedia Petri nets, the method is independent of the initial value. A formula is used to evaluate the Petri nets complexity for various of initial states. The proposed process model can assist in developing a system from analysis phase by Petri nets to O.O. Programming.

Keyword: *Petri nets, P-T table, T-P table, Multimedia*