

出國報告（出國類別：國科會短期訪問研究）

# 派屈網路應用於動態網路服務架構 之設計與分析

服務機關：國防大學理工學院

姓名職稱：黃義盛 教授

派赴國家：美國

報告日期：98年10月15日

出國時間：98年6月16日

# 摘要

對於網路服務，由於企業範圍、企業應用及其相互整合的發展，網路服務供應商不僅除了試著滿足網路服務用戶(或顧客)原有系統的要求外，更需滿足用戶系統外的功能需求，以利在更具有競爭性的市場下生存。目前最熱門的研究主題為，當用戶的差異性、服務元件性能的區別性及網路資源的限制性等上述情況被考慮下，如何配置網路服務以滿足所有用戶的需求。為了正式展現並塑模網路服務架構，本研究基於派屈網路來建造其網路服務架構。然後，在考量用最高的服務質量選擇最好的構造滿足用戶的非功能性要求下，提出一種最佳演算法則。

# 目 次

摘要	.....	2
目次	.....	3
目的	.....	4
過程	.....	6
心得	.....	8
結論	.....	14
建議事項	.....	15
參考文獻	.....	16

# 目 的

依照美國 W3C (World Wide Web Consortium) 網際網路組織協會定義，一個 Web Services 是爲了能允許不同電腦機台能作有效相互溝通而設計的一個軟體元件，甚至於可適用於一般網路架構。而 Web Services 經常是指網際網路的應用程式，而該應用程式可以透過遠端控制方式或網際網路(World Wide Web)方式去存取其資料進而達成使用者的服務要求。本研究計畫目的是針對現行網路服務架構的設計中，試圖以派屈網(Petri nets) 爲分析工具，進而提供更有效率的存取方式，以利網路使用者更方便的網路服務。

依據以上的定義，我們簡單的說明Web Services工作所需的相關幾個重要標準。因 Web Services是一種軟體元件，它能透過網際網路通訊協定及資料格式的開放式標準(例如 HTTP、XML 及 SOAP等)來爲其他的應用程式提供服務。更具體的說Web Services是一個提供服務的元件，並且是建立在網際網路的開放標準爲基礎。根據以上的認識，因此Web Services可作爲提供服務的元件，且可用來建構分散式架構系統，實現分散式架構動態整合、平衡負擔、單元升級等優點。網路服務[1]-[3]框架逐漸發展成爲一個重要的範例以利分散式的計算。網路服務供應商發佈的網路服務和調用介面，他們打算通過使用網路服務描述語言 (WSDL, Web Description Language)提供，並註冊到一個常見的註冊表，位於通用描述、探索和整合(UDDI, Universal Description, Discovery, and Integration)的網路服務。然後，應用程式發現所需的網路服務，並通過調用介面發送請求。在網路服務供應商反應之後，他們可以通過使用非同步消息傳遞或遠端程序呼叫 (RPC, Remote Procedure Call)模式中，調用簡單物件聯結協定(SOAP, Simple Object Access Protocol)的服務。

一般來說，網路服務供應商必須提供令人滿意的網路服務給所有的用戶，並且與其建立穩定的連接服務。這種機制通常被建立在某種的服務程度同意上[4]。而網路服務性能則可從兩方面進行評估。第一種是功能性的評估[5]。在這方面，所謂的功能性是指一種服務可以被提供；並且在什麼情況下，這功能一旦被提供可以完全符合使用者的需求。另一種則是非功能性的，譬如價格及服務品質(QoS, quality of service) [6]-[7]；而服務品質則是評估、選擇及服務架構所必要的。現今的時代，使用UDDI基礎下的網路服務探索僅只於在功能性上而已。它很難想像可能會有許多網路服務配置，可以滿足功能的要求下執行不同的服務品質屬性。因此，本研究將置重點於如何動態配置網路服務，而使網路服務供應商得以滿足所有用戶非功能性的要求。然而，這研究主要的挑戰

便是來自「當網路供應商提供愈大的彈性及選擇性下，將需要花更多的努力找尋最佳網路服務架構的難題」。舉例來說，根據本研究參考文獻[8]所述的相關統計資料，一群遠端程序呼叫(RPC)服務可能會有超過200多個可候選的架構。因而造成選擇一個具高效能的正確架構需要高水準的專業經驗與足夠的知識。除此之外，服務架構在動態非相同環境下執行，便會產生不同限制情形發生。另當網路服務變得無法使用或不可靠時，其它依賴應它的網路服務便必須被換置以適應新的環境。因此，如要達成運行時的高性能，網路服務架構應在面對變化的各種環境中，立即決定適用的網路服務進行合作。

