

私立東海大學
資訊工程與科學所
碩士論文

指導教授：朱正忠 教授

轉換時間圖至物件導向時間派翠網路以進行時
間性驗證

Verifying Time Properties by Transforming Timing
Diagram to Object-Oriented Time Petri Net

研究生：蔡祁名

中華民國九十四年七月

中文摘要

在系統發展中，時間因素是開發人員相當注重的特性之一，特別是在今天有著越來越多的即時性系統充斥在我們的生活之中，為此，在 UML 2.0 中，OMG(Object Management Group)新增了時間圖(Timing Diagram)此一新圖形，讓開發人員能針對時間做進一步的描述，但 UML 為一塑模工具(Modeling language)，並沒有提供進行分析驗證的相關方法，因此，開發人員在使用時間圖塑模時難以察覺其中隱含的錯誤。

在驗證方面，派翠網路(Petri Net)相關的各項技術由於多年來有不少學者研究出針對各種不同的領域的驗證技術，已有些學者的研究是利用 UML 之塑模後模型配合派翠網路之各項特性以進行分析。

在本論文的著重點為 UML 2.0 新增之時間圖，為此，我們提出一轉換時間圖至物件導向時間派翠網路(Object-Oriented Petri Nets)之機制，包含對時間圖、物件導向時間派翠網路之正規化規格及轉換規則，再轉換後利用派翠網路之各種分析能力對時間特性進行分析驗證，在本論文使用之範例為一簡單的行動通訊裝置與基地台之互動，經時間圖轉換至物件導向時間派翠網路後進行可達性分析(Reachability Analysis)，以驗證需求規格制定是否正確。

關鍵字：時間圖、物件導向時間派翠網路、可達性分析

Abstract

In system software development, time is a very important factor to the quality of the systems. Especially there are more and more real time systems and embedded systems around people. Therefore, in UML 2.0 standard, OMG developed a new diagram called timing diagram which is used to describe timing properties and interaction between objects.

However, UML is a modeling language, it doesn't provide the technique to analyze and verify. For this reason, developers use timing diagram to describe the time properties but hard to know if there has some mistakes in timing diagram.

The techniques of Petri nets are researched for years, and there are various analysis techniques to different domain. Some scholars' research is to analyze the UML model by Petri net's analyzing properties.

The focus of this thesis is on timing diagram. For this reason, we propose a mechanism to transform timing diagram to object-oriented time Petri net, it includes the specification of the timing diagram, the specification of the object-oriented time Petri net, and the transformation rules. After transforming, we utilize the analyzing abilities of Petri net to verify the timing properties. The example of this thesis is use the reachability analysis of Petri net to verify the correctness of the requirement.

Keyword : Timing diagram, Object-oriented Time Petri Net, Reachability Analysis

目錄

| | |
|------------------------------|-----------|
| 目錄..... | I |
| 圖目錄 | II |
| 第 1 章 序論..... | 1 |
| 1.1 研究動機與目的 | 1 |
| 1.2 章節安排 | 3 |
| 第 2 章 背景知識及相關研究 | 4 |
| 2.1 時間圖 | 4 |
| 2.2 派翠網路 | 8 |
| 2.3 物件導向派翠網路 | 14 |
| 2.4 時間派翠網路 | 19 |
| 第 3 章 轉換規則 | 21 |
| 3.1 本論文之方法流程 | 21 |
| 3.2 時間圖規格 | 22 |
| 3.3 物件導向時間派翠網路規格 | 26 |
| 3.4 轉換規則 | 28 |
| 第 4 章 範例及應用 | 38 |
| 4.1 圖形轉換範例 | 38 |
| 4.2 應用範例 | 45 |
| 第 5 章 結論與未來方向 | 49 |
| 參考文獻 | 51 |

圖目錄

| | | |
|------|--------------------------|----|
| 圖 1 | 時間圖圖示範例[5] | 4 |
| 圖 2 | 時間圖範例 | 7 |
| 圖 3 | 派翠網路之用途 | 9 |
| 圖 4 | 派翠網路的靜態基本圖形[19] | 9 |
| 圖 5 | 派翠網路的動態表示[7] | 10 |
| 圖 6 | 五種派翠網路塑模之類型 | 11 |
| 圖 7 | 可達樹 | 13 |
| 圖 8 | 物件導向派翠網路範例 | 17 |
| 圖 9 | 包含物件 AA 與物件 AB 之物件導向派翠網路 | 17 |
| 圖 10 | 私有物件 AA 與 AB 的物件導向派翠網路 | 18 |
| 圖 11 | 時間派翠網路範例 | 19 |
| 圖 12 | 本論文提出之方法架構 | 21 |
| 圖 13 | Step3 之示意圖 | 35 |
| 圖 14 | 範例之簡易架構示意圖 | 38 |
| 圖 15 | 依範例需求表格所繪製之時間圖 | 40 |
| 圖 16 | 轉換圖 15 所得之物件導向時間派翠網路 | 43 |
| 圖 17 | 依[9]之方法計算圖 16 所得之可達樹 | 47 |
| 圖 18 | 可達樹表示之分析結果 | 48 |

表目錄

| | | |
|------|------------------------|----|
| 表 1 | 時間圖之各項標記 | 5 |
| 表 2 | 派翠網路與其他常用工具之比較 | 8 |
| 表 3 | 派翠網路中位置與轉換的解釋[18] | 10 |
| 表 4 | 時間圖規格 | 25 |
| 表 5 | 物件導向時間派翠網路規格 | 27 |
| 表 6 | Step1 之時間規格示意表 | 29 |
| 表 7 | Step1 之物件導向時間派翠網路規格示意表 | 30 |
| 表 8 | Step2 之時間規格示意表 | 32 |
| 表 9 | Step2 之物件導向時間派翠網路規格示意表 | 33 |
| 表 10 | Step3 之時間規格示意表 | 36 |
| 表 11 | Step3 之物件導向時間派翠網路規格示意表 | 37 |
| 表 12 | 範例之時間性需求表格 | 39 |
| 表 13 | 圖 16 中各位置之說明 | 44 |

第 1 章 序論

1.1 研究動機與目的

即時性系統在現在這個分秒必爭的時代所受到的重視可說是與日俱增，尤其是在許多嵌入式系統的應用範疇中，時間是在各項評估中一項不可避免的重要因素。在我們的日常生活中有越來越多的即時性系統幫助我們解決各種問題，從家庭電器的各種小型家電、辦公室系統到大眾公共設施、工業自動控制等領域都有日漸增多的即時性系統，對時間的要求亦是日漸嚴格；即便是在個人電腦及網際網路應用中，時間回應也是非常重要的一個度量標準。因此 OMG(Object Management Group)在 UML(Unified Modeling Language) 2.0 中新增了時間圖(Timing diagram)以描述時間對系統中物件狀態的改變以及各物件間的互動情形 [5]。

在需求制定階段，時間圖可以使需求規格對於描述時間性上有著較為明瞭、易懂的方式，比較起以往使用文字敘述的方式，時間圖可以清楚地表達出物件之中各項狀態受到事件觸發影響後之改變，同時，在物件與其他物件間的互動方面，時間圖亦提供了良好的陳述方式，物件中事件發生除了改變自己的狀態外，亦可能會造成其他物件狀態改變之影響，在時間圖中能明確地將這些交互關係繪出，以方便使用者了解、溝通。

但 UML 是種物件導向的塑模語言(Modeling Language)，一般使用於需求制定以及設計階段，但 UML 在分析方面仍然需要其他技術方法與工具進行輔助，UML 本身並無法做出分析以判斷需求及設計是否正確，為了因應此問題，許多學者應用其他各方面的技術來進行 UML 設計與分析的驗證如[1][3]，同時，越來

越多的 UML 工具除了提供塑模的功能以外，還提供了開發、測試等整合性功能，甚至可以結合其他相關工具進行專案管理的，如 Borland 公司的 Together、CaliberRM、OptimizeitSuite、StarTeam 等工具即提供了相當強大的整合開發環境(Integrated Development Environment)[22]。

在計算機科學領域中，在 1962 年 C.A. Petri 所提出的派翠網路(Petri Nets)是常被用來進行設計、分析與驗證的工具之一，在設計同步、非同步、選擇、分散、決策及推測等系統模型的建構上經常被使用，關於派翠網路的研究及應用，已有許多的專家學者投注了相當龐大的心力，發展出針對不同問題及領域之異質特性的各種派翠網路，如隨機派翠網路(Stochastic Petri Nets)、彩色派翠網路(Colored Petri Nets)、物件導向派翠網路(Object-oriented Petri Nets)、模糊派翠網路(Fuzzy Petri Nets)和時間派翠網路(Time Petri Nets)等，進而可以應用於如作業系統、網路協定等不同的領域[12][18][21][23]。

在處理時間性問題方面，已有不少學者提出使用派翠網路進行驗證的相關方法[9][16]，在本論文中，我們將以轉換圖形規格的方式，並利用之前學者提出之時間派翠網路之方法[9]進行其關於時間性之驗證，因 UML 2.0 新增之時間圖提供了使用者能對物件的互動、時間的限制與呈現上做進一步的描述，然而卻缺乏分析的機制來檢驗圖形內容是否有錯誤或矛盾之處，因此，本論文提出轉換時間圖至物件導向時間派翠網路的轉換規則，再轉換至物件導向時間派翠網路之後，應用物件導向時間派翠網路之特性進行可達性分析(Reachability Analysis)為範例，對系統做出時間性之驗證。

1.2 章節安排

論文的章節安排分述如下：

第二章 背景知識及相關研究，此章說明時間圖的觀念及其中所使用的標記，以及派翠網路的基本特性、應用和物件導向時間派翠網路的介紹。

第三章 轉換規則，此章說明本方法提出定義之時間圖規格、物件導向時間派翠網路規則及轉換規則。

第四章 範例及應用，此章以前章所敘述之方式，使用範例表達轉換程序，並進行以可達性分析為應用範例。

第五章 結論及未來方向，此章將本論文之研究成果做總結，並提出未來研究之方向。

第 2 章 背景知識及相關研究

2.1 時間圖 (Timing Diagram)

OMG(Object Management Group)在 2003 年公佈的 UML 2.0 版本中[5][6]，新增了時間圖(Timing diagram)此一新的圖形，時間圖是用來表示物件的交互關係與時間的影響，並對物件中各別的狀態及物件間彼此的互動進行描述。

然而時間圖為 UML 2.0 版本所新增之互動圖(Interaction diagram)之一部分，互動圖描述物件之間各種互動、訊號及訊息的傳達、變數及行為邏輯的描述，而時間圖特別針對物件間互動與時間性質的影響部分做較細部的說明。

以下幾種圖形標記出自於 UML 2.0 Superstructure[5]中的定義：

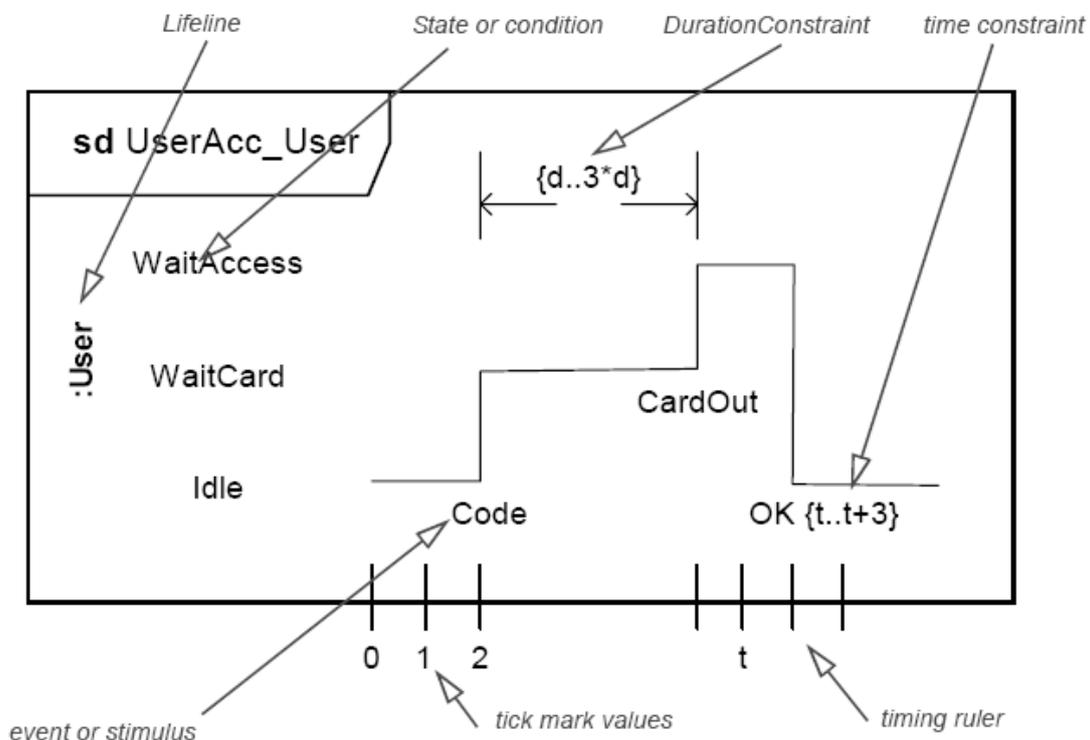
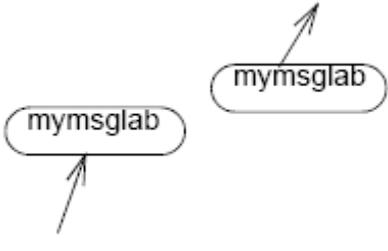
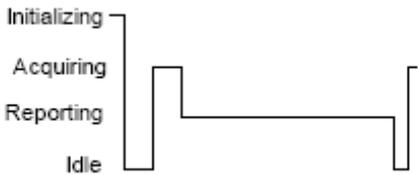
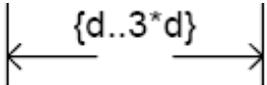
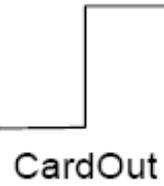
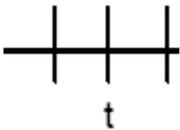
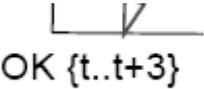


