

## 第四章 實驗過程與結果

為了驗證所提出的預測模型之準確性，我們根據大學課程網網站架構，以隨機 (Random) 的方式來產生代表網站瀏覽者複雜行為的記錄檔。該網站總共有50頁左右不同的網頁，依循這個架構，我們根據資料庫裏的路徑資料庫再隨機產生2000筆已經由前處理過的使用者瀏覽模式，其中每一筆的長度為10頁，若超過10頁的使用者行為模式，我們一律將這些行為模式依每個10頁的長度取出，不足10頁部份則捨去不用。

### 4.1 資料來源選定

在實驗中，我們以90%，約1800 筆的使用者行為區段資料，做為我們的訓練模式 (Training Pattern) 用來建立我們的預測模型，而剩下的10%，約200 筆的使用者行為區段資料則用來作為測試模式 (Testing Pattern) 來測試我們建立的預測模型的正確性。然而，就實際的運作而言，利用網站歷史存取紀錄(Log Database)建立預測模型時，必須先對原始的網站記錄資料，根據本身的需求進行資料前處理(Data Preprocessing)，取出有意義的使用者行為區段後，再以我們所提出的預測模型建構階段的各個步驟與方法建立預測模型。如前所述，我們設計的即是假設已經經過前處理完畢的資料。

### 4.2 實驗過程

為了讓我們測試的結果更具可信度，我們利用交叉驗證中，10-fold 的方式，來反覆測試我們的模型，並取實驗的平均值來解說我們模型的可靠度。所謂10-fold [25]的方式就是將2000 筆以隨機方式產生的使用者行為區段，分成10 份，每次只取1 份做為測試模式，另外9 份便做為建立模型

的網站歷史紀錄，也就是訓練模式，依此方式共做10次，也就是讓每份資料都有機會當測試模式，不會讓某些模式只當訓練模式，而有所偏差以致影響了整個模型的正確度，或者影響模型準確度的公信力。

如上所述，我們在建立預測模型的階段中，利用1800筆歷史使用者行為區段的前置部份(Prefix)，以模式發掘的方式找出具代表性的行為模式，整合所屬的後置行為區段做預測模式集，用來建立我們的預測模型。

首先，我們先將每一筆測試模式(Testing Pattern)分為兩段，前段為新使用者已瀏覽過的頁面，稱為瀏覽模式，而後段則為使用者在我們預測後所瀏覽的頁面，稱為驗證模式。然後，分別將2000筆的瀏覽模式利用所提出的預測模組進行預測，先以相似度計算從預測模式中選出相似度在門檻值以上的參考模式(Reference Pattern)以產生預測模組，最後再利用Petri-net所對應的預測結果，評量方式中的兩項指標對驗證模式與預測結果的序列，以頁面正確率進行評量，觀察預測系統預測的準確度。