網路服務構型管理問題也是網路服務管理研究中很熱門的研究議題之一。例如，故障管理需要網路服務構型資訊來追蹤缺點[9]。根據網路服務聯結和綁定管理付費與及費用收費計算[10]。根據不同的變化，所有這些管理活動均需要知道目前的相依性，探索它們的屬性和可能性，來執行服務的重新綁定。因此，開發一種不僅要考慮到基礎的服務元件和資源計畫的網路服務架構，並且要符合使用者不同的網路需求的方法論，應是當務之急。

對於網路服務，由於企業範圍、企業應用及其相互整合的發展，網路服務供應商不僅除了試著滿足網路服務用戶(或顧客)原有系統的要求外，更需滿足用戶系統外的功能需求，以利在更具有競爭性的市場下生存。目前最熱門的研究主題為，當用戶的差異性、服務元件性能的區別性及網路資源的限制性等上述情況被考慮下，如何配置網路服務以滿足所有用戶的需求。為了正式展現並塑模網路服務架構，本訪問研究的目的是基於派屈網路來建造其網路服務架構。然後，在考量用最高的服務質量選擇最好的構造滿足用戶的非功能性要求下，提出一種最佳演算法則。

## 過 程

申請人於 98 年 06 月 16 日搭乘長榮航空公司班機前往香港轉乘美國大陸航空公司班機前往美國，於當地時間下午約二點終抵達紐華克機場，而訪問學校的主持人周教授於機場接機，前往預定的飯店投宿，等安排妥當後周教授即要我一同前往西門子公司參加一個研討會，該會於晚上六點舉行並於九點結束。花了二天的時間找尋長期住宿的地方，終於經過比價與考慮交通便利性後，於是選擇住在紐澤西理工學院附近的學生宿舍 (University Central)，該宿舍每月租金為 1400 美元，離學校的實驗室約 300 公尺。來到該校報到時，系上的秘書幫我製作了學校的進出入管制卡，此卡可自由進出及使用學校的各項設施，使用上非常方便。

訪問期間，大部份時間皆在實驗室內和周教授的學生們及三位國外的訪問學者一同研究討論相關的主題。期間並前往位於波士頓的名校，計有耶魯大學、哈佛大學和麻省理工學院。雖是暑假期間但是仍然有很多教授及學生在校內從事各式各樣的活動。

此次訪問研究期間機會難得，因此行除了在紐澤西理工學院從事研究外，並積極參與附近學校的各項研討會及民間公司的產品發表會。並利用週末假日前往紐約市中心參觀名勝及一些博物館和中國城。

申請人於 9 月 15 日搭乘美國大陸航空公司返回香港並轉搭華航班機返國，完成這次三個月短期訪問研究。

申請人此次訪問研究主要是針對動態網路服務架構之設計與分析，因此和該實驗室的研究學者及學生，主要討論此一議題為主，我們一致認為網路服務框架使得傳統計算產生一種新的革命。當一個網路服務無法滿足使用者多個功能需要時，便必須藉由動態配置在一起以形成一種複合網路服務，以滿足用戶的需求。而服務構型和相關元件組成均是自然相互形成的。服務組成包括三個不同但是重疊的方面；即，行為界面、設計性與和諧性三方面[11]。行為界面係指在組成期間所捕捉給定個別服務行為的從屬性。設計性係指處理與多種服務有關達到一個共同目標的合作過程。而和諧性則是描述在商務環境中網路服務之控制流和數據從屬性，這二者通常藉由網路服務的商務過程執行語言 (BPEL, Business Process Execution Language)完成[12]。由於考慮到網路服務的商務過程執行語言(BPEL)沒有正式的規範，為了現存的網路服務效率和效益，派屈網路 (Petri nets) [13]便被使用來執行網路服務的商務過程執行語言(BPEL)過程的正式驗證。藉由網路服務的商務過程執行語言(BPEL)的句法引導，Stahl[14]將網路服務的商務過程執行語言(BPEL)過程意涵轉換到派屈網路(Petri nets)。Lohmann等人[15]則提

出一種基於派屈網路(Petri nets)的框架，正式分析網路服務的商務過程執行語言(BPEL)過程的相互作用行爲。Zeng等人[16]則是提出一種所謂“服務品質認知中介軟體”(QoS-aware middleware)於網路服務組成，他們模型法據稱可保證服務品質的整體最佳性。前面提到的研究主要是過程導向及提供進階之組成。然而，假使每個網路服務只能從事單項的工作流程[16]，這似乎便反駁了上述所談論的網路服務可以一再重覆支援許多工作流程任務的事實。