圖 1 時間圖圖示範例[5]

以下是對時間圖各項標記的基本介紹：

表 1 時間圖之各項標記

| 類型 | 標記 | 意義 |
|--------------|---|---|
| 生命線 Lifeline |  | <p>代表互動行為中的個別參與者，每個生命線代表的是一個單一的互動實體。可視為物件之實例 (Instance)。</p> |
| 框架 Frame |  | <p>此標記為互動圖中之框架，其中描述互動之行為，而左上角框線內容即為互動名稱，但在本論文之中主要探討之時間圖本身就繪於此框架中，因此本論文並沒有使用到此種標記。</p> |
| 訊息 Message |  | <p>訊息標記能代表各種不同類型的訊息，在左圖中是分別代表非同步訊息、呼叫以及回傳。</p> |

| | | |
|-----------------------------------|---|---|
| <p>訊息標記 Message Label</p> |  | <p>訊息標記是在當訊息數目眾多時，為了避免訊息符號交叉難以辨識，因此以此種標記方式紀錄訊息的來源及目的位置。</p> |
| <p>狀態 State</p> |  | <p>代表生命線之中的各種狀態。但在UML 2.0 的定義中允許以連續曲線（如溫度、濕度）的方式來表達。</p> |
| <p>持續時間限制 Duration Constraint</p> |  | <p>用來代表在一個狀態中維持的持續時間。</p> |
| <p>事件 Event</p> |  | <p>事件表示在特定時間點時，某些動作 (Action) 將對生命線狀態進行改變。</p> |
| <p>時間標記 Time Ruler</p> |  | <p>代表時間刻度。</p> |
| <p>時間限制 Time Constraint</p> |  | <p>限制事件處理的時間。</p> |

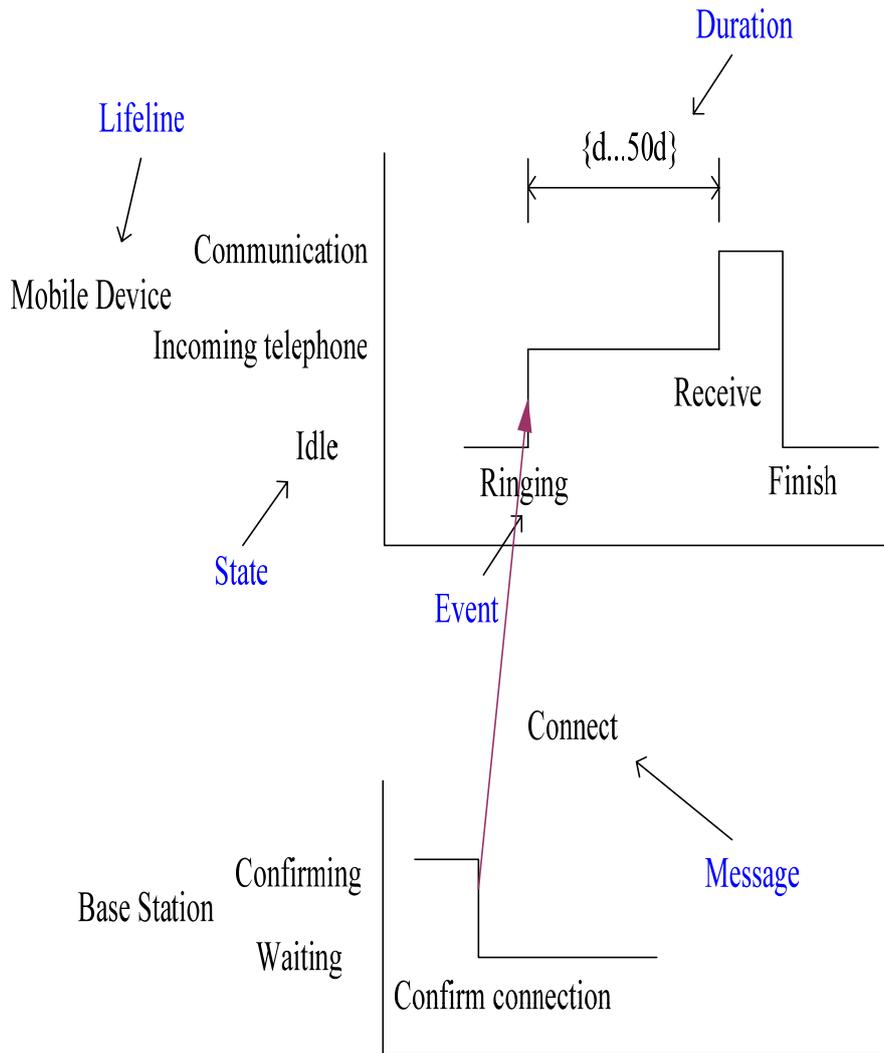


圖 2 時間圖範例

圖 2 範例為一行動通訊裝置與基地台之互動，當基地台收到有人撥打的訊號並確定連線後，基地台之狀態(State)由確認中轉變為待機，然而基地台發送訊息(Message)給行動通訊裝置時，就會引發來電的事件(Event)，發出響鈴。

關於時間圖詳細定義及應用可參考[5][6]，然而 UML 本身就是屬於物件導向之塑模語言，因此在本文中將時間圖中轉換至物件導向派翠網路以利用派翠網路的特性進行分析。

2.2 派翠網路 (Petri Nets)

派翠網路是由 C. A. Petri 在 1962 年所提出[21]，原本只用來探討電腦系統內部元件的通訊理論，但是後來發現其亦可以用於離散事件動態系統 (Discrete Event Dynamic System)。主要是因為此觀念不僅能表示系統的同步與互斥行為，而且可以對系統做定性與定量的分析。與其他用來描述系統動態行為的建模工具相互比較，像是狀態轉移圖、事件流程圖和劇本分析圖等只能依序描述系統的動態行為，派翠網路對於真實世界中許多同步進行的工作，可以做更進一步的描述，表 2 即為派翠網路與一般常見圖形表示法之比較[25]。

派翠網路為一種以抽象、正規的圖形表示，並且具有數學理論基礎的系統塑模、分析與發展的工具，如圖 3 所示。派翠網路可用來設計同步、非同步、決策或是分散等系統模型[11]，亦可分析及驗證系統行為，因此在各領域中經常被使用[12][18]。隨著處理系統複雜度、多變性增加，許多不同類型、特性的派翠網路理論也陸續被提出及研究，如時間派翠網路、物件導向派翠網路、模糊派翠網路、彩色派翠網路等，以處理各不同領域的問題[7][9][12][14][15]。

表 2 派翠網路與其他常用工具之比較

| | 資料流程圖 | 事件狀態圖 | 派翠網 |
|--------|--------|-------|-------|
| 輸入輸出 | 單一輸入輸出 | 多輸出輸入 | 多輸出輸入 |
| 決策表達能力 | 無 | 無 | 有 |
| 迴圈表達能力 | 無 | 無 | 有 |
| 程序控制 | 無 | 有 | 有 |
| 同步表示能力 | 無 | 無 | 有 |

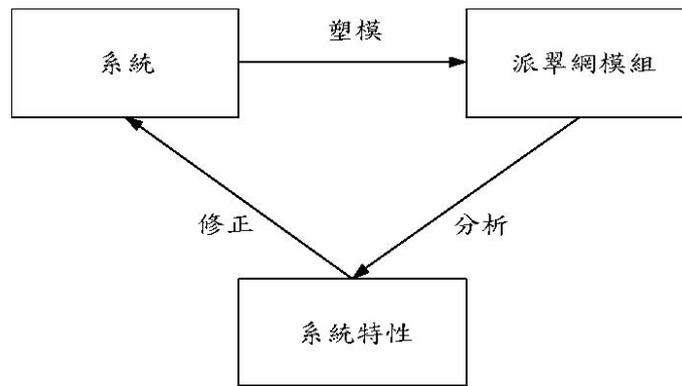


圖3 派翠網路可用來塑模、分析、修正系統

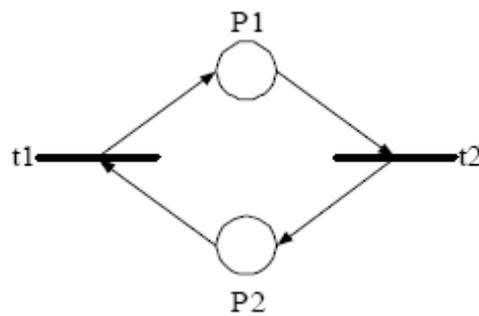


圖4 派翠網路的靜態基本圖形[19]

派翠網路的圖形表示法包含三種圖形元件：位置(Place)、轉置(Transition)以及有向弧(Directed Arc)。其中位置是以一個圓圈來表示，轉換的表示法則為橫線或長方形，而兩者之間由有方向性的弧線相連。此三者構成了類似於流程圖的派翠網路靜態圖形。當派翠網路應用在不同的領域時，會使派翠網路中的位置與轉換跟著有不同的意義[18]，表3列舉出一些派翠網路中常見位置與轉換符號的解釋。

表3 派翠網路中位置與轉換的解釋[1]

| 輸入位置 | 轉換 | 輸出位置 |
|------|-------|------|
| 前序條件 | 事件 | 後序條件 |
| 輸入資料 | 計算步驟 | 輸出資料 |
| 輸入信號 | 信號處理器 | 輸出信號 |
| 要求資源 | 工作 | 釋放資源 |
| 條件 | 邏輯子句 | 結論 |
| 緩衝器 | 處理器 | 緩衝器 |

在派翠網路動態的描述方面，則是需要再加上標記（Token）的運用，透過標記的移動來完成系統動態的描述。也就是說，當一個轉換的每一個輸入位置均至少含有一個以上的標記時，就表示該轉換有資格被激發（Fire），一旦轉換被激發之後，其輸入位置的標記將會移動到輸出位置。

