

驗證網路服務組合之服務品質的可靠性

A Validating Approach for Evaluating Reliability of Web Services Compositions

戚玉樑、李志偉

中原大學資訊管理研究所

桃園縣中壢市(32023)中北路 200 號

maxchi@mis.cycu.edu.tw, craigbaseball@yahoo.com.tw

摘要

網路服務(Web Services)是應用於系統整合的新興技術,目前已被採用在電子商務的架構上,然而一個網路服務的功能有限,因此未來的商業應用是以組合一群相關的網路服務為主。十餘種以 XML 為基礎的服務流程語言也在近年來紛紛推出,雖然它們在語法上稍有不同,但大致是以工作流程的觀點來建構服務組合。由於服務組合的成員通常是外來的,它代表流程整合者不一定能掌握成員的非功能性狀況,因此服務品質的問題亦成為近年來重要的研究議題之一。為確保服務組合的可靠性,本研究是針對服務組合在設計上的驗證為主,並以流程組合中的死結、可達性、及控制權移轉的安全性等三項問題為解決指標。因為服務組合須以 XML 格式呈現,故須轉換為等價的其他模型以利量化計算,我們因此選擇以 Petri Net 模型為標的,再分別配合適宜的分析方法驗證。研究過程中,我們克服了模型間轉換的問題,並銜接後續的各項演算法。最後,實證經驗顯示運用本研究的驗證方法,確可用於驗證服務組合在流程設計上的可靠性。

關鍵詞：網路服務、服務組合、工作流程、可靠性。

Abstract

This paper reports some experiences that we utilize Petri Net models to validate the reliability of a composite Web services. To enable a validate process, the PNML (Petri Net Markup Language) are introduced to transform a composite service into a Petri Net model that can be further used by measurable processes. The algorithms and corresponding reliable issues have been proposed, such as the coverability tree for detecting flow safeness, the incidence matrix & state equation for finding reachable issues, and a transitive matrix for detecting a deadlock problem. Empirical results suggest that the PNML is essential to facilitate composite services to become a computable model. Based on this model, more benefits can be gathered by using proper Petri Net based algorithms.

Keywords: Web services, Service composition, Workflow, Reliability

一、前言

網路服務是以訊息傳遞為基礎的新興系統整合架構，而構成這項架構的基礎元件，主要是既有的協定或由 XML 所衍生的規範組成，其進入的技術門檻相較傳統的中介軟體為低(例如 CORBA 及 COM+)，因此目前已被廣泛的應用在電子商務上。根據網際網路標準組織(W3C)對服務導向架構(SOA, Service-Oriented Architecture)的定義，網路上的角色可劃分為服務的需求者、提供者和仲介者等三種，另外 SOA 也衍生三項規範來詮釋各角色間的互動及作業依據，分別是：1).簡單物件存取協定(SOAP, Simple Object Access Protocol)，其用途是將訊息傳遞制式化，以完成服務的查詢、發佈、和執行；2).網路服務描述語言(WSDL, Web Services Description language)，用於記載與服務有關的相關資訊，諸如協議、關聯及執行細節；3).統一描述、發現、與整合協定(UDDI, Universal Description, Discovery, and Integration Protocol)，提供建立服務註冊中心所需的規範及服務提供者或需求者的使用依據，亦即規範了網路服務環境的仲介機制。簡言之，網路服務技術如同 Curbera et al.[12]所指出的要點：它利用 XML 來創造訊息的環境，各項相關技術已為應用系統在異質環境的作業中，提供了基本的運行架構。Vinoski[26]亦曾指出：一個網路服務的核心內容仍是一個傳統的應用程式(或系統)，但它應用了 WSDL 來描述它、透過 UDDI 註冊中心的發佈機制及查詢來仲介它、然後以類似 RPC(Remote Procedure Call)方式去執行及回應結果，而這些作業都必須是經過 SOAP 訊息(亦即 XML messaging)在網路環境下完成。

由於網路服務架構在需求及提供上均有別於以往的作業方式，因此使用上相對的彈性，然而單一網路服務的能力畢竟有限，因此在實務應用上多以組合多個服務成為另一個新的網路服務為主，以達成特定之商業目的。網路服務組合或簡稱為服務組合已廣泛的應用在各領域，例如 Alameh[3]透過服務的組合於地理資訊系統上，建立開放性的聯合服務介面，提供了更完整的地理圖像描繪功能給使用者。Medjahed et al.[18]曾以網路服務組合的方式，整合分散於各地的社會福利系統，達成單一服務窗口而減少重複作業的效益，其研究結論顯示：雛型的電子化政府確可藉服務組合的方式達成。Rabhi and Benatallah[22]則將服務組合應用於財務管理上，藉由聯合匯兌、交易資料、結帳和監控等組合，建構服務導向架構的資本市場系統。另外，還有更多在不同應用領域的研究結論也支持服務組合能創造更大效益的正面態度[3, 6, 10, 17, 25, 28, 29]。服務組合的流程描述方式，原則上須以 XML 為其語法格式，以利相容於網路服務的基礎架構。

服務組合由概念設計到執行雖然可視為一個工作流程，但兩者在本質上有些差異，例如傳統的工作流程中，組合的成員通常是已知的、處於特定的環境、且可選擇或替換性少的情況，因此整合者較能掌控成員的狀況。另一方面，服務組合中的成員，則是經由服務註冊中心來獲得，雖然來源更為多元化，但整合者僅能掌控成員的功能性狀況(例如執行細節或介面)，而非功能性狀態(例如品質或可靠性等)則無法獲得。Medjahed et al.[19]即指出服務組合須考量的重要問題是：它們都是外來的服務，因此其不確定性因素也相對提高。Ran[23]亦曾指出：目前在服務註冊中心登錄的服務，有高達 48%的服務並非處於堪用狀態，這些異常狀態並不能由服務查詢機制得知。另外，目前服務的流程語言僅涉及工作流的語法表達，但不能檢驗其服務組合是否可行。基於這些因素，一個設計完的服務組合是否能順利運作，其實處於高度的不確定狀態。

本研究的範圍雖涉及到服務在組合時的品質問題，但更將焦點置於組合的可靠性，以確保電子商業應用時的正確性。如同工作流程的實施，服務組合也是以規劃、定義和實作等三個階段進行，規劃階段是指由概念模型設計到尋找組合中的服務成員，定義階段則產生實際的組合結構，實作階段則根據服務組合的流程定義來實作細節。本研究的主要工作是在定義階段完成後及實作階段進行前，另外增加服務組合的結構驗證，並進一步歸納影響可靠性的三項主要因素：死結(Deadlock)、可達性(Reachability)、及控制權移轉的安全性(Safeness)等做為檢驗指標，其意義說明如下：

1. 死結：是指在服務組合的執行過程中，若因流程設計錯誤造成輸出入資料的衝突時，會導致某個成員一直無法啟動執行，則稱該服務組合發生死結問題，反之則稱該服務組合具備活性(或稱為"Liveness")。
2. 可達性：服務組合雖然是由多個網路服務共構而成，但概念上仍視同一個網路服務，故具備一組資料的輸入及輸出特性，因此在服務組合執行的過程中，資料輸出入的角色藉由流程而移轉，當流程結束時，必須到達最終的輸出位置，而此時其他成員不應擁有任何資料，否則代表該服務成員還未執行或執行完的結果還未輸出，滿足以上條件的服務組合稱之為具備可達之特性，反之則稱該服務組合為"Non-Reachable"。
3. 控制權移轉的安全性：類似可達性的資料輸出入問題，若在轉移過程中沒有出現任何重複狀況，亦即任何一個成員同時累積兩個或以上的資料輸出入，則稱該服務組合具備安全特性，反之則代表由於流程的設計錯誤造成控制權移轉不當的狀況，則稱該服務組合為"Non-Safe"。

目前服務組合是以流程語言來描述，由於它們是以 XML-based 的文字格式呈現，故無助於上述檢驗的實施，因此我們須要一個可轉置的中介模型為基礎，以利銜接後續的量化計算。本研究選擇以 Petri Net 模型為中介轉置媒介，並配合適宜的分析及演算法執行驗證，故本研究待克服的瓶頸如下：1).如何將服務組合的流程語言轉置為一個等價且可分析的 Petri Net 模型；2).如何將各檢驗指標在 Petri Net 模型下有效的驗證；3).如何將 Petri Net 模型還原回一個等價的服務組合流程語言。

二、文獻探討

2.1 服務組合之目的及效益

服務組合的目的在因應單一網路服務功能不足時，必須藉由組合來建立新的服務，提供更完備的功能。Chandrasekaran et al.[10]曾指出當個別的服務受限於它們所提供的功能時，我們必須以流程的形式去組合現存的服務，以創建新的整合功能。Lehmann[16]提出服務組合須考量實際商業效益，例如提供加值性的整合來達成企業結盟、服務的延伸及再利用、增加線上網路服務的數量、及支援 e 化組合的規劃和實作。Marton et al.[17]的研究曾定義服務組合須結合工作流程的概念為其設計及執行的基礎。Benatallah et al.[4]曾以商業上的實際應用來說明服務組合的必要性，他們以企業流程為藍圖來規劃服務組合的順序，定義了組合須擁有控制流程和資料流程的規格，並以這些規格來描述服務組合該如何進行。Zeng et al.[29] 亦指出整合分散式或異質平台上的服務，組合是有效的方式之一，而服務彼此互動的邏輯常以流程模型的方式表達。他們另一份研究[30]