## 心得

我們瞭解服務架構可以處理導向函數視圖中的問題。它通常是在服務元件體系架構(SCA, Service Component Architecture) [17]下所完成的。服務元件體系架構(SCA)經由不同的服務組成來建立在服務封裝上。在提供服務並且消耗其它服務的組成部分被實現之後，便透過服務導線建造商業應用。在與導向過程的服務組成相比較，面向功能的服務構造更能產生較多的敏捷性和靈活性。舉例來說，如果兩種服務使網路服務成形，這兩種服務可能於實現的過程中被順序執行或者是平行相依。

一個服務元件體系架構(SCA)模組透過建構和導線連接組件、入口點和外部服務被組合而成。入口點是可提供模組外部組件使用的界面代表。假如在模組裡的組件取決於在模組外部的服務提供，則此服務被描述為外部服務。服務元件體系架構(SCA)組件操作於兩個層次；例如，系統內和模組內的組合元件。動態構造可以被功能性的組件所塑模。然而，服務元件體系架構(SCA)是根據開放的導向服務架構合作所被發展且目前現今的研究中發現大多數服務元件體系架構(SCA)無法與服務品質(QoS)合作[18]。

作為一個動態形成的服務從屬圖(SDG, service dependency graph) [10]已被廣泛用來描述已選擇的網路服務內的關係。然而，服務從屬圖(SDG)是基於網路服務描述語言(WSDL)並且只討論功能性的方面。它很難支援到網路服務的動態構造處理並處理非功能性的需求。例如，Hasselmayr [10]於網路服務下提出一種從屬管理架構，並且用Jini連接技術科技。Liang和Su[19]提出一種基於“和”(AND)、“或”(OR)關係圖來探索網路服務的服務從屬圖(SDG)。前面提到的研究集中於功能的方面，完全沒有考慮到用戶的非功能性的需求。

進一步來說，雖然有很多研究論文[6]-[8]、專案，例如METEOR-S [20]、以及中介軟體，例如SwinDeW [3]以及GlueQoS [21]；然相較於服務品質(QoS)的網路服務選擇，它們並沒考慮到服務架構操作在環境下的時間效能，及接受動態改變的個別服務品質。這些相關的研究促使在本研究提出在此相關的研究工作。

網路服務提供動態構型於服務元件上的不斷改進、資源可用性的變化、或是使用者需求。而銷售管理服務取決於收集訂單的訂單管理服務、收取費用的用戶付費服務、及交付貨物的後勤服務等三大的網路服務。這些三種網路服務有“和”(AND)的相互關係。顧客付費服務更進一步地端賴兩種付款服務，例如，匯款與銀行賬戶等兩種服務方法。在圖一一是以虛線來顯示並說明此二者的虛設網路服務關係是“或”(OR)的相互關係。不過，此兩種服務方法仍有一同驗證顧客信用的功能從屬性。銀行賬戶服務也是採用虛線

來表示其是虛設的服務，因為在後續的構造只能選擇一個，也就是銀行帳戶服務僅能在幾個候選選項中做出選擇。

因此，網路服務就其功能作用選擇不同的網路服務。我們塑模這個問題作為服務功能的從屬性構造(SFDC, Service Functional Dependency Configuration)。單一的網路服務可能提供有多種服務功能的從屬性構造(SFDCs)。不過，同時段僅會有一個構造可以被單一個網路服務所選擇。若我們使用兩個規範架構[22]-[24]，將網路服務從屬性圖轉換成派屈網路；也就是轉換結合“和”(AND)和選擇“或”(OR)功能於派屈網路中。在我們增加幾個有關服務品質(QoS)的參數和測量數據之後，我們便能自動導出最佳化的網路服務架構。我們推測參考到本研究的每一研究人員已對派屈網路有初步的基本認知[13],[25]-[28]。下列定義後續將被使用。

定義 1：派屈網路是一個五個元素的組合，可定義為  $PN = (P, T, I, O, M)$ ，其中  
1)  $P = \{p_1, p_2, \dots, p_m\}$ ， $m > 0$ ，稱之庫所(Place)，是一種以圓圈表示的有限組合。

2)  $T = \{t_1, t_2, \dots, t_n\}$  其中  $n > 0$ ，稱之變遷(Transition)，是一種以條狀表示的有限組合，其與  $P$  的關係為： $P \cup T \neq \emptyset$  and  $P \cap T = \emptyset$ 。