### 4.3 實驗結果

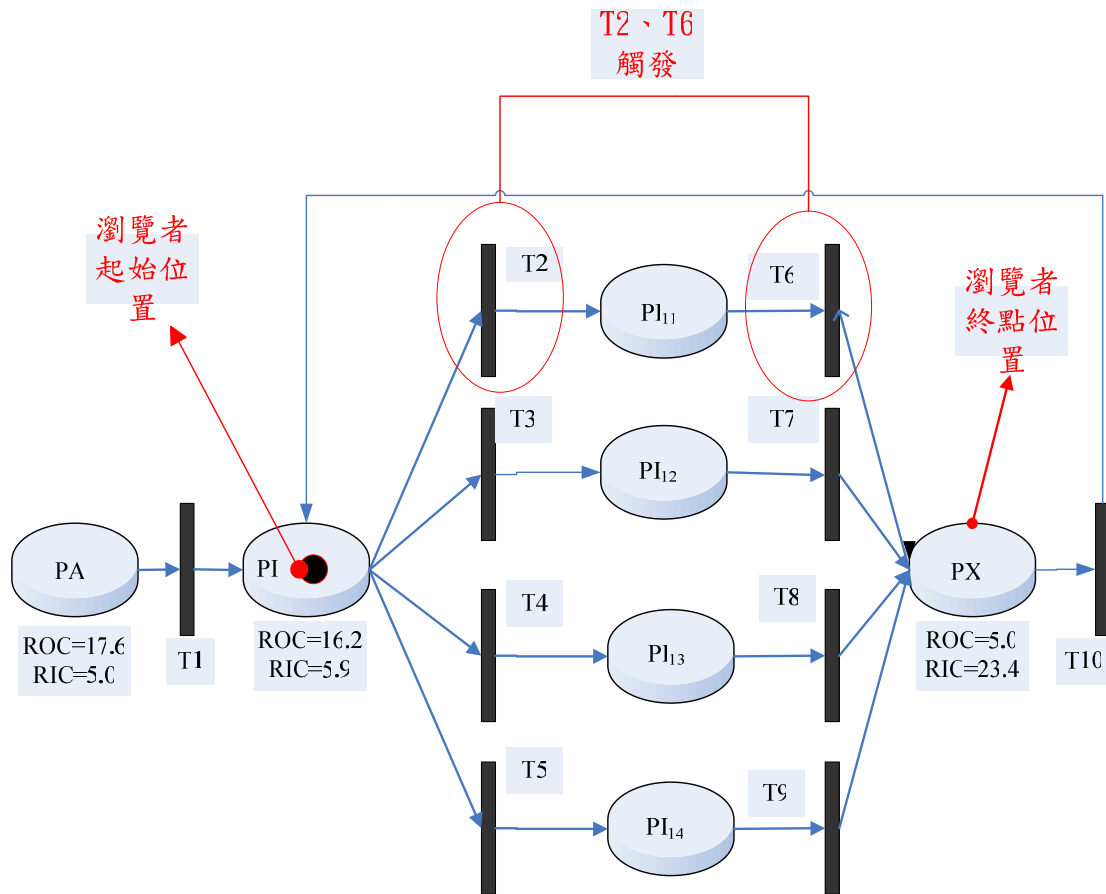


圖 4-1 使用者經過 T2、T6 觸發後的狀態路徑圖

(資料來源:本研究自行整理)

以圖 4-1 使用者進入簡易查詢網頁路徑來分析，經過觸發後的狀態

$Mk = Mi \{S\} = Mi + S \cdot D$  [26] 其中  $D$  為轉化控制矩陣，若我們假設

$$Mi = [0, 1, 0, 0, 0, 0, 0]$$

$$S = [0, 1, 0, 0, 0, 1, 0, 0, 0]$$

$$D^- = \begin{bmatrix} & PA & PI & PI11 & PI12 & PI13 & PI14 & PX \\ T1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ T2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T3 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T5 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T6 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T7 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ T8 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ T9 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D^+ = \begin{bmatrix} & PA & PI & PI11 & PI12 & PI13 & PI14 & PX \\ T1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ T2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ T3 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ T4 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ T5 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ T6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ T7 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ T8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ T9 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = D^+ - D^- = \begin{bmatrix} & PA & PI & PI11 & PI12 & PI13 & PI14 & PX \\ T1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ T2 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ T3 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ T4 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ T5 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ T6 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ T7 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ T8 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ T9 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

在轉換矩陣  $D$  的橫軸代表網頁狀態，而縱軸代表控制觸發按鈕或選項，-1 值代表  $T$  與  $P$  之間輸出結果，反之 1 值則  $T$  與  $P$  之間輸入結果，而 0 值代表無關連。經過觸發後的狀態為  $Mk_2$

$$Mk_2 = Mi + S \cdot D$$

$$= \{0, 1, 0, 0, 0, 0, 0, 0\} + \{0, 1, 0, 0, 0, 0, 1, 0, 0, 0\} \cdot D$$

$$= \{0, 1, 0, 0, 0, 0, 0, 0\} + \{0, -1, 0, 0, 0, 0, 0, 1\}$$

$$= \{0, 0, 0, 0, 0, 0, 0, 1\}$$

由以上可知，即是滿足 $T2$ 、 $T6$ 的轉換之後，系統已由狀態 $P1$ 轉變至 $Px$ ， $Mk$ 可作為系統紀錄最後的網頁狀態。本研究以資料探勘及Petri-net方法來建立預測模組，以Petri-net描述每一條規則所對應的動態行為細節，可將之轉換成預測模型回應動作。