也對服務組合的結構加以分析，並將組合的類型歸納為三種：基礎網路服務、組合的網路服務、及網路服務社群，透過這三種成員的結合，可以產生具有複雜功能的網路服務，同時社群的設計也可以達到動態網路服務的組合目的。

就服務組合的效益而言，VanderMeer et al.[25]提出服務入口的概念，如果一個系統可以提供產生和執行服務組合的能力，其價值明顯超過單一網路服務所提供的價值。Zeng et al.[30]指出以流程為基礎的服務組合，是一種新興的組織內和跨組織的企業自動化流程方法，因為流程邏輯可將個別的服務予以組合成新的網路服務。在傳統的工作流程管理系統中，流程邏輯大多是靜態且事前定義的，但對於擁有眾多替代方案的企業流程來說，是不切實際或不能因應快速變動的需求，因此服務組合的效益在於設計上的相對彈性。

2.2 服務組合的方式與流程樣板

根據 Aalst et al. [2]的調查，目前用來表達服務組合的流程語言已有十餘種，大多數是以 XML 為基礎語法並結合工作流概念，提供服務整合者利用，由於服務組合的流程語言並沒有統一的標準，因此以提供執行引擎的流程語言較為普及。Alast[1]曾比較近年來常用的流程語言，例如 WSFL、XLange、BPEL4WS、BPML 等，依據工作流程管理聯盟(WfMC)[27]對流程結構的基本需求，比較六大類共計二十種樣板(patterns)，其結果顯示沒有任一種流程語言可以達到全部項目，例如較突出的 BPEL4WS 也僅達到 14 項，但調查結果也顯示各種流程語言均達到基本流程結構的需求，例如循序、合併等。

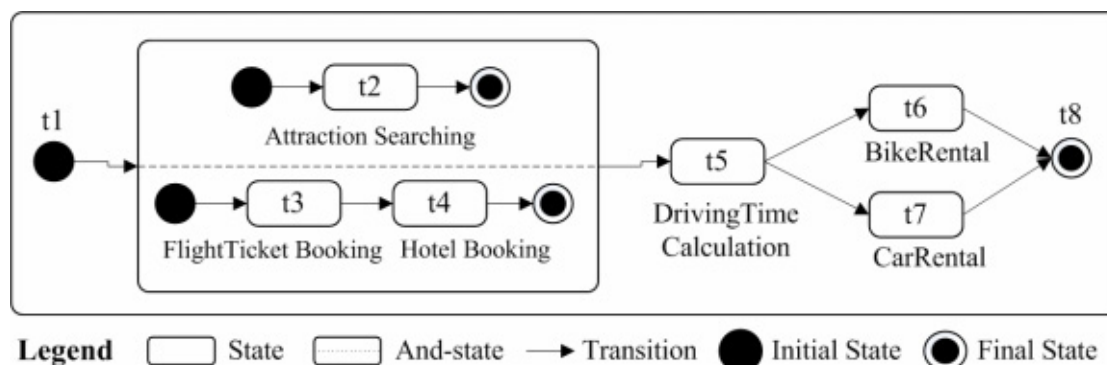
目前服務組合的另一項研究則是建置服務組合的流程整合開發平台，例如 Benatallah et al.[6]建立名為 Self-Serv 的專案，該平台是以服務入口的概念，利用圖形介面提供使用者建置流程的整合平台；VanderMeer et al.[25]的研究，曾建立名為 FUSION 的系統，但僅以點對點為基礎的服務協同模型；Casati and Shan[9]建立名為 eFlow 的專案，該研究嚐試以適性化的角度，從事動態的網路服務組合；Chandrasekaran et al.[10]則建立名為 SCET 的專案，以 IBM WSFL 為流程語言建立服務組合模型，並模擬服務組合執行的效能分析及建立回覆機制。上述的研究均以建置整合開發平台為標的，提供流程執行後的效能分析，但未提供服務組合在非功能性問題的驗證。

2.3 服務組合的流程塑模類型

在工作流程的研究中，為了能夠定義流程的設計並進行後續的模擬分析，常藉由塑模工具來表達流程的結構設計，其中又以狀態圖(State-chart)和 Petri Net 最常見，而它們也被許多研究用來塑模服務組合，以下分別就這兩種塑模工具在網路服務組合上的應用進行探討。

1. 以狀態圖塑模服務組合：狀態圖是傳統的塑模工具，以工作流程的應用而言，它是由狀態(State)和控制權移轉(Transition)所構成，除擁有定義的語法外並且包含流程結構樣板，因此可用來呈現服務的組合。例如 Zeng et al.[30]使用狀態圖來塑模網路服務的組合問題(參考圖一)，其中狀態並分為基礎(Basic)和混合(Compound)兩類，基礎狀態表示單一的網路服務，混合狀態則代表多個網路服務以 OR(包含多個選擇性執行的服務狀態圖)及 AND(包含多個同時執行的服

務狀態圖)組合而成。圓形點代表控制權的轉移，用於陳述服務組合流程中的事件、條件、或指派的操作。



圖一：以 State-chart 塑模網路服務組合的流程

- 以 Petri Net 塑模服務組合：Murata[21]指出 Petri Net 為一具有圖形特性與數學理論基礎的塑模工具，不僅能分析模型內的同步與互斥行為，另可對系統做定性與定量的分析，Petri Net 也陸續延伸出許多其他特性的衍生理論，以加強對不同類系統的塑模能力。Petri Net 塑模常以建立系統狀態的方程式、代數、及其他用於管理系統行為的數學模組表示，因此 Petri Net 塑模亦可用於服務組合的研究。例如 Vidal et al.[8]為了在多代理人的工作流程系統中建立網路服務之間的關係，將網路服務組合語言(使用 BPEL4WS)另以 Petri Net 的模型重新建立，利用圖形化方式在特定的 Petri Net 平台上進行分析。Hamadi and Benatallah [13]則提出一套服務組合的 Petri Net 塑模方法稱之為 Service Net，該研究在塑模設計時將服務的操作細節也一併納入，以建立可靠的網路服務組合為目的，惟當服務組合複雜時，因細節過多而須考量此分析方式的可行性。Yi and Kochut [28]則是為了能夠在服務組合設計階段透過驗證儘早發現錯誤並更正，以避免在執行階段才因錯誤而中斷，因此提出了一個以 Color Petri Net(CPN)為基礎的網路服務組合設計及驗證架構，將 BPEL4WS 文件中描述服務組合流程的部份以 CPN 塑模，該研究的分析方式也是須分別建立服務組合流程及 CPN，並在專屬的分析平台上進行。

2.4 Petri Net 與其結構分析方法

根據 Murata[21]的研究，一個 Petri Net 用於塑模工作流程時，可由五個元件進一步表達成如 $PN=(P, T, F, W, M)$ 的通式，其中各變數的意義如下：PN 代表通用的 Petri Net 模型；P 代表 Place，表示資源的狀態、條件、或緩衝或儲存空間的集合，故 $P=\{p_1, p_2, \dots, p_n\}$ ；T 代表 Transition，表示作業、過程、活動、或事件的集合，故 $T=\{t_1, t_2, \dots, t_n\}$ ；F 代表流程(arc)，表示流程關係的集合，流程關係集合可寫成 $F \subseteq (P \times T) \cup (T \times P)$ ；W 代表 Place 的權重函數；M 則用於標示 Petri Net 的階段狀態。利用 Petri Net 代數式來塑模工作流程之目的，是為銜接各種計量的分析方法，藉以察覺工作流程潛在的結構及行為問題，其中包括本研究所需的可達性、限制性、活性等，表一說明相關分析項目在 Petri Net 及服務流程中的意義。

表一：Petri Net之結構及行為分析對照

分析項目	Petri Net中的意義	服務流程中的意義
可達性 (Reachability)	若網路中任一狀態可藉由有限的激發序列將系統狀態變成 m ，則 m 對於狀態是可達的。	服務流程執行的結果能順利傳送到輸出的位置且只有一個輸出
限制性 (Boundedness)	若Petri Net模型中任一基態內所儲存的標記數恆小於或等於固定常數 k ，則稱該基態是K-safe或K-bounded。	無容量或發生緩衝、儲存空間不足的情形
活性(Liveness)	當Petri Net中之所有狀態皆為可激發，且不因曾被激發就不再具有可激發之特性時，稱此Petri Net具有活性。	流程中的每一服務都有可能被執行

在過去的研究中，已有許多針對上述問題的偵測或解決方法，我們彙整其中一些方法如下：(1).Karp and Miller[15]提出的 Coverability Tree，此方法將 Petri Net 模型內的狀態轉移過程，以樹狀結構的方式呈現，並追蹤 Petri Net 模型狀態的變化，因此可以再藉狀態轉移過程，分析如可達性、限制性、活性等問題，此方法的缺點在於模型複雜時，樹狀結構將過於龐大而難以分析，因此較適用於小型的模型或僅監測狀態轉移過程。(2).Murata and Church [20]提出的 Incidence Matrix and State Equation 方法，它是將 Petri Net 模型以矩陣的方式表達，再透過計算方式來預測狀態之間是否可以彼此轉移，此方法雖適用於較複雜的模型，但是僅針對模型的可達性做分析。(3).Corbett[11]針對 Petri Net 的死結問題提出幾項可行的演算法，其後繼研究由 Song and Lee[24]提出一個 Transitive Matrix 方法，此方法運用標記位置的遞移矩陣 L_{DP}^* 及 $M_i = M_{i-1} \cdot L_{DP}^*$ 公式，逐步推演模型的每一個狀態，以驗證表一的各性質，比較特別的是除了能發現死結問題外，還提供了解除的方法。