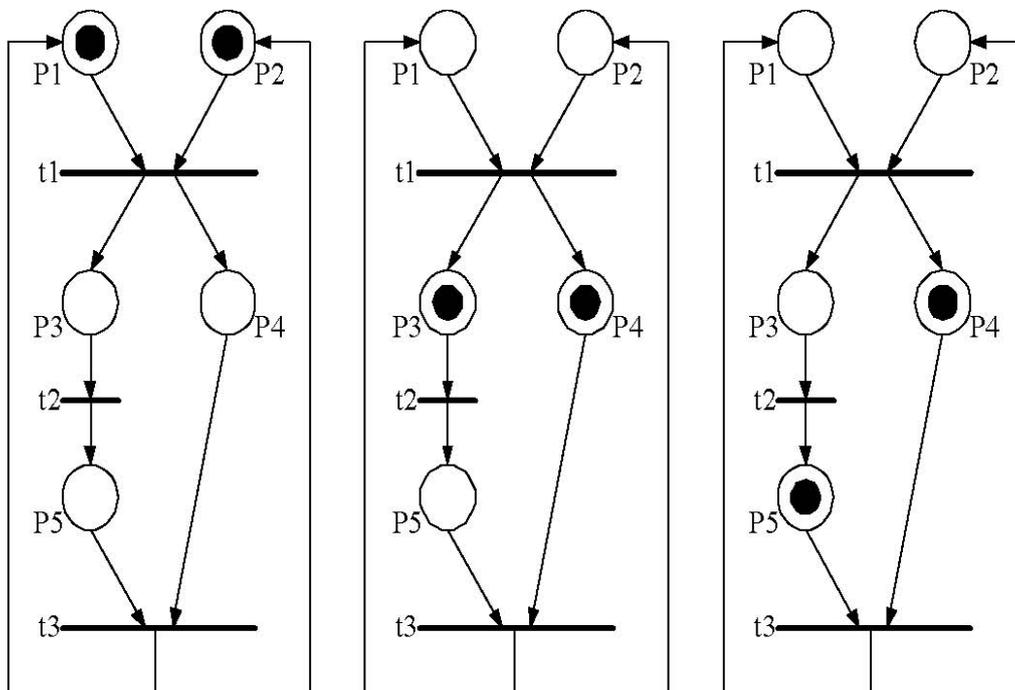
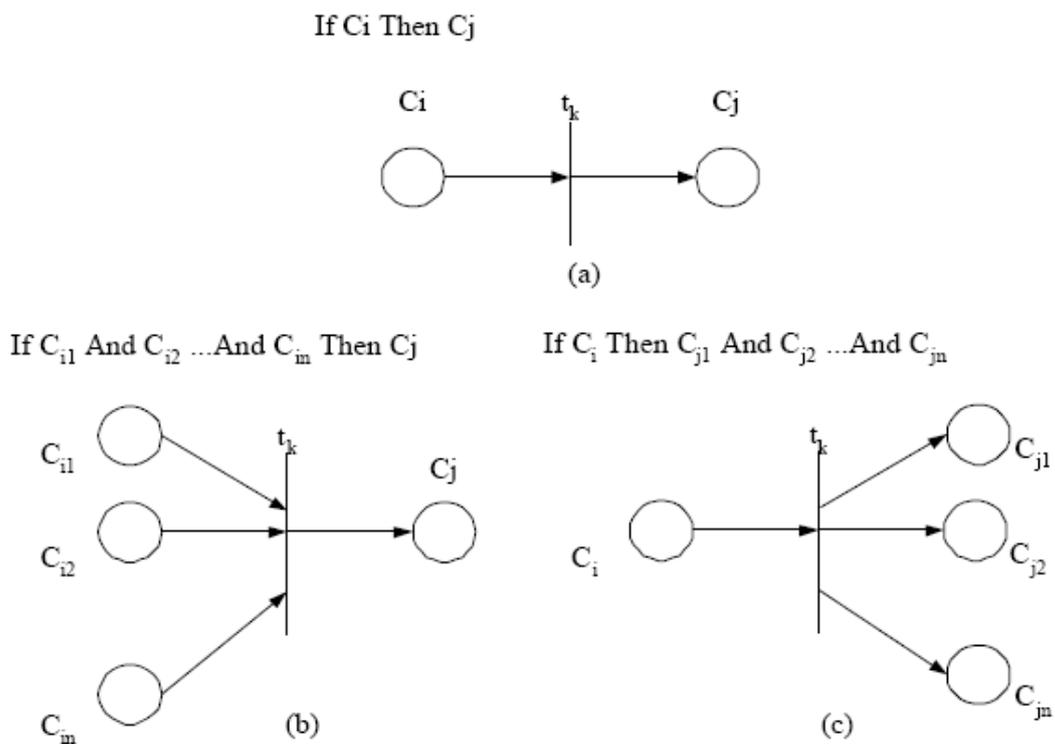


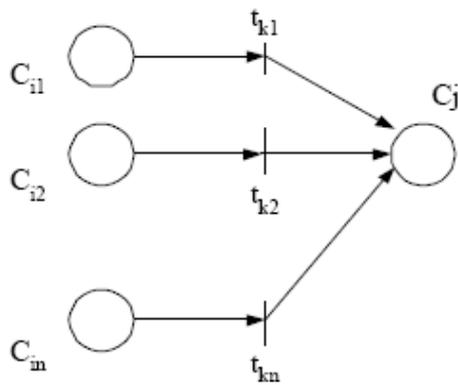
圖5 派翠網路的動態表示[7]

假設今有一派翠網路如圖5，其中的實心圓表示標記。當P1與P2 都有標記時，t1就有資格被激發，一旦t1被激發之後，P1與P2的標記就移到t1的輸出位置，即為P3與P4，如圖5 (b) 所示；同理，當P3有標記時，t2就有資格被激發，當t2被激發之後，P3的標記就移到t2的輸出位置，即為P5，如圖5 (c) 所示；當P4與P5都有標記時，t3可以被激發，並且將P4與P5 的標記移到t3的輸出位置，也就是P1與P2，因此又回到了圖5 (a)。整個系統由於轉換的激發帶動了標記的流動，因而產生了動態行為。

派翠網路可以用來塑模諸如“If...Then ”規則之類的系統，其中轉換的激發就代表著相對應法則的啟動，而輸入位置代表產生法則的前提，輸出位置代表法則的結論。因此針對不同的邏輯術語，可將之歸類為五種基本類型之派翠網路模組（如圖6所示），利用此一模組及其組合，便可用來塑模特定領域的系統。

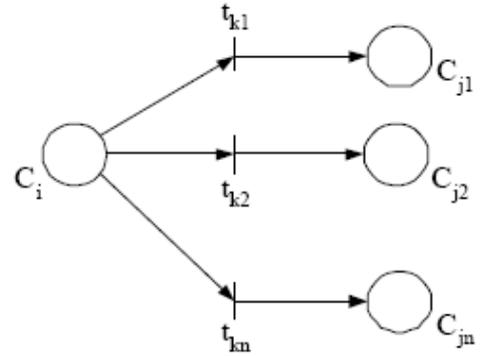


If C_{i1} Or C_{i2} ... Or C_{in} Then C_j



(d)

If C_i Then C_{j1} Or C_{j2} ... Or C_{jn}



(e)

圖6 五種派翠網路塑模之類型

派翠網路除了可以用來塑模系統之外，更可以進一步分析系統的特性，以幫助設計者在實行系統之前，預先得知系統的屬性，以便於做進一步的修改。以下有幾種比較常見的特性可以透過派翠網進行分析：

1. 限制性：限制性是指在派翠網中，每個位置內標記的數量限制。假設一個派翠網路被稱為K Bounded，就代表在這個派翠網中，每一個位置內，標記的數量最多不得超過K 個。
2. 安全性：當派翠網路中，每個位置內的標記會維持在一定數量之下，而不會無限的擴增，就稱此系統有安全性。
3. 鎖死與活性：鎖死現象是指系統到一個狀態之後就無法到達其他狀態。反之，一個系統若無此現象發生就稱此系統為活性的。

在派翠網路中有兩種比較典型的分析方法，可以用來分析上述性質：（1）可達樹（Reachability Tree）；（2）矩陣方程式（Matrix Equation）。其中可達樹可以詳細地追蹤派翠網的所有狀態，但是當系統過大時，其圖形將變成相當複雜而不容易判斷，因此其只適用於較小的系統。而矩陣方程式則是可以透過數學

矩陣的計算，推理出派翠網路的行為。

可達樹分析首先要建立如同樹狀結構的派翠網，其中每一個節點 (Node) 表示此時的狀態，上一層的節點經過轉換的激發後就可以到達下一個節點。狀態的表示法與轉換的順序如圖7 所示。利用可達樹的圖形可以用來幫助分析派翠網是否具有安全性、有限性或是活性 (Liveness) 等系統行為。

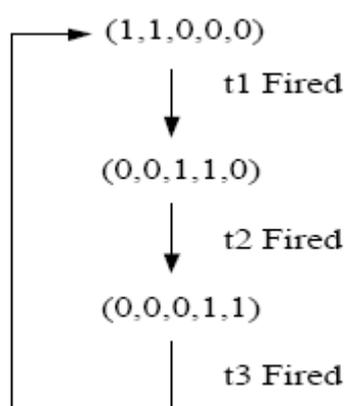


圖7 可達樹

例如圖7是根據圖5所描繪出來的可達樹，其初始狀態為(1,1,0,0,0)，分別代表(P1,P2,P3,P4,P5)是否有標記，若是有標記則以1表示，若是沒有則以0表示。因此在圖3-3中，當位置P1與P2各有一個標記，其他的位置則沒有任何標記，就表示成(1,1,0,0,0)。經過t1的激發之後，使得系統狀態變成P3與P4各有一個標記，以(0,0,1,1,0)表示。接著激發t2，來到狀態(0,0,0,1,1)。最後，當t3激發之後，系統狀態回覆到初始狀態(1,1,0,0,0)。因此透過可達樹的分析，可以確認在整個過程中並沒有發生死結與衝突。

2.3 物件導向派翠網路 (Object Oriented Petri Nets)

Sibertin-Blane和Bastide在1990年正式將物件導向觀念發展為物件導向派翠網路(Object-Oriented Petri Nets, OOPN)[17]，之後在1993年時Lee及Park將其概念發展得更完全[14]，在其研究中應用物件導向物件及訊息傳遞的觀點，而發展出物件導向高階派翠網路(Object-Oriented High-Level Petri Nets, OPNets)，以增加派翠網路的模組化、易於維護以及重覆使用等特性。

物件導向是由一群經過抽象化的物件所組成。其中一個物件是指一個實體 (Entity)，而一個實體又由兩部分組成：一是屬性，用來描述物件本身的特徵；另一是行為 (或稱為方法)，當物件接收到外來的訊息後，會對物件的屬性進行處理，並做出適當地回應。每一個物件都是唯一的，但是當物件與物件之間擁有相似的屬性和行為時，則可以透過抽象化 (Abstraction) 來剔除差異的部分，並集合這些物件相同的部分以形成一個類別 (Class)。而在同一類別中，每一個物件則稱做是這個類別中的實例 (Instance)。

由於物件是系統中的實體經過直接對應 (Direct Map) 而產生，不僅能以更接近真實系統的運作方式來描述系統的行為外，也可以達到高度的模組化，主要是因為物件導向具有以下幾點特性：

1. 封裝與資訊隱藏：一物件的行為，可以透過介面來隱藏內部行為，進而使得物件得以獨立的運作，這種使各物件的相互依存性達到最小的方法就稱為封裝或資訊隱藏。

2. 繼承：藉由類別的觀念，使得子類別內的個體可以繼承其父類別的屬性與行為，如此可以避免程式碼的重複製作。

3. 訊息溝通：一個物件若要求另一個物件產生行為，則發送端的物件就必須傳遞訊息給接收端的物件，訊息中必須包含接收端物件的名稱、欲執行的行為的名稱以及一些參數，好讓接收端的物件在收到訊息之後，可以清楚地完成發送端交付的動作或要求。

物件導向在軟體科技上帶來的最大遠景是高度的軟體再用（Software Reuse）。因為物件導向是將物件相同的部分抽象化成為類別，而繼承的觀念不僅允許自動繼承父系類別的屬性與方法，更可以額外加上自己的屬性與方法。基於此兩種符合再使用的相同與除外原則特型，因此可以達到軟體再用的目的。

在物件導向中，利用訊息傳遞的形式可以達到系統的同步化。進而藉著物件導向的觀念將整個系統分解成數個小系統，除了降低系統的複雜度之外，亦可以達到高度的模組化。由於模組化可以將系統的功能細化（Refinement），將整個系統分解為小問題，然後再集合小問題的答案讓系統的問題迎刃而解，因此在設計上有相當好的成效。不過，物件導向雖然有以上的優點，卻在發展與評估系統的同步行為上，缺少系統的分析與驗證方法。而派翠網路因為在平行系統的模型設計、分析與驗證上都有很好的定義，正好補足了物件導向的不足。因此將兩者結合，其優點可分述如下：

1. 有鑑於再使用與共享的重要性，可將派翠網路加入物件導向的觀念。
2. 抽象化資料形態、封包、以及資訊隱藏的觀念增進軟體的可靠度與模組化結構。
3. 容易擴增以及變更其表示法，以滿足變化的需求規格。
4. 多態性（Polymorphism）可以建立通用的軟體元件或使用已經存在的程式碼來建立新的類別，增強了軟體發展的彈性。

5. 繼承性 (Inheritance) 可以重複使用已發展完成的元件，同時提高元件的擴充性 (Extensibility)。
6. 完整的系統設計與分析

在物件導向派翠網路中，物件是被限制在框內以表示抽象化及封裝，一個基本的物件導向派翠網路包含[2][7][10]：

- 1)位置(Places)：用來代表物件的特定狀態。
- 2)訊息位置(Message Places)：物件與外界物件間溝通的窗口。
- 3)轉置(Transitions)：描述位置間的轉換事件。
- 4)閘道(Gateways)：類似轉置，處理物件間的連結關係。

圖8為一個簡單的系統，在這個系統中包括兩物件 O_1 、 O_2 ，其中 O_2 內還包括兩個物件 O_{21} 、 O_{22} ，故 O_2 稱為組合的物件，而物件 O_{21} 、 O_{22} 稱為私有的物件，因為在這物件裡沒有其他的物件。此外， g_1 、 g_2 、 g_{21} 、 g_{22} 、 g_{23} 、 g_{24} 以粗黑線表示為閘道 (Gate)，專司物件之間的聯絡，而圖中圓圈的部分則代表為訊息位置，專門負責接受與傳送外在物件的訊息。因此利用物件導向的觀念，可以很容易的把資料隱藏。