根據第三章的概念，對於網路服務組合Petri-net模型的性質將採用 Incidence Matrix and State Equation 分析方法驗證。經由此分析方法傳送進來的Petri-net模型物件參數，透過上述的演算法可以分別建構出 Incidence Matrix and State Equation 方法所需要的輸入矩陣 $D^-$ 和輸出矩陣 $D^+$ ，運用 Matlab 工具將相對應的預測動作轉成程式碼，將此預測動作自動模組化。表4-1、4-2、4-3為預測模型所建立矩陣的演算法。藉由實作 $D = D^+ - D^-$  運算式的演算法如表4-3 所示，建構出系統模型矩陣 $D$ 。

表4-1 輸入矩陣 $D^-$ 建構演算法

```

int[][] inMatrix =
    new int[places.length][transitions.length];
for(int i=0;i< places.length;i++){
    for(int j=0;j< transitions.length;j++){
        for(int k=0;k< arcsrc.length;k++){
            if((places[i].equals(arcsrc[k]))& (transitions [j].equals(arctrng [k]))) {
                inMatrix[i][j] = "1";
            }
        }
    }
}

```

(資料來源:本研究自行整理)

表4-2 輸出矩陣 $D^+$ 建構演算法

```
int[][] outMatrix =
    new int[places.length][transitions.length];
for(int i=0;i< places.length;i++){
    for(int j=0;j< transitions.length;j++){
        for(int k=0;k< arcsrc.length;k++){
            if((transitions [j].equals(arcsrc[k]))& (places [i].equals(arcstrg[k]))){
                outMatrix[i][j] = "1";
            }
        }
    }
}
```

(資料來源:本研究自行整理)

表4-3 系統模型矩陣 $D$ 建構演算法

```
int[][] modelMatrix =
    new int[places.length][transitions.length];
for(int i=0;i< places.length;i++){
    for(int j=0;j< transitions.length;j++){
        modelMatrix[i][j] = outMatrix[i][j] - inMatrix[i][j];
    }
}
```

(資料來源:本研究自行整理)

我們將所上述的轉換控制矩陣法演算法輸入到Artifex這套軟體，它是一套根據Petri-net理論架構所組成的模擬設計軟體。Artifex利用圖型化的程式語言(C語言、Matlab)，能夠描繪出網路系統的動態。