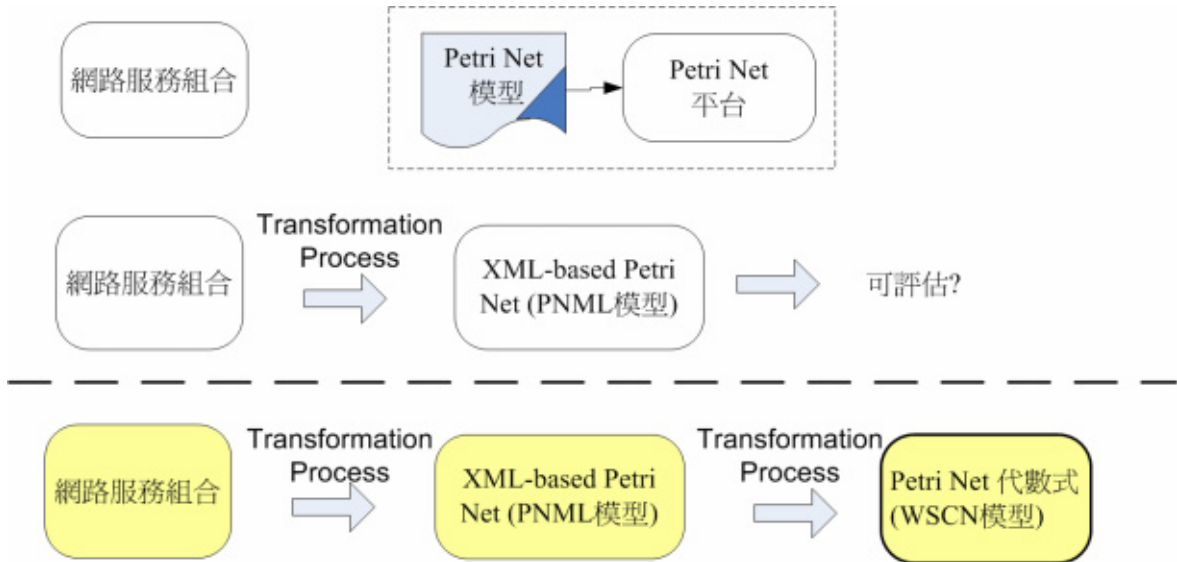
三、前置驗證的研究設計

為與現階段服務組合的開發程序相結合，本研究在定義階段後及執行階段前，加入服務組合的流程驗證程序，對未通過驗證程序的組合個案，則要求必須回到定義階段重新設計。我們提出的核心設計包含兩項：組合流程模型間的轉換及分析指標的驗證。

3.1 組合流程模型間的轉換設計

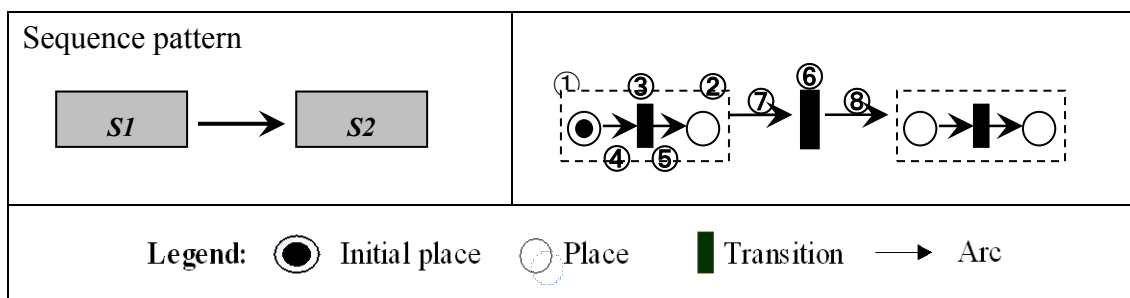
服務組合一般是由 XML-based 的流程語言表示，因語言的文字形式無法滿足進一步的分析需要，故須轉換服務組合至其他可評估的塑模格式。過去關於服務組合與 Petri Net 的研究，是以重新建置等價的組合模型為其程序，因此實際的流程模型與分析用的模型是各自獨立，而分析工作也須在特定的塑模環境下進行。本研究對組合模型間的轉換，則期望能接續原有的服務組合之架構及限制在同一平台上直接處理。本研究共須提供兩次的模型轉換設計，圖二說明須要兩次轉換的原因，圖中的第一列表達單純利用 Petri Net 來評估服務流程，因為塑模方式不同，故須分別在各自的環境下執行。第二列則是將原 Petri Net 的塑模改為利用 XML 表示，其目的為化解前述格式不同所造成須各

自獨立運作的問題，服務組合流程語言與 Petri Net 之間可利用 PNML(Petri Net Markup Language)來完成，PNML 是針對 Petri Net 所發展的 Meta Language，也是以 XML 為基礎的語言，因此一個 Petri Net 模型亦可利用 PNML 充份表達[7]。為將 PNML 內的資料提供至 Petri Net 演算法，我們仍需要轉換成 Petri Net 的代數式，圖二的第三列即完整列出本研究所需的轉換階段。

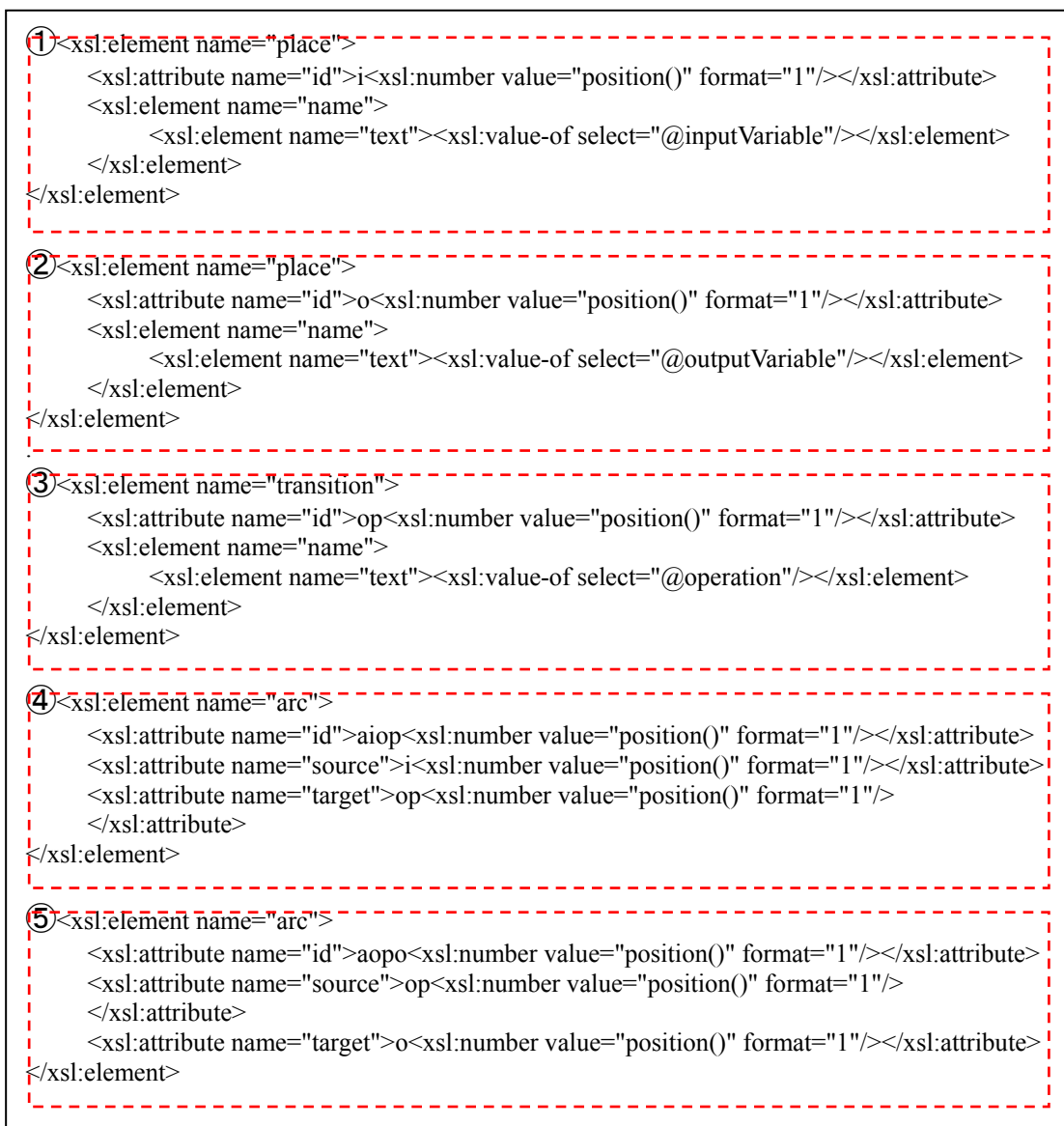


圖二：網路服務組合在模型間的轉換概念

確立圖二的模型轉換設計概念後，後續的設計進行則以訂定兩個轉換器之細節為主，並分別包裹為代理人方式納入系統。由於服務流程語言內含許多不同的工作流程樣板，而各樣板自成一組特定區塊，因此轉換時並不適合以直接對映方式轉換為 PNML 的文件。我們在轉換器的設計於是改成下列步驟：(1).透過特定流程語言的解譯並將流程樣板予以分解；(2).依整理好的流程樣板，撰寫通用的 XSLT 對映規則；(3).將轉換的流程樣板予以組裝成爲 PNML。圖三是以序列樣板(Sequence pattern)為例，左側是”服務 1”完成後才至”服務 2”的示意，右側是以 Petri Net 模型表達。我們標示①~⑧來說明：就單一的網路服務之 Petri Net 表示而言，其組成須對映至圖三右側所標示①~⑤，依序號分別是兩個 place、一個 transition、及兩個 arc。因此若將一個網路服務對映為 PNML，其規則須依這五項的次序進行，圖四是我們所編寫的 XSLT 對映規則，也分別以①~⑤的序號，區別出一個網路服務表達為 Petri Net 的 XML 格式。