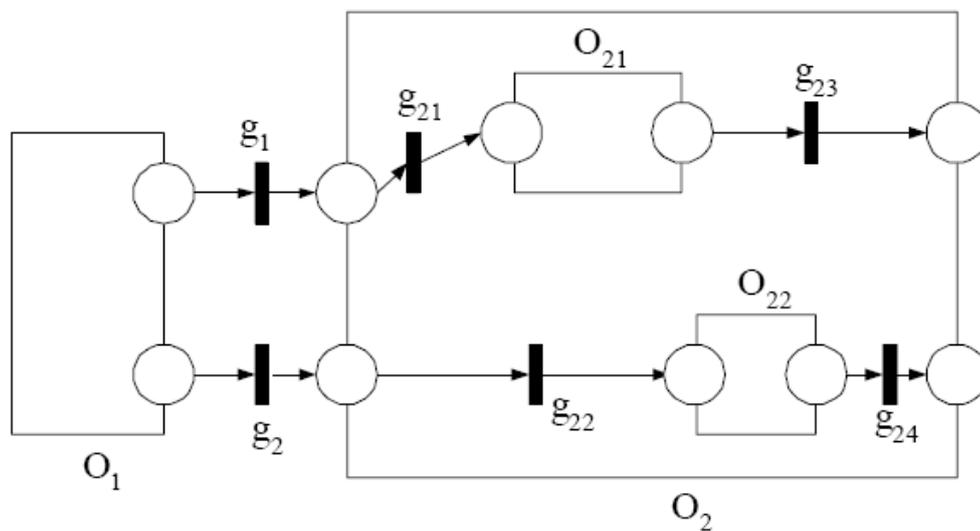


圖 8 物件導向派翠網路範例

從物件導向的觀點來看，物件的內部行為是可以封裝在其中的。因此在進行系統行為分析時，可以將內部行為忽略，只針對訊息的介面做探討，以減少系統分析上的複雜度。圖9是一個組合物件A的物件導向派翠網路圖，其中還包含兩個私有物件AA與AB（圖10）。

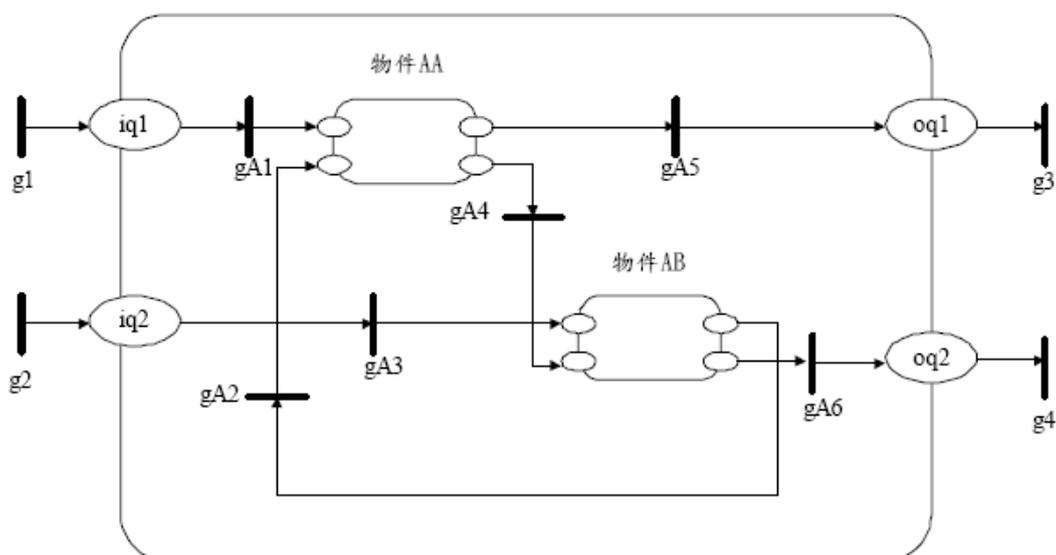


圖 9 包含物件AA與物件AB之物件導向派翠網路

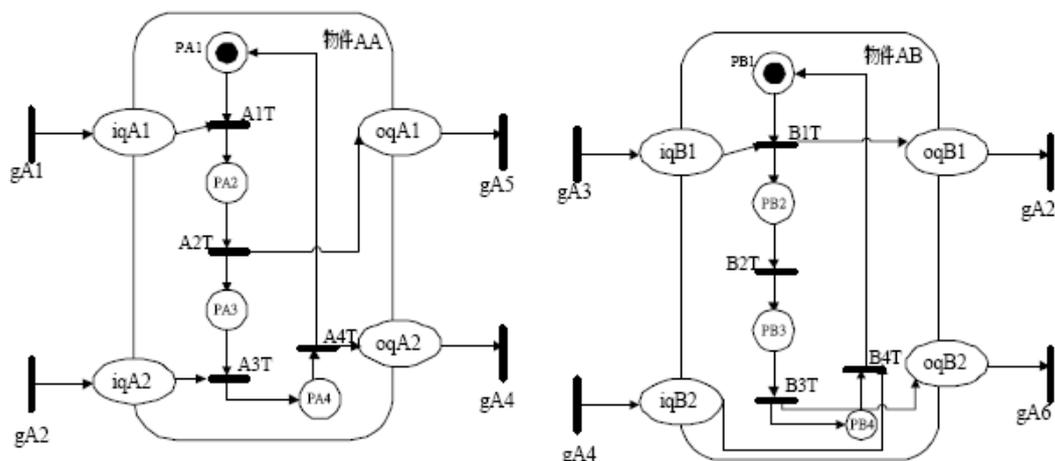


圖10 私有物件AA與AB的物件導向派翠網路

根據圖10，其初始狀態為 $(1,0,0,0,0,0,0,0)$ ，分別代表 $(PA1,PA2,PA3,PA4,iqA1,iqA2,oqA1,oqA2)$ 具有標記與否，或者直接以 $(PA1)$ 表示此時只有PA1 有標記，可以讓表示法更為簡單、清晰易懂。當gA1 被激發之後，標記會流入 iqA1，此時A1T 的輸入位置PA1 與iqA1 都具備標記了，A1T 才有資格被激發。A1T 激發之後，會將PA1 與 iqA1 的標記流入A1T 的輸出位置，也就是PA2。同理，當PA2 有標記時，A2T 可以被激發，並且將標記流入 PA3 與oqA1，須注意此時只有gA5 可以被激發，而A3T 還不能被激發，因為A3T 還有一個輸入位置iqA2 尚未擁有標記。一旦gA2被激發之後，iqA2 就可以擁有標記，而此時A3T 也可以順利被激發，並將標記流入到PA4。PA4 有標記之後，就可以激發A4T，並將標記流入oqA2 與PA1，此時，又只剩gA4 可以激發。因此根據以上的描述，可以確認出其活動轉換的激發順序為：

$gA1 \Rightarrow A1T \Rightarrow A2T \Rightarrow gA5 \Rightarrow gA2 \Rightarrow A3T \Rightarrow A4T \Rightarrow gA4$

同樣的方法，可以確認出物件AB 內部的活動轉換激發順序為：

$gA3 \Rightarrow B1T \Rightarrow gA2 \Rightarrow B2T \Rightarrow B3T \Rightarrow gA6 \Rightarrow gA4 \Rightarrow B4T$

2.4 時間派翠網路 (Time Petri Nets)

時間派翠網路是常被使用在即時系統的規格敘述與驗證上的一種方法，時間派翠網路(Time Petri Net)[20][24]是在原本的派翠網路之外做時間性的擴充，在每個轉置增加兩個時間參數做為時間區間(Time Interval)的代表，以下是時間派翠網路之定義：

一個時間派翠網路由六個欄位(P,T,B,F,M₀,I)組成[9]，其中：

- 1) $P = \{P_1, P_2, \dots, P_m\}$ 為一位置的有限非空集合。
- 2) $T = \{t_1, t_2, \dots, t_n\}$ 為一轉置的有限非空集合。
- 3) $B : P \times T \rightarrow N$ 為反向關聯函數，為紀錄當某一轉置被激發時，標記從位置到轉置的改變。
- 4) $F : T \times P \rightarrow N$ 為正向關聯函數，為紀錄當某一轉置被激發時，標記從轉置到位置的改變。
- 5) M_0 為初始標記(Initial Marking)，記錄派翠網路的初始狀態。

(以上五項欄位可定義一基本之派翠網路。)

- 6) I 為區間(Interval)。 $\forall t \in T, I(t) = [EFT(t), LFT(t)]$ ，其中 EFT 為最早激發時間(Earliest Firing Time)而 LFT 為最晚激發時間(Latest Firing Time)。

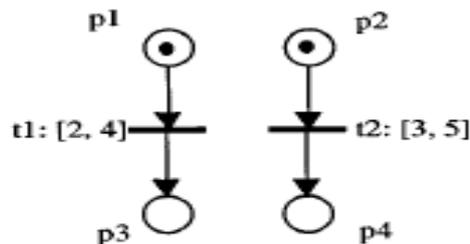


圖 11 時間派翠網路範例

以圖 11 為例，在圖 11 中時間派翠網路的起始狀態為P₁和P₂位置已有標記存在，故M₀=(1 1 0 0)，然而即使P₁及P₂位置已經有標記的存在，但在t₁及t₂要被激發還分別需要在[2,4]及[3,5]的時間區隔之中進行，當t₁或t₂轉置被激發後會分別引起B(t₁),F(t₁)及B(t₂),F(t₂)二組函數之改變，造成標記所在位於的，同時也以改變此派翠網路之狀態(Marking)。

以下是圖 11 中各項欄位之資料：

$$P = \{P_1, P_2, P_3, P_4\}$$

$$T = \{t_1, t_2\}$$

$$M_0 = (1 \ 1 \ 0 \ 0)$$

$$B(t_1) = (1 \ 0 \ 0 \ 0)$$

$$B(t_2) = (0 \ 1 \ 0 \ 0)$$

$$F(t_1) = (0 \ 0 \ 1 \ 0)$$

$$F(t_2) = (0 \ 0 \ 0 \ 1)$$

$$I(t_1) = [EFT(t_1), LFT(t_1)]$$

$$= [2, 4]$$

$$I(t_2) = [EFT(t_2), LFT(t_2)]$$

$$= [3, 5]$$

第 3 章 轉換規則

3.1 本論文之方法論流程

下圖是本論文提出之方法架構：

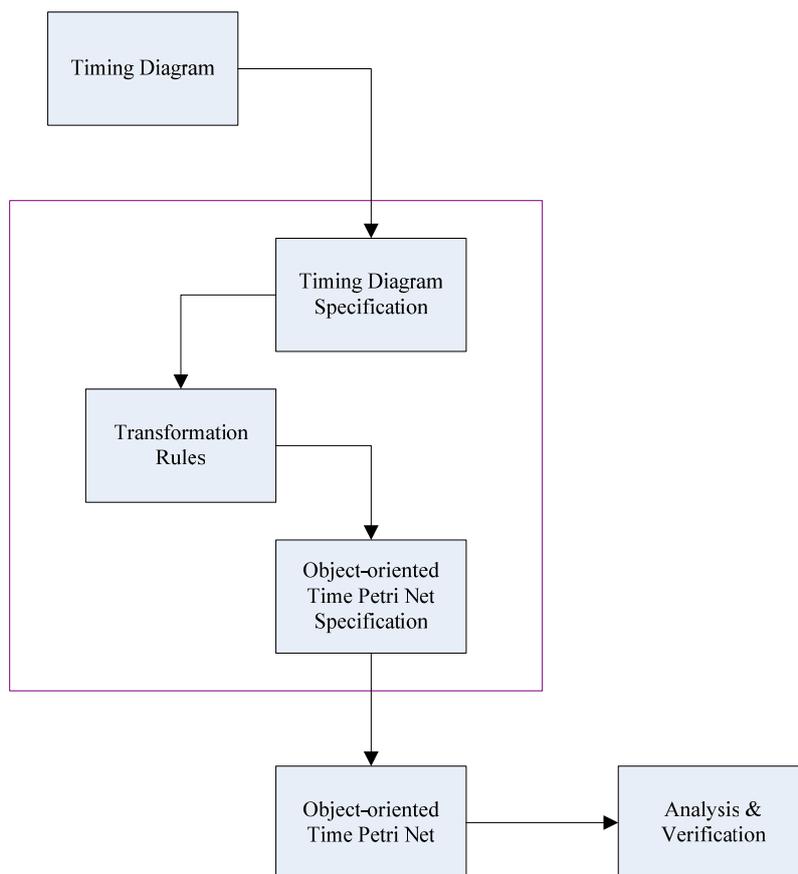


圖 12 本論文提出之方法架構

本論文提出之方法程序如下：

1. 開發人員繪製時間圖。
2. 將時間圖轉換為時間圖規格(Timing Diagram Specification)
(3.2)。
3. 依據本文提出之轉換規則，將時間圖規格轉換為物件導向時間派翠網路規格(Object-oriented Time Petri Nets Specification)(3.3)。