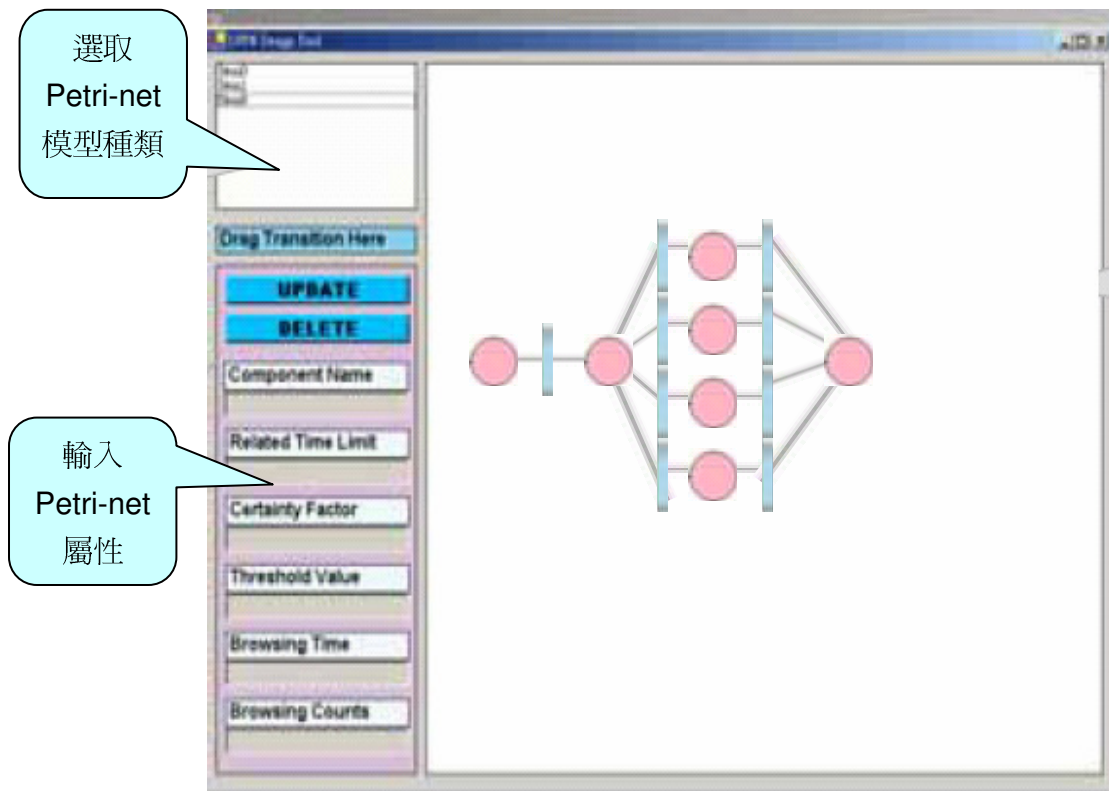


圖4-2 Artifex模擬畫面

(資料來源:本研究自行整理)

圖4-2為運用Artifex軟體所模擬的畫面，首先選取Petri-net的模型種類，選取Petri-net的屬性，接著輸入狀態控制矩陣演算法(表4-1、4-2、4-3)到每一條路徑所對應的動作，建立對應動作的規則庫，所模擬的結果實驗後可得到簡化後的網路流程組合。圖4-3所示，我們將此網路服務組合簡化前後四條執行路徑的流程傳輸次數整理成表4-4，從表中可以發現，除了第一條路徑之外，無論是執行路徑中的那一條，簡化後的傳輸次數必定會比簡化前來的少。然而這僅是一個簡單的網路服務組合，對於企業實際應用的網路服務組合來說，其流程設計的複雜度必定高於此實驗好幾倍，因此經過簡化後所能夠減少的傳輸次數必定更多。從以上的範例驗證能夠看出，藉由流程的簡化而減少網路傳輸次數將能提升網路服務組合的執行效能。