圖三：Sequence pattern 在服務流程與 Petri Net 的示意



圖四：單一的網路服務轉換為 PNML 的對映規則

圖三的流程樣板 Sequence，若以 BPEL4WS 為例，可寫成如圖五的描述方式，但若
 要轉置為 PNML 格式，則仍須圖三右側 Petri Net 模型內的標示⑥~⑧，因此本圖可再依
 序解讀為：一個 input arc 以序號⑦標示、一個 transition 以序號⑥標示、output arc 以序
 號⑧標示、及最後的服務二(虛線框)。圖六是將圖三 Petri Net 模型序號⑥~⑧，利用 XSLT
 的對映規則表示。

```

<sequence>
  <invoke partner="S1" operation="ProcessSRL" inputVariable="newsRequest"
    outputVariable="newsResponse"/>
  <invoke partner="S2" operation="BabelFish" inputVariable="englishNews"
    outputVariable="translatedNews"/>
</sequence>
    
```

圖五：Sequence pattern 在 BPEL4WS 的陳述


```

⑥<xsl:element name="transition">
  <xsl:attribute name="id">t<xsl:value-of select="position()"/></xsl:attribute>
  <xsl:element name="name">
    <xsl:element name="text">Sequence<xsl:value-of select="position()"/></xsl:element>
  </xsl:element>
</xsl:element>

⑦<xsl:element name="arc">
  <xsl:attribute name="id">aot<xsl:value-of select="position()"/></xsl:attribute>
  <xsl:attribute name="source">o<xsl:value-of select="position()"/></xsl:attribute>
  <xsl:attribute name="target">t<xsl:value-of select="position()"/></xsl:attribute>
</xsl:element>

⑧<xsl:element name="arc">
  <xsl:attribute name="id">ati<xsl:value-of select="position()"/></xsl:attribute>
  <xsl:attribute name="source">t<xsl:value-of select="position()"/></xsl:attribute>
  <xsl:attribute name="target">i<xsl:value-of select="position()+1"/></xsl:attribute>
</xsl:element>

```

圖六： Sequence pattern 對映至 PNML 的 XSLT 規則

製作 XML 文件之間的轉換規則是一項繁鎖的工作，前面圖三的流程樣板示意圖，在轉換時須結合圖四及圖六的轉換規則，最後獲得如圖七的 PNML 文件。為使服務流程語言與 PNML 間的轉換對映順利進行，本研究認為應至少完成基本的流程樣板的轉換規則設計，另外藉由基本的樣板可以再組成其他不同的樣板。

```

<place id="i1"><name><text>newsRequest</text></name></place>
<place id="o1"><name><text>newsResponse</text></name></place>
<place id="i2"><name><text>englishNews</text></name></place>
<place id="o2"><name><text>translatedNews</text></name></place>
<transition id="op1"><name><text>ProcessSRL</text></name></transition>
<transition id="op2"><name><text>BabelFish</text></name></transition>
<transition id="t1"><name><text>Sequence1</text></name></transition>
<arc id="aiop1" source="i1" target="op1">...</arc>
<arc id="aopo1" source="op1" target="o1">...</arc>
<arc id="aiop2" source="i2" target="op2" >...</arc>
<arc id="aopo2" source="op2" target="o2" >...</arc>
<arc id="aot1" source="o1" target="t1" >...</arc>
<arc id="ati1" source="t1" target="i2" >...</arc>

```

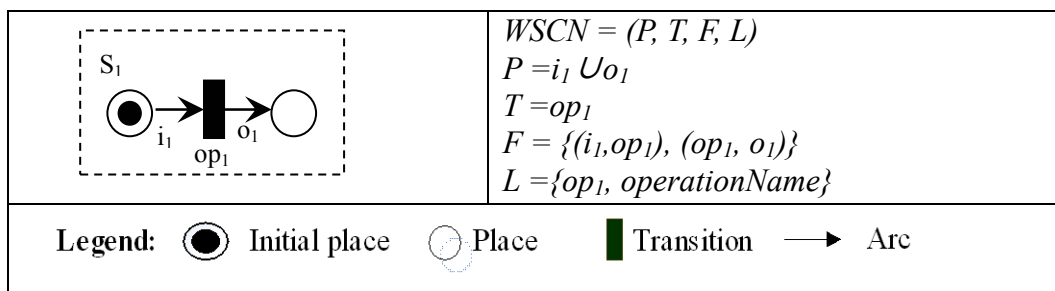
圖七： Sequence Pattern 的 PNML 文件

一個服務組合一旦轉換成爲 PNML 格式後，我們便可視爲獲得了一個 Petri Net 模型，但 PNML 仍須進一步將 XML 格式換成代數式以利結合演算法分析的需要。我們考量後續分析所需的元素，並融合了 Murata[21]及 Hamadi and Benatallah[13]對 Petri Net 的定義式，重新提出一個 Petri Net 的塑模式，並命名爲 WSCN(Web Services Composition

Net)，WSCN 主要是藉代數式描述 place、arc、及 transition 間的關係，並以 4-tuple 寫成 $WSCN = (P, T, F, L)$ ，其定義說明如下：

- P ：代表有限的 place 集合，place 是指網路服務的狀態。
- T ：代表有限的 transition 集合，transition 則代表網路服務的作業。
- F ：代表狀態與作業的流向關係(arc)，可以代數式表達為 $F \subseteq (P \times T) \cup (T \times P)$ 。
- L ：代表有限的功能集合，可以代數式表達為 $T \rightarrow N \cup \{\emptyset\}$ ，其中 N 代表網路服務操作的名稱， \emptyset 代表流程結構上的操作並非實際的網路服務操作。

一個網路服務以 Petri Net 塑模後的成員，可利用圖形展示於圖八左側，其中各符號的意義如下： S 及其序號用於區分不同的網路服務； i 及其序號代表網路服務接受的輸入訊息(input place)； o 及其序號代表網路服務的輸出訊息(output place)； op 及其序號代表網路服務內部所執行的操作亦即 transition，網路服務之間的 op 則以不同的符號 t 及其序號代替，亦表示 transition 的意義。根據以上的符號及前述定義，一個網路服務以 WSCN 定義的陳述式如圖八右側。

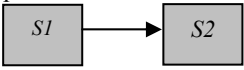
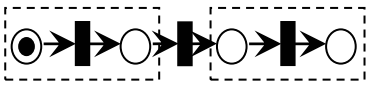
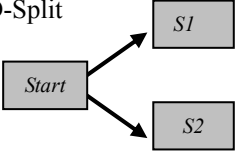
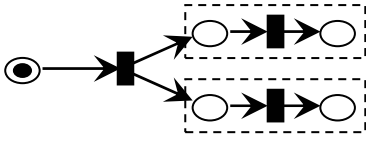
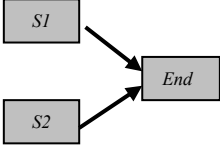
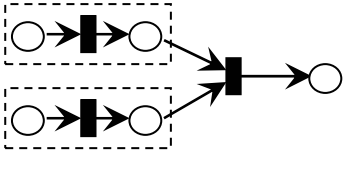
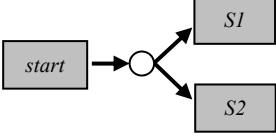
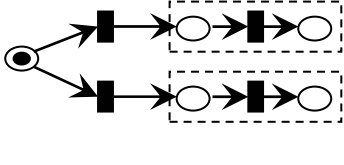
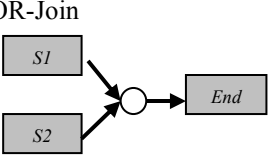
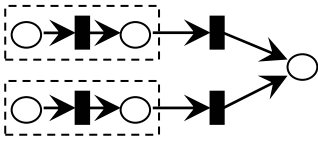
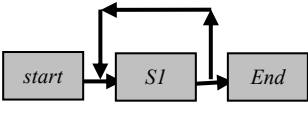
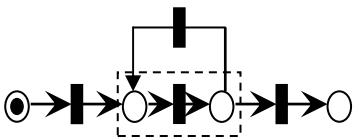