3)  $I : P \times T \rightarrow \{0, 1\}$ ，為輸入函數，其表示為從  $P$  到  $T$  所有的方向弧組合。

4)  $O : P \times T \rightarrow \{0, 1\}$ ，為輸出函數，其表示為從  $T$  到  $P$  所有的方向弧組合。

5)  $M : P \rightarrow \{0, 1\}$ ，是一個  $m \times 1$  列的向量，它第  $i$  個元素代表第  $i$  個  $P$  所內含有Token 的數量，且最初的標記被註記為  $M_0$ ，Token 則是用黑點表示。

所有從變遷(transition)出去到庫所(place)的組合註記為  $t^{\cdot}$ ，而所有從庫所(place)進入到變遷(transition)的組合，則註記為  $^{\cdot}t$ ；另同樣地，所有從庫所(place)出去到變遷(transition)的組合與所有從變遷(transition)進入到庫所(place)的組合，則分別被註記為  $t^{\cdot}$  及  $^{\cdot}t$ 。在圖二中， $t^{\cdot}_1 = \{p_{2-4}\}$ 、 $^{\cdot}t_1 = \{p_1\}$ 、 $p^{\cdot}_5 = \{t_{2-4}\}$ ，且  $(p^{\cdot}_5)^{\cdot} = t^{\cdot}_2 \cup t^{\cdot}_3 \cup t^{\cdot}_4 = \{p_{6-8}\}$ 。

定義2：服務組合  $Se = WS \cup \{s_{dummy}\}$ ，其中  $WS = \{s_{1-z}\}$  為有限的網路服務組合， $s_{dummy}$  表是虛設的服務，其可能是為實際網路服務所需選擇的要項或是組合要項。

定義 3：服務功能從屬構型網路 (S-net, service functional dependency configuration net)，為一非週期循環性的派屈網路。如果：

1) 對所有的  $s$  屬於  $Se$ ，便會存在  $p$  屬於  $P$ ， $s$  將對應到派屈網路服務  $p$ 。同樣地，

如果  $s$  是需設網路服務，其對應到的派屈網路服務  $p$ 。

2) 對所有的  $t$  屬於  $T$ ，假如  $|t'| > 1$  且對所有的  $p_1, p_2$  存在於  $t'$  中，則一定會有“和”(AND)的關係式存在於  $p_1$  與  $p_2$  之間。另對所有的  $t$  屬於  $T$ ，假如  $|t'| = 1, t' = \{p\}$  且  $|p'| > 1$ ，則會存在有“或”(OR)關係式於  $(p')$  中。

3) 每一個變遷(transition)僅有一個輸入的方向弧(input arc)且至少一個輸出的方向弧(output arc)。

4) 註記為  $p'$  的庫所(place) 表示其無任何的方向弧輸入，這輸入的方向弧表示為顧客的需求服務，因此可以得到  $M(p') = 1$ ，且  $M(p) = 0$ ，對所有的  $p \neq p'$ 。

5) 標記(marking)於服務功能從屬構型網路(S-net)的變化規則如下：

i. 假如對所有的  $p$  屬於  $t'$ ， $M(p) \geq I(p, t)$ ，則對於所有  $t$  屬於  $T$  均可以被擊發，反之亦然。並且我們註記  $E(M)$  為一個在標記(marking)之下可以擊發的變遷(transition)組合。

ii. 從標記  $M$  經某一個變遷  $t$  擊發到  $M'$ ，可被註記為  $M \xrightarrow{t} M'$ ，其中如果  $I(p, t) = 1$ ，則  $M'(p) = 0$ ；另一方面，如果  $O(p, t) = 1$  且  $M'(p) = M(p)$  則  $M'(p) = 1$ 。

注意到這邊所提出的服務功能從屬構型網路(S-net)是在下列文獻所提的派屈網路各項範疇內 [13], [29] - [35]。

定義4: 服務功能的從屬性構造(SFDC)如果在時間  $\zeta$  下，可以被註記為  $\alpha(p', \zeta)$ ，且與  $P$  的子網路關係為：

1)  $p'$  屬於  $\alpha(p', \zeta)$  且

2) 所有的  $p$  屬於  $\alpha(p', \zeta)$ ，假如  $p' \neq \emptyset$ ，存在  $t$  屬於  $p'$  且所有的  $p''$  屬於  $t'$ ， $p''$  屬於  $\alpha(p', \zeta)$ 。

舉例來說，網路服務從屬圖(Web service dependency graph)的服務功能從屬構型網路(S-net)。且可列出所有庫所(place)與網路服務(Web service)的關係。最後，我們將舉一個服務功能的從屬性構造(SFDC)， $\alpha(p', \zeta) = \{p', p_{1-8}, p_{12}, p_{14}\}$ 來做說明，為了方便，虛構網路服務(dummy Web service)將不被列入構型中。