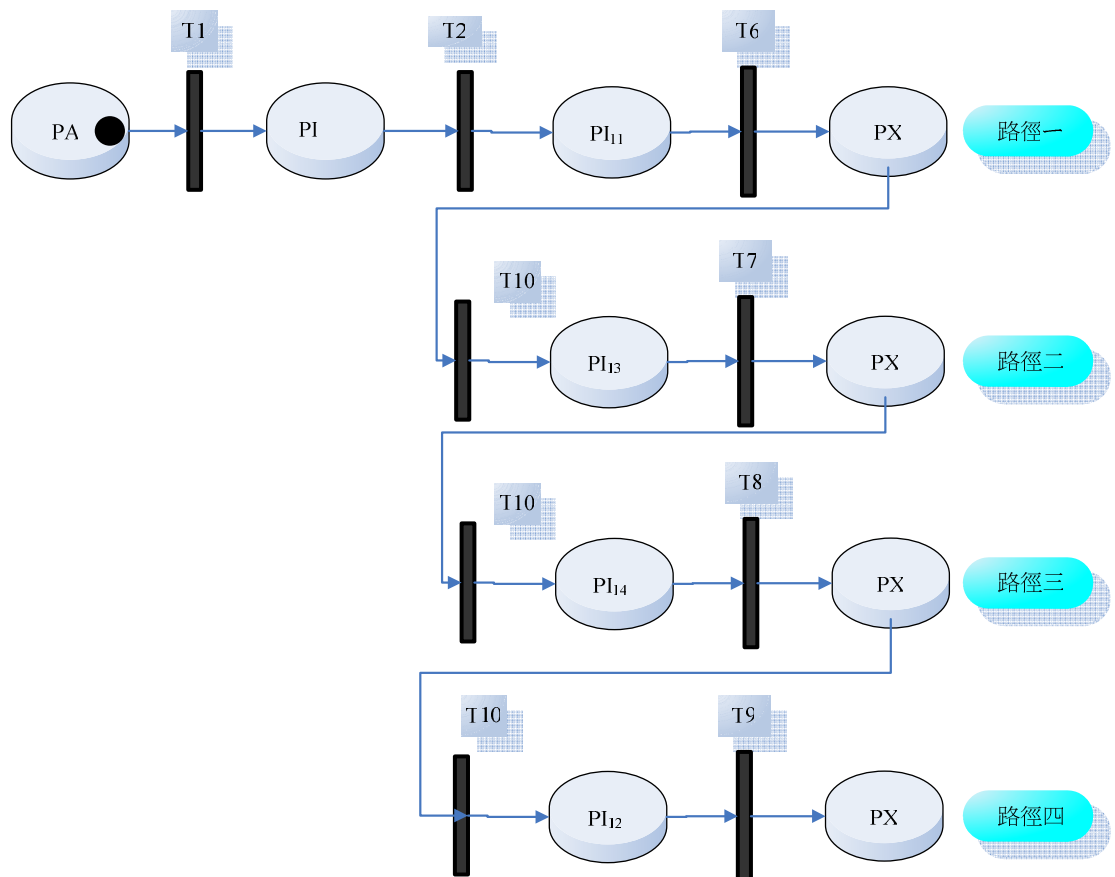


圖4-3 網頁簡化後傳輸路徑次數圖

(資料來源:本研究自行整理)

表4-4 網頁簡化前後傳輸次數比較表

傳輸次數	簡化前 (圖4-1)	簡化後 (圖4-3)
路徑一 (PA→PI→PI <sub>11</sub> )	3	3
路徑二 (PA→PI→PI <sub>11</sub> →PI <sub>13</sub> )	6	5
路徑三 (PA→PI→PI <sub>11</sub> →PI <sub>13</sub> →PI <sub>14</sub> )	9	7
路徑四 (PA→PI→PI <sub>11</sub> →PI <sub>13</sub> →PI <sub>14</sub> →PI <sub>12</sub> )	12	9

(資料來源:本研究自行整理)



我們的實驗結果將比較在不同長度的比對模式(Match Pattern)下，各個相似度門檻值( $\theta$ )時，預測結果的頁面正確率( $\alpha$ )。以下我們就 $\theta=1, 2, 3$ 的情形進行實驗。

我們提出的頁面正確率公式如下： $\alpha = \frac{count}{length}$  [26]，其中 $count$ 代表以測試模式(Testing Pattern)中的比對模式所預測的結果，與測試模式的後置行為模式相同的頁面數。而 $length$ 表示測試模式中後置行為模式長度。例如： $T$  為測試模式中的驗證模式， $N$  為預測模型預測的結果，若 $T=\{ABCDE\}$ ， $N=\{ACIDJ\}$ ，則 $\alpha=3/5$ 。表4-5為預測系統的準確度。

我們定義 $\theta$ 為頁面比對數，若 $\theta$ 設為1 時，表示至少一個頁面被比對到。當 $\theta$ 設為2 時，表示瀏覽模式與比對模式至少吻合兩頁。當 $\theta$ 設為3 時，表示排除只有前二頁比對到的所有狀況，而包含至少三頁有比對到的情況。

表4-5 預測系統的準確度表格

相似度	PI <sub>11</sub>	PI <sub>13</sub>	PI <sub>14</sub>	PI <sub>12</sub>
$\theta=0$	71%	60%	44%	12%
$\theta=1$	X	73%	62%	46%
$\theta=2$	X	X	74%	57%
$\theta=3$	X	X	X	80%

(資料來源:本研究自行整理)