圖八：網路服務的 Petri Net 塑模及其 WSCN 定義

本研究將常用的六項基礎流程樣板，以流程樣板名稱、Petri Net 模型、及 WSCN 塑模式分別列示於表二，各流程樣板說明如下：

1. **Sequence**：也稱為 sequential routing 或 serial routing，是最常見的流程結構，用於安排流程執行序列。Petri Net 是以一個 transition 串連第一個網路服務的 output place 和第二個網路服務的 input place 來表達執行序列。
2. **AND-Split**：也稱為 parallel -split 或 fork，用於同時啟動數個網路服務，使數個網路服務平行執行。Petri Net 是以 transition 連接兩個或以上的網路服務之 input place，來表達同時及平行的概念。
3. **AND-join**：也稱為 synchronization 或 rendezvous，用來同步化數個平行執行的網路服務，亦即等待前面所有的服務完成後，才可以執行下一個服務或到達下一個狀態。
4. **XOR-split**：也稱為 exclusive choice 或 switch，流程結構根據決策條件或服務的參數資料，選擇啟動其中一個服務。
5. **XOR-join**：也稱為 asynchronous join 或 merge，流程結構等待前面服務的訊息，只要其中有一個成功，就可以執行下一個服務或到達下一個狀態。
6. **Iteration**：也稱為 loop 或 cycle，代表一個或多個服務可以重覆執行，直到符合某一個條件才執行下一個服務或到達下一個狀態。

表二：流程樣板與 WSCN 塑模式對照表

流程樣板	Petri Net 模型	WSCN 塑模式
Sequence 		$P = P_1UP_2$ $T = T_1UT_2U\{t\}$ $F = F_1UF_2U\{(o_1, t), (t, i_2)\}$ $L = L_1UL_2U\{\phi\}$
AND-Split 		$P = P_1UP_2 U \{i\}$ $T = T_1UT_2U\{t_i\}$ $F = F_1UF_2U\{(i, t_i), (t_i, t_1), (t_i, t_2)\}$ $L = L_1UL_2U\{t_i, \phi\}$
AND-Join 		$P = P_1UP_2 U \{o\}$ $T = T_1UT_2U\{t_o\}$ $F = F_1UF_2U\{(o_1, t_o), (o_2, t_o), (t_o, o)\}$ $L = L_1UL_2U\{t_o, \phi\}$
XOR-Split 		$P = P_1UP_2 U \{i\}$ $T = T_1UT_2U\{t_{i1}, t_{i2}\}$ $F = F_1UF_2U\{(i, t_{i1}), (i, t_{i2}), (t_{i1}, i_1), (t_{i2}, i_2)\}$ $L = L_1UL_2U\{(t_{i1}, \phi), (t_{i2}, \phi)\}$
XOR-Join 		$P = P_1UP_2 U \{o\}$ $T = T_1UT_2U\{t_{o1}, t_{o2}\}$ $F = F_1UF_2U\{(o_1, t_{o1}), (o_2, t_{o2}), (t_{o1}, o), (t_{o2}, o)\}$ $L = L_1UL_2U\{(t_{o1}, \phi), (t_{o2}, \phi)\}$
Iteration 		$P = P_1UP_2U\{i, o, p_1, \dots, p_5\}$ $T = T_1UT_2U\{t_0, t_1, \dots, t_4\}$ $F = F_1UF_2U\{(i, t_1), (t_i, p_1), (t_i, p_2), \dots, (p_5, t_0), (t_0, o)\}$ $L = L_1UL_2U\{(t_i, \phi), (t_1, \phi), \dots, (t_0, \phi)\}$
Legend:  Initial place  Place  Transition  Arc		

3.2 驗證方法設計

Petri net 以 WSCN 轉置後，便可以運用分析方法模擬流程狀態的進行，本研究結合死結、可達性、及控制權移轉的安全性等三項問題，做為服務組合可靠性的驗證指標，控制權移轉的安全性問題將利用 Karp&Miller[15]提出的 Coverability Tree 方法；可達性問題將利用 Murata and Church[20]提出的 Incidence Matrix and State Equation 方法；死結問題則是利用 Song and Lee[24]提出 Transitive Matrix 方法。一個服務組合必須通過以上的三項驗證，才可視為具可靠性的流程，以下我們分別說明各驗證方法在 WSCN 塑模式的設計與意義：

1. Coverability Tree 方法：在 WSCN 塑模中，token 代表網路服務組合需求的輸出入，合格的服務流程中，至多僅有一個輸出入，因此必定為 1-bounded Petri Net 否則即判定為不安全。根據本演算法對於一個 $Petri\ Net(N, M_0)$ 的分析過程如下：

- (i). 將一個 Petri Net 的起始狀態 M_0 ，設為樹的起源並且標記為 M_{new} 。
 - (ii). 選擇一個新的狀態 M ，如果它的狀態等同於從樹的起源到狀態 M ，則標記狀態為 M_{old} ，並且進行另外一個新狀態，如果在狀態 M 時沒有任何 transition 可以被激發，則標記狀態為 $M_{dead-end}$ 。當狀態 M 有 transition(t)可被激發時，則藉由 t 激發獲得新的狀態 M' ，如果存在另一個狀態 M'' ，對於在狀態 M'' 中的每一個狀態(p)，不同於狀態 M' 的情形下，若能滿足 $M'(p) \geq M''(p)$ ，則以變數 ω 替換 $M'(p)$ 中的狀態，使得 $M'(p) > M''(p)$ 。
 - (iii). 將狀態 M' 視為樹的一個節點，從狀態 M 畫一個弧線到狀態 M' 並標記為 M'_{new} 。因此，在實作上要驗證 Petri Net(N, M_0) 是否為安全的流程，其關鍵在檢驗樹狀結構中每一個節點狀態 $M(p)$ 的標記，每一個節點狀態的標記只能出現 0 或 1 的情形，若出現 2 或 2 以上的狀態標記則為不安全。
2. Incidence Matrix and State Equation 方法：將 Petri Net 模型以輸入矩陣 D 和輸出矩陣 D^+ 的方式分別表達，每個矩陣皆擁有 m 列和 n 行，分別代表 Petri Net 模型中不同的 transition 和 place。輸入矩陣 D 代表輸入到 transition 的 place，表示為 $D[i,j]=\#(p_i, I(t_j))$ ；其中 $I(t_j)$ 為 0 或 1，若 p_i 具有一流程路徑輸入至 t_j ，則 $I(t_j)$ 為 1，反之則為 0。輸出矩陣 D^+ 代表由 transition 輸出的 place，表示為 $D^+[i,j]=\#(p_i, O(t_j))$ ；其中 $O(t_j)$ 的值為 0 或 1，若 t_j 具有一流程路徑輸出至 p_i ，則 $O(t_j)$ 為 1，反之則為 0。而代表整個 Petri Net 系統模型的矩陣 D ，可以由以下的式子獲得 $D = D^+ - D^-$ 。舉例而言，若服務組合僅單純如流程樣板的 XOR-join，則 $D = D^+ - D^-$ 的意義如下：

$$D^+ = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad D^- = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad D = \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \end{bmatrix}$$

本項方法分兩階段實施，在利用 Incidence Matrix 獲得完整的流程矩陣模型後，須繼續下一階段的狀態驗證(State equation)，亦即檢查起始狀態 M_0 是否可以到達結束狀態 M_n 。將狀態 M_0 和 M_n 分別以 $1 \times n$ 的矩陣方式表達(n 代表 place 的個數)，狀態 M_0 和 M_n 的矩陣表達方式，分別如 $M_0 = [1 \ 0 \ 0 \ \dots \ 0]_{1 \times n}$ 及 $M_n = [0 \ 0 \ 0 \ \dots \ 1]_{1 \times n}$ ，我們再以下列的計算式 $M_n = M_0 + x \cdot D$ 來求解。若 x 有解則代表起始狀態 M_0 可以到達結束狀態 M_n ，反之則代表起始狀態 M_0 不能到達結束狀態 M_n 。

3. Transitive Matrix 方法：本分析方法類似前面的作法，也是將 Petri Net 模型以輸入矩陣 D 和輸出矩陣 D^+ 的方式表達，其中各矩陣的內容也分別是 $D[i,j]=\#(p_i, I(t_j))$ 及 $D^+[i,j]=\#(p_i, O(t_j))$ 。另外，我們利用遞移矩陣 L_{DP} 將 Petri Net 的 place 狀態予以集合，亦即遞移矩陣 L_{DP} 的元素，是用於描述 token 從一個 place 透過 transition，轉移到另一個 place 的轉換關係。因此，遞移矩陣可表達為 $L_{DP} = D \cdot \text{diag}(t_1, t_2, \dots, t_n) \cdot (D^+)^T$ ，其中 $t_i (i=1, 2, \dots, n) \mid t_i = \begin{cases} 1 & \text{fire } t_i \\ 0 & \text{not fire } t_i \end{cases}$ ，亦即 1 為激發。若同一行中的 t_k 出現 s 次，則以 (t_k/s) 取代原本的 t_k ，並且形成一個 place 的新標記之遞移矩陣 L_{DP}^* 。經由遞移矩陣的運算可以評估 transition 是否可以被激發，並且計算 transition 的序列和數量。在得到最終遞移矩陣 L_{DP}^* 後，須利用 $M_i = M_{i-1} \cdot L_{DP}^*$ 運算式，驗證是否會出現死結：
 - (i). 首先以 $i=1$ 代入 $M_i = M_{i-1} \cdot L_{DP}^*$ 運算式，得到 $M_1 = M_0 \cdot L_{DP}^*$ 。例如初始狀態是

$M_0=[1\ 0\ \dots\dots\ 1\ 1]$ ，它實際代表各 place $[p_1\ p_2\ \dots\dots,\ p_n]$ 的 token 狀態，亦即每一個 place 是否含有 token(以 1 表達有 token，0 則沒有)。