一般來說，每一個功能性的網路服務可能會有幾個服務品質(QoS)參數，且可藉由實際量測來評估。服務品質(QoS)參數可以被視為一種運行庫、事務支援、構型管理、成本及相關安全性的性質[7]。每個均由相關度量值及次度量值所組成。例如較大的輸送量、較高的可靠性、較快的機動性和較大的彈性。較快的反應時間，較少的滯留時間

和較低的成本值也是首選因素。我們推測在時間  $\zeta$  下的非虛構網路服務服務品質參數是  $\phi(p, \zeta)$ 。一般情況下，高數值的  $\phi(p, \zeta)$  表示高的品質。例如  $\phi(p, \zeta)$  可以是生產力，其表示一段時間內已完成的服務請求數量。為了反映成本， $\phi(p, \zeta)$  被定義為零值減去所涉及的請求服務。如果從  $p$  的得不到任何的服務，則  $\phi(p, \zeta) = -\infty$ 。考慮服務的全功能取決於它自己的功能；它依賴的次服務，我們推測一個  $C(p', \zeta)$  的服務品質(QoS)參數是所有非虛設相關服務的服務品質(QoS)參數。例如： $QoS(C(p', \zeta)) = f(\phi(p, \zeta) \mid p \in C(p', \zeta) \setminus \{p_{dummy}\}) \sum_{p \in C(p', \zeta) \setminus \{p_{dummy}\}} \phi(p, \zeta)$ 。我們的演算法則為，給定  $p'$  及  $\zeta$ ，尋找派屈網路服務功能從屬構型網路(S-net)中的最佳  $C(p', \zeta)$ ，正如有著最高  $QoS(C(p', \zeta))$  的服務功能的從屬性構造(SFDC)。

一般來說，當更多相依的網路服務被加進到構型時，構型服務品質(QoS)參數便會變差。例如，當與單一個別的網路服務相比較，整體的網路服務似乎在可靠性、反應性及成本上均來得比較差。基於此種特性，本研究所提的演算法則，在發現目前構型的服務品質參數比最佳化的構行糟時，便會停止繼續延此網路服務構型搜尋下去，並立即轉向搜尋另一種構型。因此， $C(p', \zeta)$  的服務品質(QoS)參數便會依此演算法則不斷地上升。依此，我們可以得到一個結論，那就是當我們所提的演算法完成運算時，此時便會得到擁有最佳服務品質參數( $QoS(C(p', \zeta))$ )的構型( $C(p', \zeta)$ )。另一提的是，此演算法則所遇到最差的情形就是搜尋所有的可達樹狀圖來得到最佳化的最佳服務品質參數( $QoS(C(p', \zeta))$ )的構型( $C(p', \zeta)$ )。

定理1: 假如  $|P \setminus \{p', p_{dummy}\}| = \eta$ ，前面所提及最差的演算法則情況為  $O(e^\eta)$ ，其中  $e \approx 2.71828$  為此演算法則的基礎。

證明: 我們註記構型編號為  $\Omega$ 。另在最糟的情況下，我們有  $k$  個虛設網路服務，註記為  $k$  個群，其中每一群與其它群之間有一個“和”(AND)的關係式。在第  $i$  個群中，我們會有  $x_i$  ( $i = 1, 2, \dots, k$ ) 的網路服務，其中每一個網路服務與其它的網路服務存在有“或”(OR)的關係式。然後，我們便會得到  $\Omega \leq \prod_{i=1}^k x_i$  及  $\sum_{i=1}^k x_i \leq \alpha \eta$ ，其中  $\alpha = \text{Max}_{p \in P} (| \cdot p |)$ 。

當  $x_1 = x_2 = \dots = x_k = \alpha \eta / k$ ，便可證明  $\prod_{i=1}^k x_i$  便可得到最大值  $(\alpha \eta / k)^k$ 。在最糟的形狀中，我們便必須比較  $(\alpha \eta / k)^k$  的次數來得到最佳構型。我們依我們的演算法則將  $(\alpha \eta / k)^k$  轉換成  $y(k) = k \ln(\alpha \eta / k)$ 。然後，可以得出  $y'(k) = \ln(\alpha \eta) - \ln(k) - 1$  及  $y''(k) = -1/k$ ，其中  $y'(k)$  與  $y''(k)$  分別為  $y(k)$  的一階微分導數及二階微分導數。讓  $y'(k) = 0$ ，我