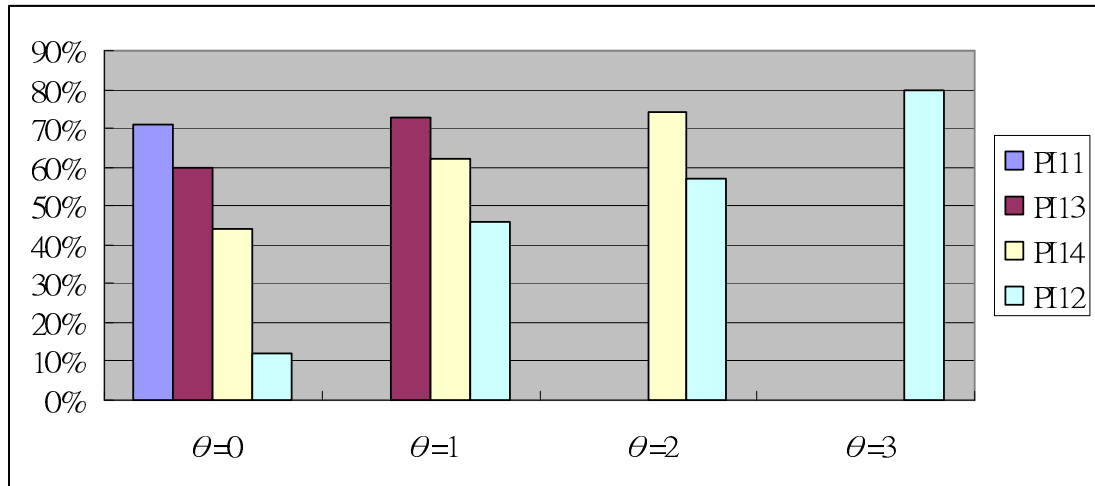


圖4-4 預測系統的準確度圖形

表4-5為在 $\theta=0、1、2、3$ 時，預測系統的準確度。我們可以知道使用者越後面的瀏覽行為與比對模式的相似度越高則預測確度較高。所以，由表可知我們的預測系統在比對模式長度為3時，準確度最高。

本篇論文中所提出的預測模型，確實可以提供我們做為預測的基礎。Petri-net是取相似度門檻值之上的預測模式做預測，而單一預測則是取相似度最高的預測模式做預測，可驗證本篇論文所提以Petri-net預測網路使用者路徑確實能符合我們的預測。因此，可以證明加入Petri-net預測方式的確比只用單一比對的方式更適合運用在預測網路使用者瀏覽路徑。

一般都是以瀏覽頁面的相似度來預測使用者的瀏覽路徑，而我們則是提出 Petri-net 來做預測。在預測網路使用者瀏覽路徑時，驗證本篇論文所提出來之預測方法確實可行。而預測模組所預測的位置順序為  $PI_{11}$ 、 $PI_{13}$ 、 $PI_{14}$ 、 $PI_{12}$ ，與實際網頁的架構不同。預測系統能夠做預先處理，藉此改善網頁架構，由此可證明，本篇論文提出的以 Petri-net 建立網路使用者預測模型確實能對網頁效能做改善，並且能成功預測出網路使用者行為下一步的路徑。

以下將以本研究所提出的網路服務組合塑模驗證及簡化的三個階段為基準，比較其他相關研究在每個階段是否有提出進行方法以及所提出的做法為何。從表4-6的比較可以發現，其他相關研究並未提出每階段的實際做法，大多僅是針對其中的一個或兩個階段提出方法。只有本研究完整的針對三個階段提出方法並且建立一個Petri-net為架構的預測模組實現網路服務塑模驗證及簡化的三階段。

表4-6 研究成果比較表

網路服務塑模、驗證及簡化三階段	Buhler & Vidal (2003)	Hamadi & Benatallah (2003)	Yi & Kochut (2004)	本研究
塑模	設計 BPEL4WS 標籤和Petri-net模型的轉換方式完成塑模	提出數種網路服務組合以Petri-net塑模的方式	設計 BPEL4WS 文件CP-net模型轉換	提出以 Petri-net塑模方式建立預測模型，並加入轉換控制矩陣作預測
驗證	無	無	有	有
簡化	無	提出網路服務組合的轉換規則但是轉換前後並未達到簡化的目的	無	有(參照表4-4)

(資料來源:本研究自行整理)