- (ii). 經由 $M_1= M_0 \cdot L_{DP}^*$ 運算後所得到的初始狀態，若某個 place 的 $\sum(t_{ki}/s_i)$ 大於等於 1，則代表該 *transition*(t_k) 可以被激發，例如

$$M_1=[0, (p_1(t_1/2)+p_8(t_1/2)), p_7(t_2/2), 0, (p_5(t_4/2)+(p_7(t_4/2))), p_8(t_5/2), p_8(t_5/2), 0]$$

則代表 t_1 和 t_4 可以被激發。

- (iii). 將 M_1 予以正規化，亦即具有可激發的 place 以 1 取代，反之以 0 取代， M_1 因此轉換為 $M_0=[0\ 1\ 0\ 0\ 1\ 0\ 0\ 0]$ 。

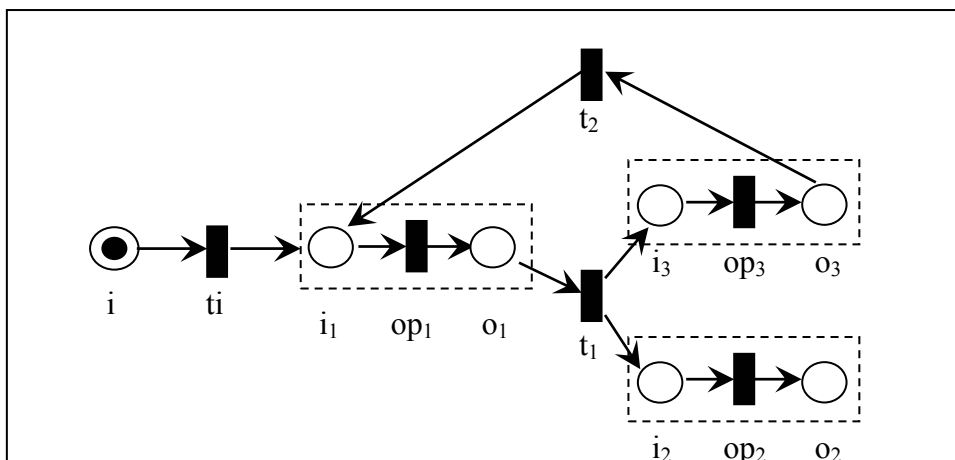
- (iv). 再回到步驟(i)，以 $i=2$ 代入 $M_i= M_{i-1} \cdot L_{DP}^*$ ，得到 $M_2= M_1 \cdot L_{DP}^*$ ，重覆以上的演算步驟直到 $i=n$ 。

四、驗證實作

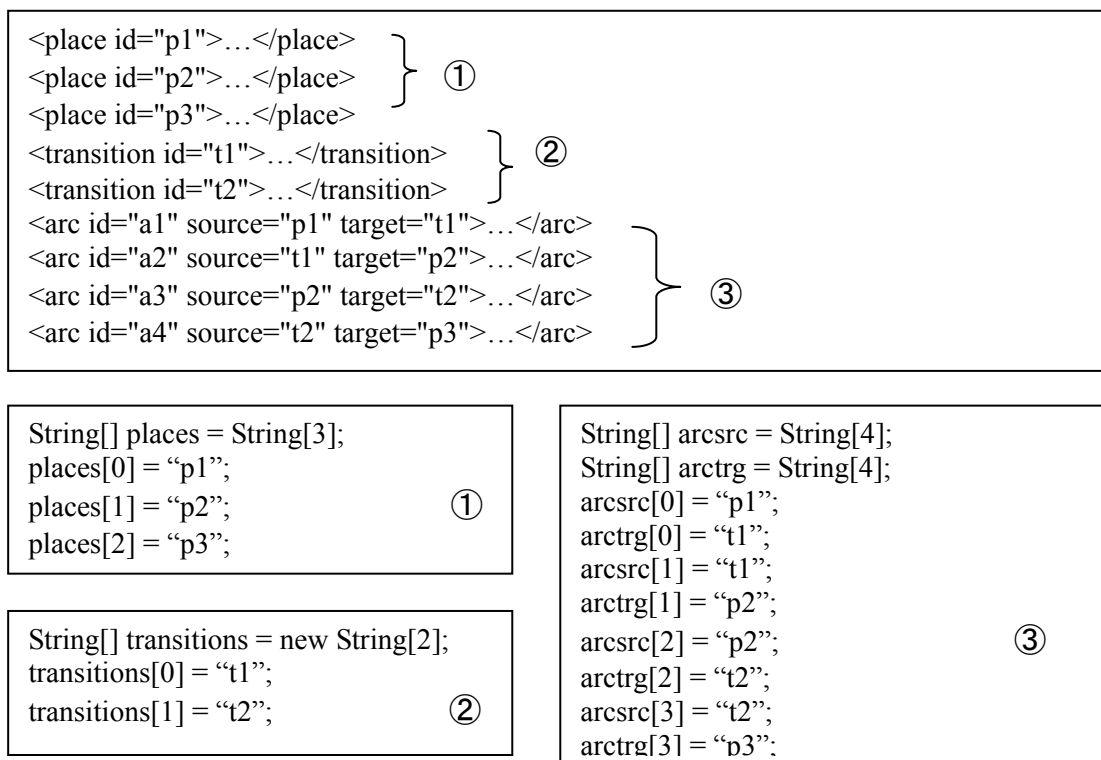
4.1 流程模型間的轉換

本研究實證將採用的服務流程語言是 BPEL4WS(或簡稱為 BPEL)[14]，它是由 IBM、Microsoft 和 BEA 所共同發表，BPEL 結合了由 IBM 之前所研發的服務流程語言 WSFL 及 Microsoft 為 BizTalk 所開發的商業流程語言 XLANG，因此 BPEL 是目前較常用服務組合語言之一。本研究透過研究設計對流程模型轉換的設計，製作了 BPEL to PNML 轉換代理人，這個代理人的組成如下：

- BPEL Parsing and Decomposer：將 BPEL 文件載入並予以解譯，再將組合流程中的特定標籤例如 sequence、flow、switch 等逐一拆解，並對映儲存至不同文件中。
- Translator：依據拆解後的 BPEL 文件，利用已設計好的 XSLT 規則對映成 PNML 的流程樣板。
- PNML Composer：由於轉換後的流程樣板仍是各自獨立，因此本階段的組合器會根據原始 BPEL 文件所描述的流程邏輯，重新組成一份完整的 PNML 文件。
- WSCN Parser：用於解析 PNML 文件中的 Petri Net 模型，將 place、transition 和 arc 以陣列方式儲存，亦即根據我們設計的 Petri Net 塑模式 $WSCN = (P, T, F, L)$ 表達。