們可得出 $k=\alpha\eta/e$ 。另考慮 $y''(k)<0$  ( $k$ 值 $>0$ )， $y(k)$ 便會在 $k=\alpha\eta/e$ 時，達到它的最大值。結果，我們可以得到 $\Omega \leq \prod_{i=1}^k x_i \leq e^{\alpha\eta/e}$ 及最差的演算法則情況為 $O(e^\eta)$ 。

因此，“構型最佳化(*Optimal Configuration*)”演算法則的複雜性於最糟情況下約等於 $O(e^\eta)$  (考慮相對小的 $\alpha$ 、 $e$ 及在大型服務功能從屬構型網路(S-net)的 $\eta$ )。雖然在最糟的情況下，本研究的演算法則已被證明約是呈現指數形態的時間常數，但是其時間變化的幅度僅依從屬構型網路(S-net)中的“或”(OR)關係情況來做改變(非依據所有庫所(place)來變化)，且亦端賴實際的環境情況。舉例來說，雖然在從屬構型網路(S-net)， $\eta = 15$ 、 $\alpha = 2$ ，但是僅有16個可以選擇的構型。另一方面，在我們的演算法下，其構型產生可以是同步進行運算。這意謂，每一個有關聯的變遷(transition)  $t$ ，可以被同時擊發。一旦此多重運算機制成立，便可更加速我們所提的演算法則。進一步地說，我們在獲得最佳構型的情況下，亦可獲得比使用者(或顧客)預期更好的效能。因此，一旦他們找到他們想要的網路服務構型，本演算法亦會立即停止運算，來符合到使用者(或顧客)的實際期盼及需求。這也說明了本研究所提的演算法則合乎應用性。

爲了評估 “構型最佳化(*Optimal Configuration*)”演算法則，本研究報告將模擬構型在顧客滿意率中的表現做爲評估依據。以給定的時間( $\Upsilon$ )爲執行週期下，我們以 *Invoking*( $\Upsilon$ )符號代表需求值、以 *Satisfied*( $\Upsilon$ )符號代表滿意度值。因此，使用者滿意率便爲  $C\gamma(\Upsilon) = \text{Invoking}(\Upsilon)/\text{Satisfied}(\Upsilon)$ 。除此之外，我們將與目前已被提出的兩種典型演算法則來做比較。第一種方法稱之固定式(*Fixed*)的演算法，其選擇第一種可行的架構且永遠不會改變。另一個則是稱之隨機式(*Random*)，其採隨機選擇構型而不考慮性能的方式。

本研究報告爲了方便模擬，特將網路服務成本(cost of the Web service)作爲服務品質(QoS)的參數。在正常的情况下，吾人可以推測，當處理變化性很高的環境中，其網路服務成本便會隨之變化。

然而，隨機式(*Random*)演算法並非在所有情況都比固定式(*Fixed*)的演算法好。在模擬環境過程中，在考慮到正常分佈變化總數與其正常分佈相符情況下，我們註記在正常分佈下，固定式(*Fixed*)與隨機式(*Random*)演算法的獲得成本分別爲 $N(\mu_F, \sigma_F^2)$ 及 $N(\mu_R, \sigma_R^2)$ ，其中 $N(\mu, \sigma^2)$ 表示平均爲 $\mu$ 的正常分佈，而 $\sigma$ 爲其標準差。從成本分佈參數，可以很容易初步得到  $\mu_F > \mu_R$  及  $\sigma_F \approx \sigma_R$ 。另推測可承擔的價格爲 $N(\mu_U, \sigma_U^2)$ ，我們可得到固定

式 (*Fixed*) 與隨機式 (*Random*) 演算法的顧客滿意率分別  $\Phi((\mu_U - \mu_F) / \sqrt{\sigma_F^2 + \sigma_U^2})$  及  $\Phi((\mu_U - \mu_R) / \sqrt{\sigma_R^2 + \sigma_U^2})$ 。其中  $\Phi(x)$  為拉氏函數 (*Laplace function*)。在可承受的價格固定在 390 時，我們可以計算以固定式 (*Fixed*) 演算法下的顧客滿意率為  $\Phi(x) = \Phi(0.2941) = 0.6141$ ，這數字與圖四的模擬結果相近。因此當  $\mu_F > \mu_R$  及  $\sigma_F = \sigma_R$ ，隨機式 (*Random*) 演算法將有較高的顧客滿意率。然而，如果固定式 (*Fixed*) 演算法第一次所選擇的服務構型為  $\mu_F < \mu_R$  (例如， $\{P', P_{1-6}, P_8, P_{10}, P_{12}, P_{15}\}$ )，它便會得到比隨機式 (*Random*) 演算法高的顧客滿意率。