4. 以物件導向時間派翠網路規格為依據，將物件導向派翠網路具體呈現。
5. 利用派翠網路的相關特性及應用方法，進行分析與驗證。

以下各章節將對上述提及各部分做說明。

3.2 時間圖規格 (Timing Diagram Specification)

時間圖規格(Timing Diagram, TDS)是由三項欄位基本元素組成， $TDS = (TU, L, M)$ ，其中：

1. TU(Time Unit)為時間單位。
2. $L = \{l_1, l_2, \dots, l_{|L|}\}$ 為時間圖中各生命線(lifeline)之集合。
3. $M = \{m_1, m_2, \dots, m_{|M|}\}$ 為訊息(Message)之集合。

又可細分為：

- (1) 生命線包含五個部分 $l_i = (N_i, S_i, IS_i, E_i, DC_i)$ ，其中：
 - 1) N_i 是生命線之名稱(name of lifeline)。
 - 2) $S_i = \{s_{S_{i_1}}, s_{S_{i_2}}, \dots, s_{S_{i_{|S_i|}}}\}$ 為生命線中狀態之集合(set of states of the lifeline)。
 - 3) IS_i 為生命線之起始狀態(the initial state of the lifeline)， $IS_i \in S_i$ 。
 - 4) $E_i = \{e_{E_{i_1}}, \dots, e_{E_{i_{|E_i|}}}\}$ 為事件之集合(set of the events of the lifeline)。
 - 5) $DC_i = \{dc_{DC_{i_1}}, \dots, dc_{DC_{i_{|DC_i|}}}\}$ 為持續時間(duration time constraint)。

- 6) $e_i = (N_{e_i}, ss_{e_i}, ts_{e_i}, st_{e_i}, et_{e_i})$ 為 E 中之事件(event)，其中：
- 甲、 N_{e_i} 為事件名稱(name of the event)。
 - 乙、 ss_{e_i} 為事件之起始狀態(staring state of the event)。
 - 丙、 ts_{e_i} 為事件之終止狀態(terminal state of the event)。
 - 丁、 st_{e_i} 是事件之最少所需下限時間(the minimum execution time of the event)。
 - 戊、 et_{e_i} 為事件執行所需之最長上限時間(the maximum execution time of the event)。
 - 己、事件 e_i 觸發狀態改變由啟始狀態 ss_{e_i} 轉變為終止狀態 ts_{e_i} ，其中 $1 \leq i \leq |E|$ 。
- 7) l_i 中之持續時間 dc_i 由二部分 (st_{dc_i}, et_{dc_i}) 組成， st_{dc_i} 為最少所需下限時間 dc_i ， et_{dc_i} 為上限時間 dc_i 。 $1 \leq i \leq |DC|$
- 8) l_i 的實體(instance) 以 $s_i(0, t_1) e_i(t_1, t_2) s_j(t_2, t_3) e_j(t_3, t_4) s_k(t_4, t_5) \dots$ 的方式表示， $s_i, s_j, s_k \in S_i$ ，且 $e_i, e_j \in E_i$
- 9) 訊息 $m_i = (N_{m_i}, sl_{m_i}, tl_{m_i}, e_{sl}, e_{tl}, st_{m_i}, tt_{m_i})$ ，其中：
- 甲、 N_{m_i} 為訊息之名稱。
 - 乙、 sl_{m_i} 為訊息起始之生命線。
 - 丙、 tl_{m_i} 為訊息終止之生命線。
 - 丁、 e_{sl} 為引發訊息之事件。
 - 戊、 e_{tl} 為訊息將觸發之事件。

己、 st_{m_i} 為訊息最少花費之時間。

庚、 tt_{m_i} 為訊息之最長花費時間。

事件 e_{sl} 的發生導致訊息 m_i 從起始的生命線 sl_{m_i} 發生被送到終止之生命線 tl_{m_i} 並觸發事件 e_{tl} 。

表 4 以表格方式列出 TDS 之各項欄位。

表 4 時間圖規格

| TDS (Time Diagram Specification) | | | |
|----------------------------------|-----------|-----------|-------------|
| TU | 時間單位 | | |
| L 生命線 | N_i | 生命線之名稱 | |
| | S_i | 生命線中狀態之集合 | |
| | IS_i | 生命線之啟始狀態 | |
| | DC_i | 持續時間 | |
| | E_i | 事件之集合 | |
| | e_i | N_{ei} | 事件名稱 |
| | | SS_{ei} | 事件之起始狀態 |
| | | tS_{ei} | 事件之終止狀態 |
| | | st_{ei} | 事件之最少所需下限時間 |
| | | $tSei$ | 事件之最長上限時間 |
| M 訊息 | N_{mi} | 訊息之名稱 | |
| | sl_{mi} | 訊息起始之生命線 | |
| | tl_{mi} | 訊息終止之生命線 | |
| | e_{sl} | 引發訊息之事件 | |
| | et_l | 訊息將觸發之事件 | |
| | st_{mi} | 訊息最少花費之時間 | |
| | tt_{mi} | 訊息最長花費之時間 | |

3.3 物件導向時間派翠網路規格 (Object-oriented time Petri Net Specification)

物件導向時間派翠網路規格 OOTPNS 由九個欄位元素 (P, T, B, F, M₀, I, OP, G, C) 組成，其中：

- (1) $P = \{p_1, p_2, \dots, p_{|P|}\}$ 為位置之集合。
- (2) $T = \{t_1, t_2, \dots, t_{|T|}\}$ 為轉置之集合，每轉置代表動作、功能。
- (3) $B : P \times T \rightarrow N$ 為反向關聯函數。
- (4) $F : T \times P \rightarrow N$ 為正向關聯函數。
- (5) M_0 為初始標記(Initial Marking)。
- (6) I 為時間區間(Interval)。 $\forall t \in T, I(t) = [EFT(t), LFT(t)]$ ，其中 EFT 為最早激發時間(Earliest Firing Time)而 LFT 為最晚激發時間(Latest Firing Time)。
- (7) $OP = \{OP_1, OP_2, \dots, OP_{|OP|}\}$ 為訊息位置之集合。
- (8) $G = \{G_1, G_2, \dots, G_{|G|}\}$ 為閘道之集合。
- (9) $C = \{c_1, c_2, \dots, c_{|C|}\}$ 為物件類別之集合，每個類別中包括其屬性（在本文中對應為位置）及方法（在本文中對應為轉置）。

表 5 以表格方式列出 OOTPNS 各項欄位。

表 5 物件導向時間派翠網路規格

| OOTPNS (Object Oriented Time Petri Net Specification) | |
|---|---------|
| P | 位置之集合 |
| T | 轉置之集合 |
| B | 反向關聯函數 |
| F | 正向關聯函數 |
| M_0 | 初始標記 |
| I | 時間區隔 |
| OP | 訊息位置之集合 |
| G | 閘道之集合 |
| C | 物件類別之集合 |

3.4 轉換規則 (Transformation Rules)

令時間圖規格 $TDS = (TU, L, M)$ 、物件導向時間派翠網路規格 $OOTPNS = (P, T, C, V)$ 。

Step1: 將 TDS 中的各項生命線(lifeline)轉換成物件導向時間派翠網路中各位置及事件。

將

TDS 中的 $L = \{l_1, l_2, \dots, l_{|L|}\}$ 轉換為 OOTPN 中之 $C = \{c_1, c_2, \dots, c_{|L|}\}$ ，

令 $l_1 = c_1, l_2 = c_2, \dots, l_{|L|} = c_{|L|}$

而 l_i 其的狀態 $S_i = \{s_{i_1}, s_{i_2}, \dots, s_{i_{|S_i|}}\}$ 轉換為 c_i 為 $P_i = \{p_{i_1}, \dots, p_{i_{|P_i|}}\}$ ，

令 $s_1 = p_1, s_2 = p_2, \dots, s_{|i|} = p_{|i|}$

l_i 中之 IS_i 轉換為 M_0 ，

令 $IS_i = M_0$

l_i 中之 $E_i = \{e_{i_1}, e_{i_2}, \dots, e_{i_{|E_i|}}\}$ 轉換為 $T = \{T_1, T_2, \dots, T_i\}$ ，

此步驟以轉換 TDS 中各生命線之狀態至 OOTPNS 中之各轉置為主。

接下來我們以表格 6,7 呈現 Step 1 之對應，以方便參照。

表 6 Step1 之時間規格示意表

(星號為表示對應。)

| TDS (Time Diagram Specification) | | | | |
|----------------------------------|------------------|------------------|-------------|--|
| TU | 時間單位 | | | |
| L 生命線 | *N _i | *生命線之名稱 | | |
| | *S _i | *生命線中狀態之集合 | | |
| | *IS _i | *生命線之啟始狀態 | | |
| | *DC _i | *持續時間 | | |
| | E _i | 事件之集合 | | |
| | e _i | N _{ei} | 事件名稱 | |
| | | SS _{ei} | 事件之起始狀態 | |
| | | tS _{ei} | 事件之終止狀態 | |
| | | st _{ei} | 事件之最少所需下限時間 | |
| | | tSei | 事件之最長上限時間 | |
| M 訊息 | N _{mi} | 訊息之名稱 | | |
| | sl _{mi} | 訊息起始之生命線 | | |
| | tl _{mi} | 訊息終止之生命線 | | |
| | e _{sl} | 引發訊息之事件 | | |
| | et _i | 訊息將觸發之事件 | | |
| | st _{mi} | 訊息最少花費之時間 | | |
| | tt _{mi} | 訊息最長花費之時間 | | |

表 7 Step1 之物件導向時間派翠網路規格示意表

(星號為表示對應。)

| OOTPNS (Object Oriented Time Petri Net Specification) | |
|---|---------|
| *P | *位置之集合 |
| T | 轉置之集合 |
| B | 反向關聯函數 |
| F | 正向關聯函數 |
| *M ₀ | *初始標記 |
| I | 時間區隔 |
| OP | 訊息位置之集合 |
| G | 閘道之集合 |
| C | 物件類別之集合 |

Step2: 將 E_i 中的各項事件(event)及其相關欄位轉換成物件導向時間派翠網路對應之屬性。

令 $e_1 = I(t_1), e_2 = I(t_2), \dots, e_i = I(t_i)$

e_i 中的 st_{e_i} 等於 $I(i)$ 中之 $EFT(t_i)$, et_{e_i} 等於 $LFT(t_i)$ 。

即物件導向時間派翠網路中之時間間隔(Time Interval)。

e_i 中之 ss_{e_i} 轉換為 B

令 $ss_{e_i} = B(t_i)$

e_i 中之 ts_{e_i} 轉換為 F

令 $ts_{e_i} = F(t_i)$

表 8,9 呈現 Step 2 之對應，以方便參照。

表 8 Step2 之時間規格示意表

(星號為表示對應。)

| TDS (Time Diagram Specification) | | | |
|----------------------------------|-----------|------------|-------------|
| TU | 時間單位 | | |
| L 生命線 | N_i | 生命線之名稱 | |
| | S_i | 生命線中狀態之集合 | |
| | IS_i | 生命線之啟始狀態 | |
| | DC_i | 持續時間 | |
| | $*E_i$ | 事件之集合 | |
| | $*e_i$ | $*N_{ei}$ | 事件名稱 |
| | | $*SS_{ei}$ | 事件之起始狀態 |
| | | $*t_{Sei}$ | 事件之終止狀態 |
| | | $*st_{ei}$ | 事件之最少所需下限時間 |
| | | $*t_{sei}$ | 事件之最長上限時間 |
| M 訊息 | N_{mi} | 訊息之名稱 | |
| | sl_{mi} | 訊息起始之生命線 | |
| | tl_{mi} | 訊息終止之生命線 | |
| | e_{sl} | 引發訊息之事件 | |
| | et_j | 訊息將觸發之事件 | |
| | st_{mi} | 訊息最少花費之時間 | |
| | tt_{mi} | 訊息最長花費之時間 | |

表 9 Step2 之物件導向時間派翠網路規格示意表

(星號為表示對應。)

| OOTPNS (Object Oriented Time Petri Net Specification) | |
|---|---------|
| P | 位置之集合 |
| *T | *轉置之集合 |
| *B | *反向關聯函數 |
| *F | *正向關聯函數 |
| M ₀ | 初始標記 |
| *I | *時間區隔 |
| OP | 訊息位置之集合 |
| G | 閘道之集合 |
| C | 物件類別之集合 |