圖九：流程控制權轉移安全性的測試範例



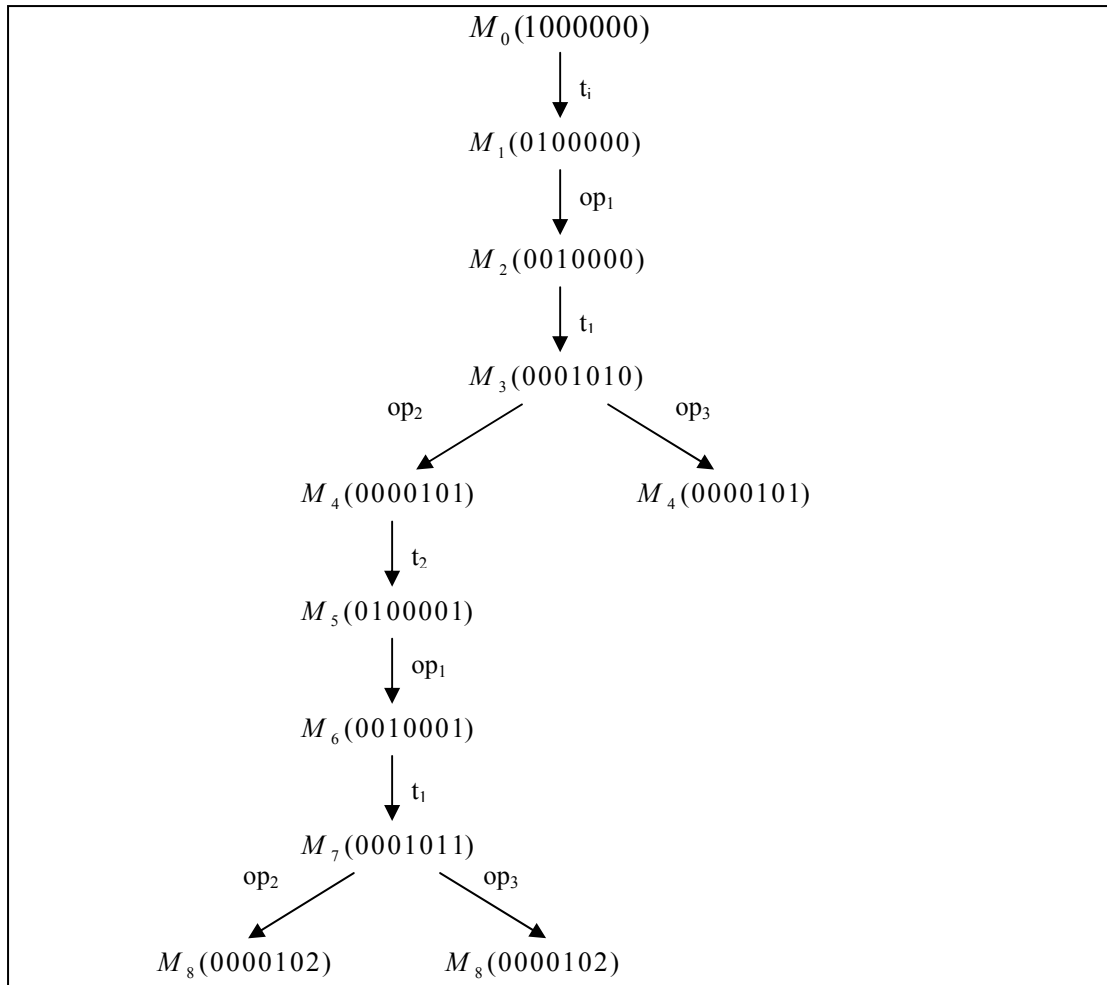
圖十：以陣列型態將 PNML 儲存為 WSCN

在最後的 WSCN 解譯步驟中，我們以圖九的 Petri Net 模型為例，其中 *i*, *o*, *t*, 及 *op* 符號的意義分別是：input place, output place, transition, 及 operation，它們是驗證分析所須要的參數，PNML Parser 以 WSCN 對各流程樣板的定義，將各參數以陣列型態儲存為物件。圖十上方即為以 PNML 描述圖九 Petri Net 的模型，圖十下方的三個區塊則分別是儲存 place, transition, 及 arc 的陣列物件，這些物件將以參數方式傳遞給後續驗證者使用於代理人中。

4.2 可靠性驗證

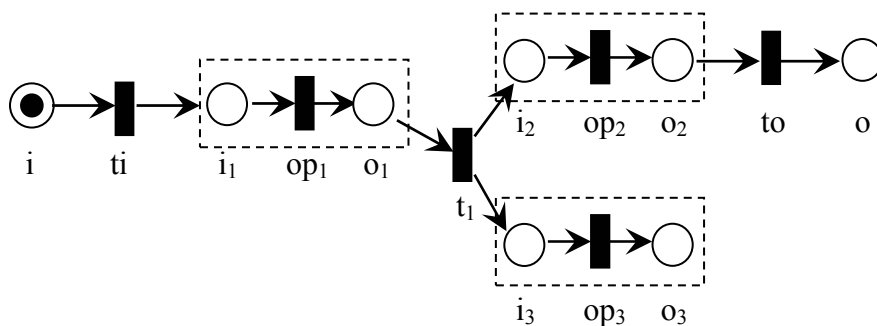
本研究的可靠性驗證是以代理人的形式實施，並將研究設計中的三項分析方法予以納入，驗證程序是由較容易的控制權轉移安全性開始，其次是可達性，最後是死結偵測。驗證過程中，只要有一項無法通過就會中止整個驗證程序。為能逐一說明本研究的驗證成效，我們將分別以有瑕疵的服務組合設計來說明，各驗證程序如下：

1. 控制權轉移安全性：以圖九的 Petri Net 模型為例，利用 Coverability Tree 演算法逐步推演各狀態的進行。如圖十一所示，其中狀態為 M_x ，假設 $x=1,2,\dots,8$ 表示狀態的演進。狀態 M_x 的內容記為 $M_x(t_k(i), t_k(i_1), t_k(o_1), t_k(i_2), t_k(o_2), t_k(i_3), t_k(o_3))$ ，其中 transition $t_k(i)$ 或 $t_k(o)$ 代表該 place 所擁有的 token 數，圖十一的樹狀結構一直延伸至最末狀態 M_8 ，其中 $M_8(0\ 0\ 0\ 0\ 1\ 0\ 2)$ 出現 place o_3 擁有 2 個 token 的情況，此時 Petri Net 模型便不符合 1-bounded 的條件，因此可以驗證此結構為 Non Safe 的流程。



圖十一：以 Coverability Tree 驗證的樹狀結構

2. 可達性：本項驗證以圖十二的 Petri Net 模型為例，透過 Incidence Matrix 演算法，建立輸入矩陣 D 和輸出矩陣 D^+ ，並且透過 $D=D^+ - D$ 運算式，可以得到一個 Petri Net 模型的矩陣 D ，各矩陣如下：



圖十二：流程可達性的測試範例

$$D^- = \begin{matrix} & i & i_1 & o_1 & i_2 & o_2 & i_3 & o_3 & o \\ ti & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ op_1 & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ t_1 & \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ op_2 & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ op_3 & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\ to & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad D^+ = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

第二階段則是狀態驗證，本例的初始狀態為 $M_0=[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$ ，結束狀態為 $M_n=[0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]$ ，根據公式 $M_n=M_0 + x \cdot D$ ，可以得到以下運算式：

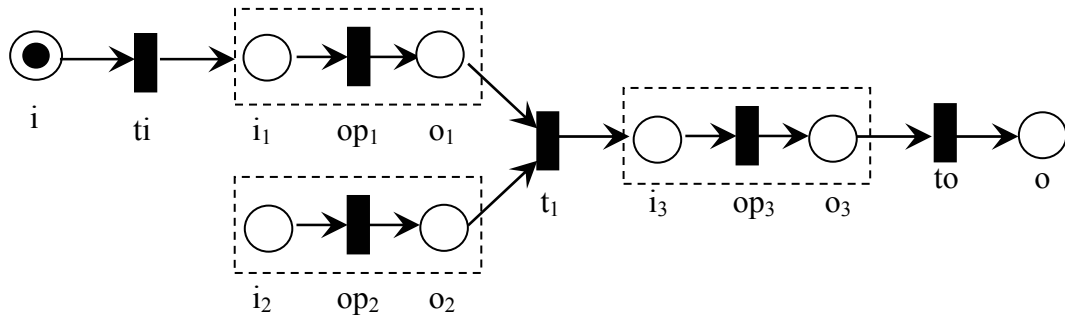
$$[0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]=[1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]+x \cdot \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

我們求解 x 須以 $[x_1\ x_2\ x_3\ x_4\ x_5\ x_6]$ 矩陣替代，可以得到以下的聯立方程式，經由計算可以發現矩陣無解，亦即代表 x 無解。因此可以判斷初始狀態 M_0 無法到達結束狀態 M_n ，其意義為圖十二的 Petri Net 範例並不具有可達性。

$$[-1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1]=[x_1\ x_2\ x_3\ x_4\ x_5\ x_6] \cdot \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} -x_1 = -1 \\ x_1 - x_2 = 0 \\ x_2 - x_3 = 0 \\ x_3 - x_4 = 0 \\ x_4 - x_6 = 0 \\ x_3 - x_5 = 0 \\ x_5 = 0 \\ x_6 = 1 \end{cases}$$

3. 死結偵測：本項驗證以圖十三的 Petri Net 模型為例，透過 Incidence Matrix 演算法，建立輸入矩陣 D 和輸出矩陣 D^+ 如下：



圖十三：流程死結偵測的測試範例

$$D^- = \begin{matrix} & i & i_1 & o_1 & i_2 & o_2 & i_3 & o_3 & o \\ \begin{matrix} ti \\ op_1 \\ op_2 \\ t_1 \\ op_3 \\ to \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} & D^+ = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

根據 $L_{DP} = D \cdot \text{diag}(t_1, t_2, \dots, t_n) \cdot (D^+)^T$ 運算式，將輸入矩陣 D 和輸出矩陣 D^+ 納入運算，可得到遞移矩陣 L_{DP} ，並且透過對 transition 的取代規則，獲得新的遞移矩陣 L_{DP}^* ，各遞移矩陣如下：

$$L_{DP} = \begin{matrix} & i & i_1 & o_1 & i_2 & o_2 & i_3 & o_3 & o \\ \begin{matrix} i \\ i_1 \\ o_1 \\ i_2 \\ o_2 \\ i_3 \\ o_3 \\ o \end{matrix} & \begin{bmatrix} 0 & ti & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & op_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & op_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & op_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & to & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & L_{DP}^* = \begin{bmatrix} 0 & ti & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & op_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & op_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & op_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & to & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

完成以上的作業後，利用 $M_i = M_{i-1} \cdot L_{DP}^*$ 公式，計算狀態 M_i 的演變，它所代表的 place 分別為 $[i \ i_1 \ o_1 \ i_2 \ o_2 \ i_3 \ o_3 \ o]$ 。

- (i). 以 $i=1$ 代入前述之公式，可得到 $M_1 = M_0 \cdot L_{DP}^*$ 。而 $M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ ，因此由 $M_1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \cdot L_{DP}^*$ 可以得到 $M_1 = [0 \ i(t_i) \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ ，由於 $i(t_i)$ 大於等於 1，因此 (t_i) 可以被激發。再將 M_1 正規化(可激發的 transition 之 place 以 1 取代，其他維持 0)，得到 $M_1 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ 。
- (ii). 再以 $i=2$ 代入公式，得到 $M_2 = M_1 \cdot L_{DP}^*$ 。根據公式求解 M_2 ，可寫成下面的聯立式 $M_2 = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0] \cdot L_{DP}^*$ ，計算後的 M_2 為 $M_2 = [0 \ 0 \ i_1(op_1) \ 0 \ 0 \ 0 \ 0 \ 0]$ ，由於 $i_1(op_1)$ 大於等於 1，因此 op_1 可被激發。再將 M_2 正規化為 $M_2 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0]$ 。
- (iii). 以 $i=3$ 代入公式，得到 $M_3 = M_2 \cdot L_{DP}^*$ 。根據公式求解 M_3 ，可寫成下面的聯立式