總結來說，前面提及的模擬結果證明我們的演算法最適用於在傳送高服務品質 (QoS) 的網站服務 (Web Service) 下請求變化度非常高的還境中。

## 結 論

爲因應網路服務和服務元件與資源的限制情況，及面對多元化和急變的使用者要求下，如何管理存在的網路服務來形成一種高性能架構以迎合非功能性的需求，便變得越來越重要。本研究爲解決此問題，美國理工學院周教授研究團們曾特提出一種系統化的方法(註：即以上所提的研究方法)。

首先，爲了展現網路服務及自動的構型，其研究基於派屈網路下，提出一種服務功能性的從屬構型。第二，基於這個構型網路，再提出一種最佳化的演算法，使其能夠傳回有最佳服務品質(QoS)參數的服務構型。經與其它強制窮舉的搜索方法相比，此演算法通常無需要搜索的所有可能的構型，來獲得最佳的一個構型。請注意到Zeng等人於服務品質網路服務組成認知(QoS-aware Web services composition)一文[16]中所提到演算法則[16]。因此我們可以在使用者不同需求及改變下能夠建立最高效能構型的演算法則。經與他們的演算法則做比較，在處理大規模的問題上，其網路服務的選擇將造成電腦運算成本的負擔；然而，就他們所提的演算法則，其不僅將網路服務可再使用性納入考率，且可以同時處理多種程序而顯得更有效能。最後，無論是理論的分析及實際的模擬結果，均可證實他們所提的演算法則，確實可以達到最高的顧客滿意率。

除此之外，他們所提的演算法則還可以輕易地整合已存在網路服務環境，說明如下。首先，依根據客製化或是特殊規範需求，探索所有的網路服務。第二，藉由在服務資訊中分析所有功能組成及選擇，來建立完整的服務功能從屬構型網路。第三，在所有的構型中選擇服務品質屬性(QoS attribute)。第四，對所有非虛設的網路服務，收集其服務品質屬性值。

延伸本研究所提出的演算法來處理顧客多重需求的網路服務(例如，最低成本構型可能無法最佳的顧客滿意率)，將是未來我們未來的工作目標。未來的探索將朝向在多重需求情況下，尋找較適宜對等機制來建立合適的構型，而這也將是重要且有趣的議題。目前於派屈網路理論中被提出的基本虹吸(elementary siphon)及彩色派屈網路(colored resource-oriented Petri nets)等觀念，應可以做爲未來複雜的網路服務之塑模、分析、規劃及驗證。

## 建議事項

- 一、國科會補助金額似乎不足以應付紐澤西的生活物價指數，建議應提高補助金額。
- 二、本次短期訪問研究發現大陸地區的教授出訪人數眾多而且訪問期間長達二年，建議國科會應提高教授訪問研究的人數及延長訪問期間，以期增進研究能量的開發。
- 三、本次短期訪問研究發現大陸地區及印度出國的留學生數目非常的多，反而國內的留學生卻很少見到，建議政府相關單位應鼓勵國內優秀學子到國外進修研究，以培養國際觀。
- 四、感謝國科會及國防大學理工學院，給予申請人此次訪問研究機會，雖然時間不長，但個人收穫甚多。