Step3: 轉換 TDS 中之訊息至物件導向時間派翠網路中之閘道與訊息位置。

生命線 l_i 中之事件 e_i 引發訊息 m_i 從生命線 l_i 傳送至 l_j ，並引發 l_j 中之 e_j 事件，其中 i and $j \leq |L|$ 。

從 Step1 中，我們可知生命線 l_i 轉換成物件 c_i ， l_j 轉換為物件 c_j ， e_i 轉換為 t_i 及 e_j 轉換為 t_j 。

針對 TDS 中之 M 集中每個訊息 m_k ，按照以下的步驟轉換每個訊息：

- (1) 在物件 c_i 建立訊息位置 op_i ， $op_i \in OP$ 。
- (2) 在物件 c_j 建立訊息位置 op_j ， $op_j \in OP$ 。
- (3) 在訊息位置 pc_i 與訊息位置 pc_j 中建立閘道 G_i 代表訊息 m_k 。
- (4) 根據 m_k 中紀錄訊息起始之生命線 sl_{m_i} ，將 op_i 連結至 G_i (連結為有向性)。
- (5) 根據 m_k 中紀錄訊息終止之生命線 tl_{m_i} ，將 G_i 連結至 op_j (連結為有向性)。
- (6) 根據 m_k 中紀錄引發訊息之事件 e_{sl} ，將 e_i 連結至 op_i (連結為有向性)。
- (7) 根據 m_k 中紀錄訊息將觸發之事件 e_{tl} ，將 op_j 連結至 e_j (連結為有向性)。
- (8) m_k 中之 st_{m_i}, tt_{m_i} 轉換為時間區間。

$$\text{令 } st_{m_i} = \text{EFT}(G_i), \quad tt_{m_i} = \text{LFT}(G_i)$$

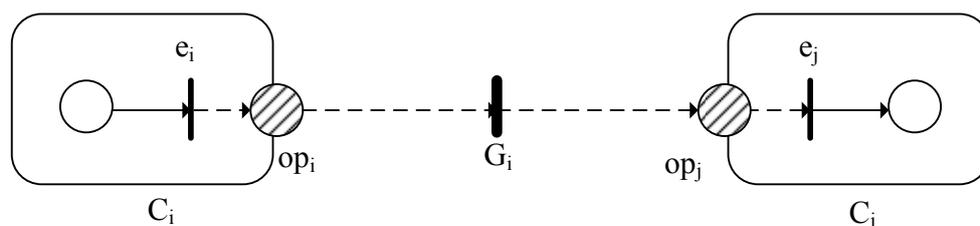


圖 13 Step3 之示意圖

圖 13 為 Step3 處理之結果，亦是本論文中處理訊息之方法，圖中之 op_i 、 G_i 、 op_j 都是因處理事件 e_i 所引發之訊息 m_k 所產生，因此圖中之 op_i 、 op_j 以實心表示為新增之訊息位置，而虛線部分表示是處理訊息部分所新增之連結。

表 10,11 呈現 Step 3 之對應，以方便參照。

表 Step3 之時間規格示意表

(星號為表示對應。)

| | |
|----------------------------------|------|
| TDS (Time Diagram Specification) | |
| TU | 時間單位 |

| | | | | |
|----------|-------------|-----------|-------------|--|
| L 生命線 | N_i | 生命線之名稱 | | |
| | S_i | 生命線中狀態之集合 | | |
| | IS_i | 生命線之啟始狀態 | | |
| | DC_i | 持續時間 | | |
| | E_i | 事件之集合 | | |
| | e_i | N_{ei} | 事件名稱 | |
| | | SS_{ei} | 事件之起始狀態 | |
| | | tS_{ei} | 事件之終止狀態 | |
| | | st_{ei} | 事件之最少所需下限時間 | |
| | | $tSei$ | 事件之最長上限時間 | |
| *M 訊息 | * N_{mi} | 訊息之名稱 | | |
| | * sl_{mi} | 訊息起始之生命線 | | |
| | * tl_{mi} | 訊息終止之生命線 | | |
| | * e_{sl} | 引發訊息之事件 | | |
| | * et_l | 訊息將觸發之事件 | | |
| | * st_{mi} | 訊息最少花費之時間 | | |
| | * tt_{mi} | 訊息最長花費之時間 | | |

表 11 Step3 之物件導向時間派翠網路規格示意表

(星號為表示對應。)

| | |
|---|--------|
| OOTPNS (Object Oriented Tine Petri Net Specification) | |
| P | *位置之集合 |

| | |
|-------|---------|
| T | 轉置之集合 |
| B | 反向關聯函數 |
| F | 正向關聯函數 |
| M_0 | *初始標記 |
| *I | 時間區隔 |
| *OP | 訊息位置之集合 |
| *G | 開道之集合 |
| *C | 物件類別之集合 |

第 4 章 範例及應用

4.1 圖形轉換範例

在本章節中，將以一簡易之行動通訊裝置，探討其基本功能之時間需求，此行動通訊裝置功能有接收他人來電以及撥出通話，本範例以撥出通話請求至基地台為主，如圖 14 所示。在需求中分為二種物件：行動通訊裝置(Mobile Device)及基地台(Base Station)，因此將分別對二者內部之時間狀態及交互影響之時間需求列出，在此範例中定時間單位為 10ms。

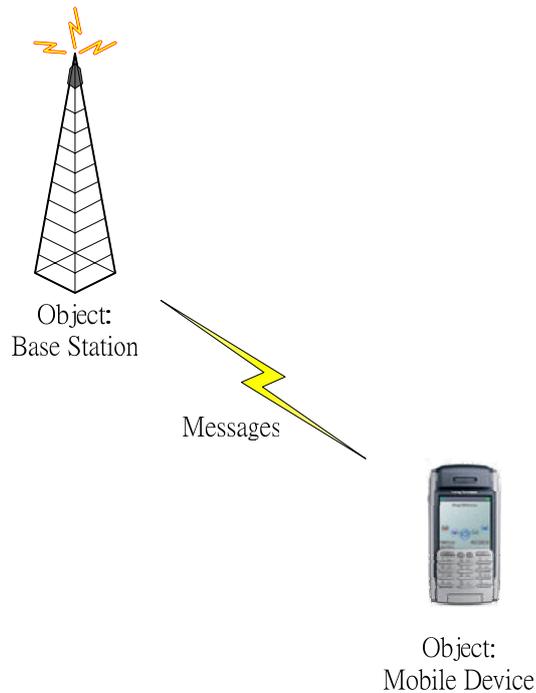


圖 14 範例之簡易架構示意圖

以下即為時間性需求之描述：其中 A~J 表示事件，其中 A~G 表示行動裝置的事件，H~J 表示基地台的事件。K,L,M 分別表示不同的訊息，N,O 為持續時間，N 表示行動裝置的持續時間，O 表示基地台的持續時間。

表 12 範例之時間性需求表格

| 代碼 | 描述 |
|----|--|
| A | 當行動通訊裝置接收到來電訊號，必須在 5 個時間單位內做出反應（如振動、響鈴等）。 |
| B | 使用者可取消或超過 150 個時間單位，需在 150 個時間單位內完成。 |
| C | 當使用者接聽來電之後，需在 10 個時間單位內進入通話狀態。 |
| D | 當使用者或對方結束通話後，在 10 個時間單位內回到待機狀態。 |
| E | 在使用者撥號後，需在 25 個時間單位內傳送訊息並進入等待通話狀態。 |
| F | 若是對方目前無法通話，則在 25 個時間單位內回到待機狀態。 |
| G | 當基地台建立雙方連線後，需在 25 個時間單位進行同步等動作確立連線成功，進入通話狀態。 |
| H | 當基地台從閒置狀況收到使用者撥號要求時，需在 10 個時間單位內進入確認狀態。 |
| I | 若是基地台確認對方目前狀況為無法通話，則在 10 個時間單位內準備發送訊息，並回到閒置狀態。 |
| J | 當確定對方目前可通話時，在 20 個時間單位內進行連線，並回到閒置狀態。 |
| K | 由使用者行動通訊裝置發給基地台的請求需在 15 個時間單位內傳至基地台。 |
| L | 基地台傳送至使用者行動通訊裝置需在 15 個時間內傳送。 |
| M | 當基地台確認使用者所撥出之號碼為有效並建立連線的時間需在 10 個時間單位內完成 |
| N | 確認狀況需要 20 個時間單位。 |
| O | 通話中之狀態為 30 到 50 個時間單位內。 |

經由上述規格描述，我們繪製一個時間圖如圖 15 所示。

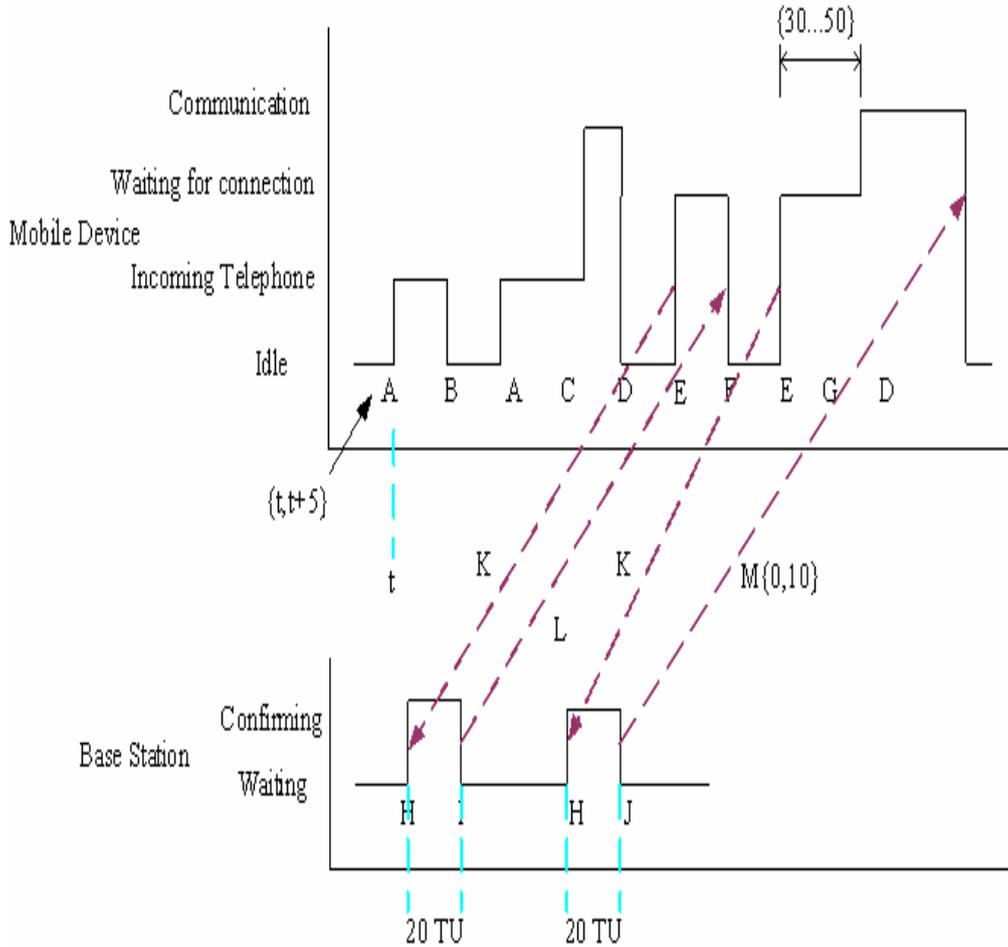


圖 15 依範例需求表格所繪製之時間圖

接下來我們依據第三章之定義，實作上述的範例。

- (1) 首先我們定義 $TDS=(10ms, L, M)$ ， $M = \{m_K, m_L, m_M\}$ ， $m_K = \{K, \text{行動通訊裝置, 基地台, } 0TU, 15TU, e_E, e_H\}$ ， $m_L = \{L, \text{基地台, 行動通訊裝置, } 0TU, 15TU, e_I, e_F\}$ ， $m_M = \{M, \text{基地台, 行動通訊裝置, } 0TU, 10TU, e_J, e_G\}$ ， $L = \{l_1, l_2\}$

- (a) 其中 $l_1 = (\text{行動通訊裝置, } S_1, \text{待機s1, } 0, E_1, DC_1)$ ， $S_1 = \{\text{待機s1, 顯示來電s2, 等待通話s3, 通話中s4}\}$ ， $E_1 = \{e_A, e_B, e_C, e_D, e_E, e_F, e_G\}$ ， $DC_1 = \{dc_O\}$ ， $dc_O = (30TU, 50TU)$