$M_3 = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0] \cdot L_{DP}^*$ ，計算後的 M_3 為 $M_3 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ ，由於 $o_1(t_1/2)$ 小於 1，因此 t_1 無法被激發，再將 M_3 正規化為 $M_3 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]$ 。因檢視狀態 M_3 中出現所有 place 皆為 0，故代表圖十三的 Petri Net 模型已經發生死結問題。

五、結論

本研究針對網路服務在目前的流程組合中，其可靠性常出現的死結、可達性、及控制權移轉的安全性等問題上，提出了一個以 Petri Net 為基礎的前置驗證程序。由於服務組合所用的流程語言皆為 XML 的格式，因此我們設計了轉換方法協助塑模之間的置換，另外結合過去研究在 Petri Net 上解決可靠性問題的演算法，應用於驗證服務的流程組合中。以下總結本研究的貢獻如下：

- 塑模之間的轉換：服務組合通常以 XML 形式呈現，其目的是供流程組合引擎的解譯，但並不具有被量化分析的條件。由於 Petri Net 模型具有模擬流程的優點，本研究因此選定 Petri Net 為塑模對象，為結合服務流程在實務上的三階段現況：規劃、定義、及執行，因此以塑模間的轉換為主，以利作業連貫。為簡化流程模型轉換間的格式差異，我們以同為 XML 的 PNML 來描述 Petri Net。在研究過程中發現：因為流程語言運用了 workflow 樣板的概念來設計語法，故與專為呈現 Petri Net 的 PNML 無法適度的對映，因此我們進一步的改善轉換程序，將 workflow 樣板予以先分類，透過拆解、轉換、及重組後，獲得轉換後的 PNML 格式。最後，PNML 雖然等同於一個 Petri Net 模型，但它仍是 XML 型式，對於分析工作也無助益，我們因此定義了一項塑模式 $WSCN = (P, T, F, L)$ 並將各 workflow 樣板的 WSCN 塑模式定義於表二，它們是將 PNML 的內容再予以擷取為可量化處理的代數式。本研究最後以服務流程語言 BPEL4WS 為例，應用先前的轉換設計，已成功的將其轉換為 PNML 及 WSCN。雖然實證過程並未測試其他的流程語言，且不同的流程語言在 workflow 樣板的定義及 XSLT 轉換等也應有不同，但本研究設計的程序、實施、及 WSCN 塑模式，則應可提供相關研究擴充。
- 可靠性的驗證：服務組合最終以 WSCN 塑模式呈現後，我們鎖定影響可靠性的控制權移轉安全性、可達性、及死結等為指標，因此一個具可靠性的服務組合必須通過全部的指標。本研究以 Coverability Tree 方法，檢查控制權移轉是否有安全性問題、以 Incidence Matrix and State Equation 方法，確認流程的可達性、最後以 Transitive Matrix 方法，偵測流程是否有死結產生。本研究最後將三項演算法予以程式化，以代理人的形式銜接由 WSCN 塑模式而來的 Petri Net 模型。為驗證確能分析出服務組合在流程設計的問題，我們分別設計三個有瑕疵的服務組合(如圖九, 十二, 十三)，而驗證結果顯示各分析法確能反應出錯誤所在。

參考文獻

1. Aalst, W.M.P. v.d. "Don't go with the flow: Web services composition standards exposed," *IEEE Intelligent Systems* (15:2), 2003, pp.72-85.
2. Aalst, W.M.P. v.d., Hofstede, A.H.M., Kiepuszewski B., and Barros, A.P. "Workflow Patterns," *Distributed and Parallel Databases*(14:1), 2003, pp.5-51.
3. Alameh, N. "Chaining geographic information Web services," *IEEE Internet Computing* (7:5), 2003, pp.22 -29.
4. Benatallah, B., Dumas, M., Fauvet, M.C., and Rabhi, F.A. "Towards Patterns of Web Services Composition," *Patterns and skeletons for parallel and distributed computing*, 2003, pp.265-296.
5. Benatallah, B., Dumas, M., Fauvet, M. C., Rabhi, F. A. and Sheng, Q. Z. "Overview of Some Patterns for Architecting and Managing Composite Web Services," *ACM SIGecom Exchanges* (3:3), 2002, pp.9-18.
6. Benatallah, B., Sheng., Q.Z., and Dumas, M. "The Self-Serv Environment for Web Services Composition," *IEEE Internet Computing* (7:1), 2003, pp.40-48.
7. Billington, J., Christensen, S., Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., and Weber M. "The Petri Net Markup Language:Concepts, Technology, and Tools," *proceedings of the ICATPN 2003*, 2003.
8. Vidal, J.M.; Buhler, P., and Stahl, C., "Multiagent Systems with Workflows," , *IEEE Internet Computing* (8:1), 2004, pp. 76-82.
9. Casati, F. and Shan, M. "Event-based Interaction Management for Composite E-Services in eFlow," *Information Systems Frontiers* (4:1), 2002, pp.19-31.
10. Chandrasekaran, S., Miller, J.A., Silver, G.S., Arpinar, B., and Sheth, A.P. "Performance Analysis and Simulation of Composite Web Services," *The International Journal of Electronic Commerce and Business Media*(13:2), 2003, pp.120-132.
11. Corbett, J.C. "Evaluating Deadlock detection methods for concurrent software," *IEEE Transaction Software Engineering*(22:3), 1996, pp.161-180.
12. Curbera, F., Duftler, M., Khalaf, R., Nagy, W., Mukhi, N., and Weerawarana, S. "Unraveling the Web services Web: An Introduction to SOAP, WSDL, and UDDI," *IEEE Internet Computing* (6:2), 2002, pp.86-93.
13. Hamadi, R. and Benatallah, B. "A Petri Net-based Model Web Service Composition," *in Database Technologies 2003*, Schewe, K. D.and Zhou X.(eds.), 2003, pp.191-200.
14. IBM, Microsoft, and BEA. "Business Process Execution Language for Web Services (BPEL4WS) ver. 1.1," 2003. Available at <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
15. Karp, R.M. and Miller, R. "Properties of a model for parallel computations: Determinacy, termination, queuing," *SIAMJ. Applied Math* (14:6), 1966, pp.1390-1411.
16. Lehmann, M. "Web Services Composition," Available at http://www.isys.uni-klu.ac.at/ISYS/Courses/03SS/S_DKE/lehmann.ppt

17. Marton, A., Piccinelli, G., and Turfin, C. "Service Provision and Composition in Virtual Business Communities," *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*, 1999, pp.336-341.
18. Medjahed, B., Rezgui, A., Bouguettaya, A., and Ouzzani M. "Infrastructure for E-Government Web Services Middleware for Web Services," *IEEE Internet Computing* (7:1), 2003, pp.58 -65.
19. Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A.H.H., and Elmagarmid, A.K. "Business-to-Business Interactions: Issues and Enabling Technologies," *The VLDB Journal*(12:1), 2003, pp.59-85.
20. Murata, T. and Church, R.W. "Analysis of Marked Graphs and Petri Nets by Matrix Equations," *Chicago: Univ. of Illinois, Dept. of Information Eng., Research Report No. M.D.C. 1.1.8*, 1975.
21. Murata, T. "Petri Nets: Properties, Analysis and Applications," *Proceedings of the IEEE* (77:4), 1989, pp.541-580.
22. Rabhi, F. A. and Benatallah, B. "An Integrated Service Architecture for Managing Capital Markets Systems," *IEEE Network Magazine* (16:1), 2002, pp.15-19.
23. Ran, S. "A Model for Web Services Discovery With QoS," *ACM SIGecom Exchanges*(4:1), 2003, pp.1-10.
24. Song, Y. and Lee, J. "Deadlock Analysis of Petri Nets Using the Transitive Matrix," *Proceedings of the 41st SICE Annual Conference*, Vol. 2, 2002, pp.689-694.
25. VanderMeer, D., Datta, A., Dutta, K., Thomas, H., Ramamritham, K., and Navathe, S. B. "FUSION: A System Allowing Dynamic Web Services Composition and Automatic Execution," *Proceedings of IEEE International conference on E-Commerce*, 2003, pp.399-404.
26. Vinoski, S. "Web services interaction models. Current practice," *IEEE Internet Computing* (6:3), 2002, pp.89 -91.
27. Workflow Management Coalition "Workflow Management Coalition Terminology & Glossary," *Workflow Management Coalition Doc. No. WfMC TC-1011*, 1999.
28. Yi, X. and Kochut, K. J. "A CP-nets-based Design and Verification Framework for Web Services Compositions," *Proceedings of 2004 IEEE International Conference on Web Services*, 2004, pp.756-760.
29. Zeng, L., Benatallah, B., Lei, H., Ngu, A.H.H., Flaxer, D., and Chang, H. "Flexible Composition of Enterprise Web Services," *Electronic Markets - The International Journal of Electronic Commerce and Business Media*(13:2), 2003, pp.141-152.
30. Zeng, L., Benatallah, B., Ngu, A.H.H., Dumas, M., Kalagnanam, J., and Chang, H. "QoS-Aware Middleware for Web Services Composition," *IEEE Transactions on Software Engineering*(30:5), 2004, pp.311-327.