- $e_A=(A, \text{待機}, \text{顯示來電}, 0TU, 5TU)$
- $e_B=(B, \text{顯示來電}, \text{待機}, 0TU, 150TU)$
- $e_C=(C, \text{顯示來電}, \text{通話中}, 0TU, 10TU)$
- $e_D=(D, \text{通話中}, \text{待機}, 0TU, 10TU)$
- $e_E=(E, \text{待機}, \text{等待通話}, 0TU, 25TU)$
- $e_F=(F, \text{等待通話}, \text{待機}, 0TU, 25TU)$
- $e_G=(G, \text{等待通話}, \text{通話中}, 0TU, 25TU)$

(b) $l_2 = (\text{基地台}, S_2, \text{等待}s1, 0, E_2, DC_2)$, $S_2 = \{\text{等待}s1, \text{確認}s2\}$, $E_2 = \{e_H, e_I, e_J\}$ $DC_1 = \{dc_o\}$ $dc_o = (30TU, 50TU)$

- $e_H=(H, \text{等待}, \text{確認}, 0TU, 10TU)$
- $e_I=(I, \text{確認}, \text{等待}, 0TU, 10TU)$
- $e_J=(J, \text{確認}, \text{等待}, 0TU, 20TU)$

(2) l_1 的實體 s1 A s2 B s1 A s2 C s4 D s1 E s3 F s1 E s3 G s4 D s4

l_2 的實體 s1 H s2 I s1 H s2 J s1

接下來進行轉換 TDS 至 OOPTNS 之步驟：

Step 1:

$L = \{l_1, l_2\} = C = \{c_1, c_2\}$ 如圖 所示，

$c_1 = l_1$, $c_2 = l_2$ 。

$S_1 = \{\text{待機}s1, \text{顯示來電}s2, \text{等待通話}s3, \text{通話中}s4\}$ 轉換為 $P =$

$\{p_1, p_2, p_3, p_4\} \in c_1$,

$p_1 = s1$, $p_2 = s2$, $p_3 = s3$, $p_4 = s4$ 。

$S_2 = \{\text{等待}s1, \text{確認}s2\}$ 轉換為 $P = \{p_5, p_6\} \in c_2$,

$p_5 = s1$, $p_6 = s2$ 。

$E_1 = \{e_A, e_B, e_C, e_D, e_E, e_F, e_G\}$ 轉換為 $T_1 = \{A, B, C, \dots, G\} \in c_1$,

$A = e_A, B = e_B, C = e_C, D = e_D, E = e_E, F = e_F, G = e_G$ 。

而 l_1 中之起始狀態 s_1 和 l_2 中之起始狀態轉換為 M_0 ,

即 $M_0 = (1\ 0\ 0\ 0\ 1\ 0\ 0)$, 其中前四項為 l_1 轉換後對應之位置, 即 p_1 至 p_4 , 而後三項為 l_2 轉換後對應之位置。

Step 2:

各事件 e 與其對應 $I(t_i)$ 中, 將 TDS 中各事件所花費時間對應至 OOTPNs 中各轉置之時間區隔:

$$st_{e_i} = EFT(t_i), et_{e_i} = LFT(t_i)。$$

如事件 A 所花費時間為 0 至 5 個時間單位, 因此對應之時間區隔為:

$$EFT(t_a) = 0, LFT(t_a) = 0,$$

令 $ss_{e_i} = B(t_i)$, 將 TDS 中各事件之起始狀態視為 OOTPNs 中之反向關聯函數。

如事件 A 起始狀態為 s_1 , 對應之 OOTPNs 之位置為 p_1 , 因此 $B(t_a)$ 中即為 $(1\ 0\ 0\ 0\ 0\ 0\ 0)$ 。

令 $ts_{e_i} = F(t_i)$, 將 TDS 中各事件之終止狀態視為 OOTPNs 中之正向關聯函數。

如事件 A 終止狀態為 s_2 , 對應之 OOTPNs 之位置為 p_2 , 因此 $F(t_a)$ 中即為 $(0\ 1\ 0\ 0\ 0\ 0\ 0)$ 。

Step 3:

針對訊息 K, L, M 進行轉換, 讓我們以訊息 K 為例:

1. m_k 中之訊息起始生命線 sl_{m_i} 為 C1, 因此在 C1 建立訊息位置 OP1

2. m_k 中之訊息終止生命線 tl_{m_i} 為 C2, 因此在 C2 建立訊息位置 OP2

3. 在訊息位置 OP1 與 OP2 間建立開道 K 以代表 m_k 。

4. 根據 m_k 中紀錄訊息起始之生命線 sl_{m_i} ，將 OP1 連結至 k。
5. 根據 m_k 中紀錄訊息終止之生命線 tl_{m_i} ，將 K 連結至 OP1。
6. 根據 m_k 中紀錄引發訊息之事件 e_{sl} ，將 E 連結至 OP1。
7. 根據 m_k 中紀錄訊息將觸發之事件 e_{tl} ，將 OP2 連結至 H。
8. m_k 中之 st_{m_i}, tt_{m_i} 轉換為時間區間，亦即 $st_{m_i} = EFT(m_k), tt_{m_i} = LFT(m_k)$ ， $I(m_k) = [EFT(m_k), LFT(m_k)] = [0, 10]$ 。

最後我們可得到如圖 16 之物件導向時間派翠網路。

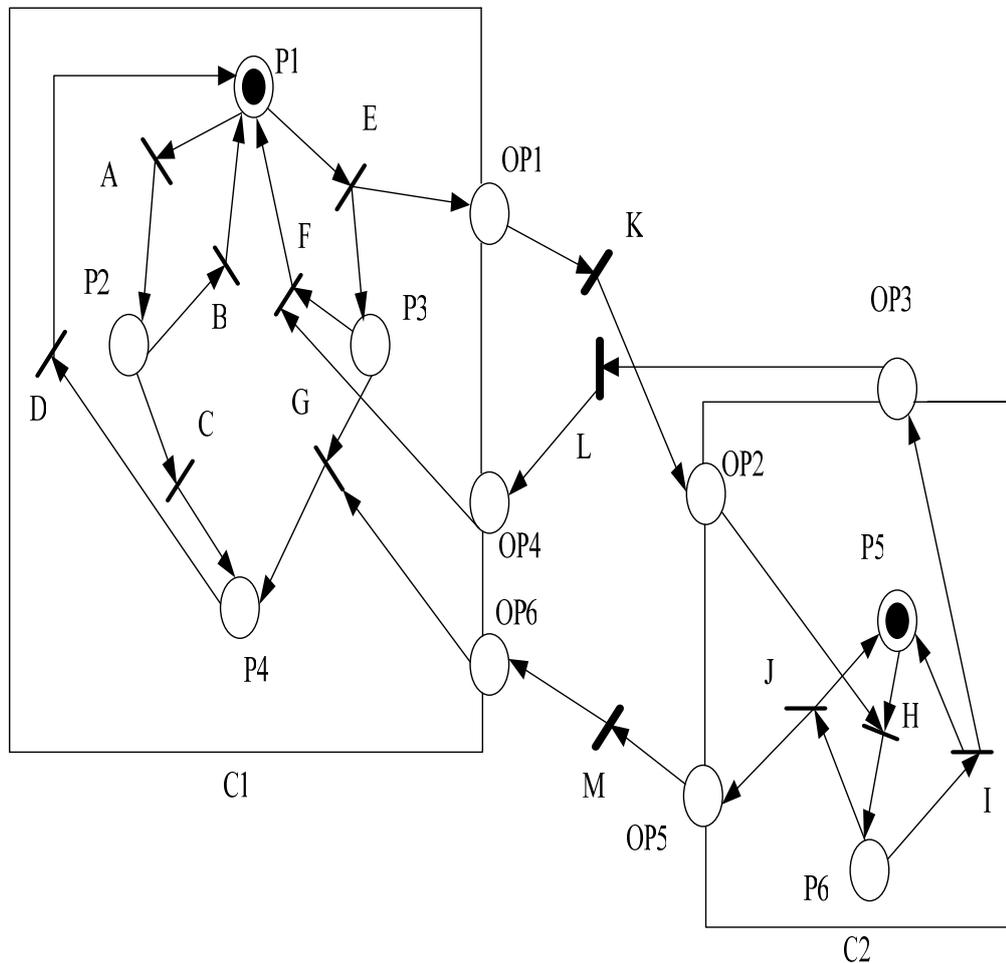


圖 16 轉換圖 15 所得之物件導向時間派翠網路

表 13 圖 16 中各位置之說明

(其中轉置之時間間隔在以文字表達之時間需求中。)

| 位置 | 定義 |
|-----|----------------------|
| P1 | 行動通訊裝置之待機狀態。 |
| P2 | 行動通訊裝置之顯示來電狀態。 |
| P3 | 行動通訊裝置之等待通話狀態。 |
| P4 | 行動通訊裝置之通話中狀態。 |
| P5 | 基地台之等待狀態。 |
| P6 | 基地台之確認狀態。 |
| OP1 | 本方法處理時間圖中之訊息所產生之訊息位置 |
| OP2 | 同上。 |
| OP3 | 同上。 |
| OP4 | 同上。 |
| OP5 | 同上。 |
| OP6 | 同上。 |

4.2 應用範例

在本章節中以[9]提出方法對前部分所述之範圍做可達性分析(Reachability Analysis)，在[9]中除了對時間派翠網路之定義外（見 2.4 中提到之時間派翠網路定義），還新增了對時間部分之定義 CS-Class (Clock Stamped state class),以下為 CS-Class 之定義。

CS-class 由三個欄位組成， $C=(M,D,ST)$ ，其中

1. M 為狀態標記。
2. D 為激發範圍(Firing Domain)，其中紀錄在某一狀態下可被激發之轉置及其時間區間D (t_i)代表可被激發的時間區間，D(t_i)之下限為t_i之EFT，上限為t_i之LFT，以上章結之範例，在M₀狀態時可被激發之轉置有A及E，因此
 $D_0:D_0(A)=[0,5],D_0(E)=[0,25]$
3. ST 為時間標誌(time stamp)，代表進行至目前狀態之全域時間。

依據[9]中之定義，處理上章節之範例，由物件導向時間派翠網路及需求規格所得的資訊，其中標記 $M=(P1,P2,P3,P4,P5,P6,OP1,OP2,OP3,OP4,OP5,OP6)$ ，我們可知在最初狀態時CS-class C₀中之M₀= (100010000000)，而可被激發的轉置有A、E兩項，時間區間分別為I(A)=[0,5]及I(E)=[0,25]，而狀態之時間標誌為[0,0]，於是我們可以得到如下之CS-class C₀:

$$\begin{aligned}C_0:M_0: (100010000000) \\ ST_0=[0,0] \\ D_0:D_0(A)=[0,5], \\ D_0(E)=[0,25].\end{aligned}$$

以事件A被激發為例，根據[9]中所提出之計算步驟，我們進行以下的計算，以求出A被激發後之CS-class C₁:

1. $ST_1 = [EFT_0(A), \min(LFT_0(A),LFT_0(E))]$
 $= [0,\min(5,25)] = [0,5]$

$$2.M_1' = M_0 - B(A) = (000010000000)$$

$$\begin{aligned} D_1(B) &= [\max(EFT_0(B), EFT_0(A)), LFT_0(B)] \\ &= [\max(0, 0), 155] \\ &= [0, 155] \end{aligned}$$

$$\begin{aligned} D_2(C) &= [\max(EFT_0(C), EFT_0(A)), LFT_0(C)] \\ &= [\max(0, 0), 15] \\ &= [0, 15] \end{aligned}$$

$$3.M_1 = M_1' + F(A) = (010010000000)$$

於是我們可以得到CS-class C_1 如以下所示：

$$C_1:M_1: (010010000000)$$

$$ST_1=[0,5]$$

$$D_1:D_1(B)=[0,155],$$

$$D_1(C)=[0,15].$$

繼續依據[9]中之方法計算，我們可得到圖 17 的可達樹(Reachability Tree)。

以上計算步驟之詳細內容，可參考[9]中提出之方法。

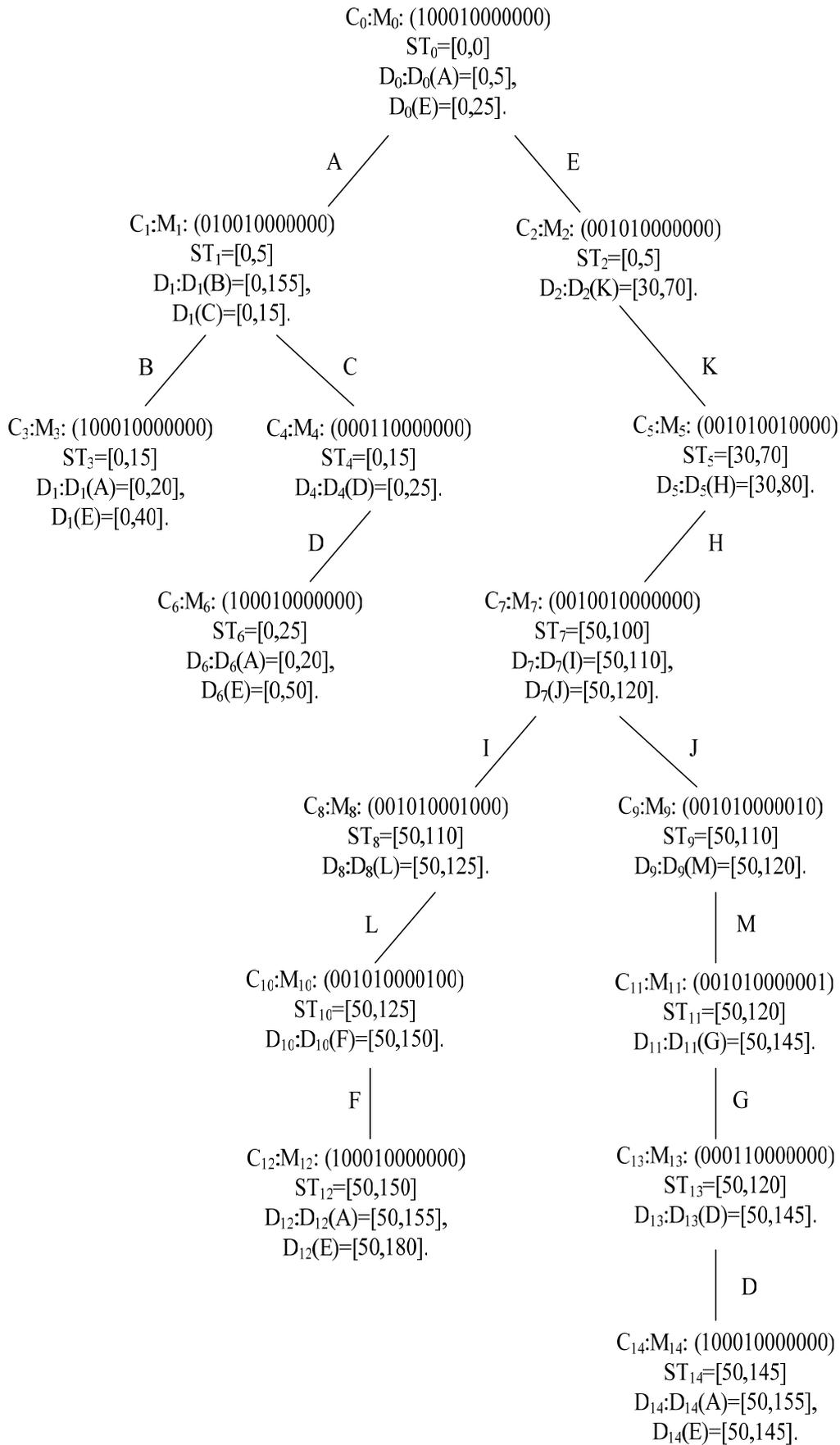


圖 17 依[9]之方法計算圖 16 所得之可達樹

在可達樹建立後，我們可以依據其中之內容進行分析，如下圖 18 中之標記部分所示，在B轉置被激發後，時間標誌 ST_3 只有 $[0,15]$ ，這並不符合事件B中要等待至 150 個時間單位的需求，在此狀況之下，當時間超過 15 個時間單位後，行動通訊裝置將無法被接聽，直到此狀態維持 150 個時間單位，因此我們可以發現需求中的事件B需進行修正，需指定狀態持續時間，並修改處理此事件之時間間隔。

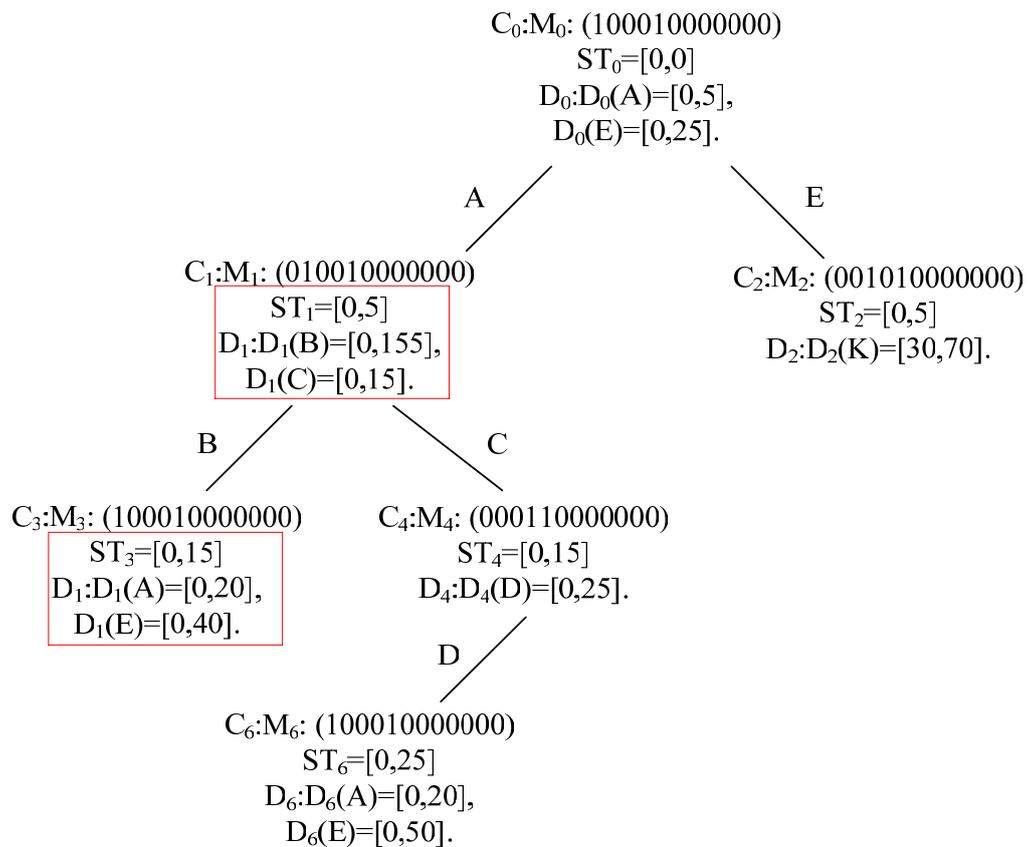


圖 18 可達樹表示之分析結果

第 5 章 結論及未來方向

隨著資訊時代的來臨，資訊產業的重要性日益提高，人們的生活與計算機相關產品可以說是息息相關，其中，即時性系統產品更扮演了極為重要的角色，並在近年來有著日趨重要。然而對即時性系統而言，十分重要的特性就是在對反應時間及處理時間等時間性特質上有著相當的要求，時間性特質對即時系統影響毋庸置疑。

近年來統一塑模語言(Unified Modeling Language, UML)逐日盛行，開發的組織 OMG(Object Management Group)也意識到時間性對未來系統的重要性將是與日俱增，因此於 UML 2.0 中新增時間圖(Timing Diagram)，讓開發人員能對物件互動與時間性質進行更細部的描述。

然而，在開發者針對系統之時間性進行時間圖之描繪時，對其正確性難以進行判斷，在需求或設計階段所產生的錯誤或矛盾在時間圖上仍然可能無法被發現，進而影響到其後的開發過程。

在本論文中，提出轉換時間圖至物件導向時間派翠網路之轉換規則，以補足時間圖在分析與驗證上不足的部分，在派翠網路的相關研究上，已有許多的學者提出針對不同系統、不同特性之各種研究，諸如同步、資源共享、可達性……等方面，發展出各種分析的方法。

而本論文則以 J. Wang, Y. Deng 和 G. Xu 提出的可達性分析法做為轉換時間圖至物件導向派翠網路後之應用分析範例，並以一簡易行動通訊裝置與基地台之通話互動為例。

然而，在本研究中對於時間圖中之訊息(Message)部分，對於在時間圖原本定義時間圖的幾種訊號類型，皆視為非同步訊號，在未來，此一部份問題仍需要詳細定義以及改進。

對於正規化轉換部分，若要實作自動化工具，則需再將時間圖規格及物件導向時間派翠網路中所定義之各項欄位進行更細部的資料結構定義，這點在本論文

所提出的各種表格並無法完全明確地表達。

再者，本論文使用為例之可達性分析方法著重於時間性部分，而派翠網路可分析之特性相當多（如 Liveness, Safeness, Boundness），本論文提出之轉換規則仍需要經過其他方法之驗證，以確定此一轉換方法是否適合其他不同種類之派翠網路之分析法。

在未來，本論文之方法論仍需要改進以上二種問題，並應以更多實際範例對此方法進行驗證，以增加此方法之效用。除了方法論之外，在工具的實作及改進上配合較好的使用者介面也是我們努力的目標。

參考文獻

- [1] Chang-Pin Lin, Yi-Pin Lin, Min Der Jeng, "Design of intelligent manufacturing systems by using UML and Petri net," *2004 IEEE International Conference on Networking, Sensing and Control*, vol1, pp. 501-506, 2004.
- [2] Chun-Che Huang and Wen Yau Liang, "Object-oriented development of the embedded system based on Petri-Nets," *Computer Standards & Interfaces* 26, pp 187-203, 2004
- [3] Jin-Shyan Le, Pau-Lo Hsu, "Design and implementation of the SNMP agents for remote monitoring and control via UML and Petri nets," *IEEE Transactions on Control Systems Technology*, vol.12, no. 2, pp. 293-302, 2004.
- [4] Martin Flower, "UML Distilled third edition," Addison Wesley, 2003.
- [5] Object Management Group, "UML 2.0 Superstructure Final Adopted specification," 2003, document number: ptc/03-08-02.
- [6] Object Management Group, "UML 2.0 Infrastructure Final Adopted specification," 2003, document number: ptc/03-08-02.
- [7] Jo I Chen, "Apply Object-Oriented Concept and Petri Net into the development of the embedded system," A master's dissertation of information management, Da-Yeh University, Taiwan, 2000.
- [8] Jose Marcelino Arrozal Nicdao, "Fundamental Structures in Petri Nets," A master's dissertation, National Cheng Chi University, Taiwan, 2000.
- [9] J. Wang, Y. Deng and G. Xu., "Reachability analysis of real-time systems using time Petri nets," *IEEE Transactions on Systems, Man and Cybernetics- Part B.*, Vol. 30, no. 5, pp. 725-736, 2000.
- [10] Denis Mukhin, and Boleslaw Mikolajczak, "A Method of Concurrent Object-oriented Design Using High-Level Petri Nets," IEEE International Conference,

Vol. 1, pp. 295-300, 1998.

[11] Dar-Chin Rau, Chien-Yun Dai and Ching-Wen Chiou, “ An Object Oriented Petri Nets to Construct Prototype of Hypermedia System,” Proceeding, IASTED International Conference Applied Informatics, Austria, February 21-23, 1995.

[12] Zurawski, R. and MengChu Zhou, “Petri nets and industrial applications: A tutorial,” *IEEE Transactions on Industrial Electronics*, Vol. 41 , no. 6, pp. 567-583, 1994

[13] Wang, L, and Chang Y. J., “The Development of and Object-oriented Petri Net Model,” Working paper W06/93, Department of Industrial Engineering, Tunghai University.

[14] Y.K. Lee and S.J. Park, “OPNets: an object-oriented high level Petri net model for real-time system modeling,” *Journal of Systems Software* 20, pp. 69-86, 1993.

[15] David Rene and Alla Hassane, “Petri Nets and Grafcet,” Prentice Hall, 1992.

[16] B. Berthomieu and M. Diaz, “Modeling and verification of time dependent systems using time Petri Nets,” *IEEE transaction on Software Engineering*, vol. 17, no. 3, pp. 259-273, 1991

[17] C. Sibertin-Blane, R. Bastide, “Object-Oriented Structure for High Level Petri Nets,” *11th Conference of Application and Theory of Petri Nets*, Toulouse, France, 1990.

[18] T. Muraya, “Petri nets: Properties, analysis and applications,” *Proceedings of the IEEE*, Vol. 77, No. 4, pp. 541-580, 1989

[19] J.L. Peterson, “Petri Net Theory and the Modeling of Systems,” Prentice-Hall, Englewood Cliffs, N. J., 1981.

[20] P. M. Merlin and D. J. Farber, “Recoverability of Communication Protocols - Implications of a Theoretical Study,” *IEEE Trans. on Communications*, vol. 24, no. 9, pp. 1036–1043, 1976.

- [21] C. A. Petri, "Kommunikation mit Automaten," A Ph.D. dissertation, University of Bonn, Bonn, 1962.
- [22] Borland, <http://www.borland.com.tw/>
- [23] Petri Nets Worlds, <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- [24] 劉儒斌, 以時間派翠網路進行 RosettaNet PIPs 之死結驗證, 暨南大學資訊管理研究所碩士論文, 2002.
- [25] 林木盛, 個體式派曲網路做系統發展方法, 師範大學工業教育研究所碩士論文, 1993.