## 參考文獻

- [1]“Web service activity,” *World Wide Web Consortium*. [Online]. Available: [www.w3.org/2002/ws/](http://www.w3.org/2002/ws/)
- [2]F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi and S. Weerawarana, “Unraveling the web services web: An introduction to SOAP, WSDL, and UDDI,” *IEEE Internet Comput.*, vol. 6, no. 2, pp. 86–93, Mar./Apr. 2002.
- [3]J. Yan, Y. Yang, and G. K. Raikundalia, “SwinDeW—A p2p-based decentralized workflow management system,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 5, pp. 922–935, Sep. 2006.
- [4]D. A. Menasce, “Mapping service-level agreements in distributed applications,” *IEEE Internet Comput.*, vol. 8, no. 5, pp. 100–102, Sep./Oct. 2004.
- [5]J. Cardoso and A. Sheth, “Semantic e-workflow composition,” *J. Intell. Inf. Syst.*, vol. 21, no. 3, pp. 191–225, Nov. 2003.
- [6]D. A. Menasce, “QoS issues in web services,” *IEEE Internet Comput.*, vol. 6, no. 6, pp. 72–75, Nov./Dec. 2006.
- [7]S. Ran, “A model for web services discovery with QoS,” *SIGecom Exchanges*, vol. 4, no. 1, pp. 1–10, Mar. 2003.
- [8]C. Yi and K. Nahrstedt, “QoS-aware dependency management for component-based systems,” in *Proc. 10th IEEE Int. Symp. High Performance Distrib. Comput.*, San Francisco, CA, Aug. 2001, pp. 127–138.
- [9]A. Keller and G. Kar, “Dynamic dependencies in application service management,” in *Proc. Int. Conf. Parallel Distrib. Process. Techn. Appl.*, Las Vegas, NV, 2002.
- [10]P. Hasselmayer, “Managing dynamic service dependencies,” in *Proc. 12<sup>th</sup> Int. Workshop Distrib. Syst.: Oper. Manage.*, Nancy, France, Oct. 2001, pp. 141–150.
- [11]A. Barros, M. Dumas, and P. Oaks, “Standards for web service choreography and orchestration: Status and perspectives,” in *Business Process Management Workshops*, vol. 3812. Berlin, Germany: Springer-Verlag, 2006, pp. 61–74.
- [12]T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana, *Business process execution language for web services*, May 2003. Version 1.1 [Online]. Available: [http://www-106.ibm.com/developerworks/web\\_services/library/ws-bpel](http://www-106.ibm.com/developerworks/web_services/library/ws-bpel)
- [13]M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.
- [14]C. Stahl, “A Petri net semantics for BPEL,” Humboldt-Universität zu Berlin, Berlin, Germany, Informatik-Berichte 188, Jul. 2005.
- [15]N. Lohmann, P. Massuthe, C. Stahl, and D. Weinberg, “Analyzing interacting BPEL processes,” in *Proc. 4th Int. Conf. Bus. Process Manage.*, Vienna, Austria, Sep. 2006, pp. 17–32.

- [16]L. Z. Zeng, B. Benatallah, A. H. H. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 311–327, May 2004.
- [17]J. L. Fiadeiro, A. Lopes, and L. Bocchi, "A formal approach to service component architecture," in *Proc. 3rd Int. Workshop web Serv. Formal Methods*, Vienna, Austria, Sep. 2006, pp. 193–213.
- [18]Z. L. Zou and Z. H. Duan, "Building business processes or assembling service components: Reuse services with BPEL4WS and SCA," in *Proc. 4th IEEE Eur. Conf. web Serv.*, Zurich, Switzerland, Dec. 2006, pp. 138–147.
- [19]Q. A. Liang and S. Y. W. Su, "AND/OR graph and search algorithm for discovering composite web services," *Int. J. web Serv. Res.*, vol. 2, no. 4, pp. 48–67, Oct.–Dec. 2005.
- [20]A. Sheth, J. Cardoso, J. Miller, and K. Kochut, "QoS for service-oriented middleware," in *Proc. Conf. Systemics, Cybern. Informatics*, Orlando, FL, Jul. 2002, pp. 528–534.
- [21]E. Wohlstadter, S. Tai, T. Mikalsen, I. Rouvellou, and P. Devanbu, "GlueQoS: Middleware to sweeten quality-of-service policy interactions," in *Proc. 26th Int. Conf. Softw. Eng.*, Edinburgh, U.K., May 2004, pp. 189–199.
- [22]J. Q. Li, Y. S. Fan, and M. C. Zhou, "Performance modeling and analysis of workflow," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 2, pp. 229–242, Mar. 2004.
- [23]R. Bouyekhf and A. E. Moudni, "On the analysis of some structural properties of Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 35, no. 6, pp. 784–794, Nov. 2005.
- [24]J. H. Park, "A deadlock and livelock free protocol for decentralized Internet resource coallocation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 123–131, Jan. 2004.
- [25]B. Hruz and M. C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems (Advanced Textbooks in Control and Signal Processing)*. London, UK: Springer, 2007, p. 341.
- [26]M. C. Zhou and M. P. Fanti, Eds., *Deadlock Resolution in Computer- Integrated Systems*. New York: Marcel Dekker, 2005.
- [27]M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.
- [28]M. C. Zhou and M. D. Jeng, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 333–357, Aug. 1998.
- [29]E. Zussman and M. C. Zhou, "A methodology for modeling and adaptive planning of disassembly processes," *IEEE Trans. Robot. Autom.*, vol. 15, no. 1, pp. 190–194, Feb. 1999.
- [30]E. Zussman and M. C. Zhou, "Design and implementation of an adaptive planner for disassembly processes," *IEEE Trans. Robot. Autom.*, vol. 16, no. 2, pp. 171–179, Apr. 2000.
- [31]Y. Tang, M. C. Zhou, E. Zussman, and R. J. Caudill, "Disassembly modeling, planning and applications," *J. Manuf. Syst.*, vol. 21, no. 2, pp. 200–217, 2002.
- [32]M. Gao, M. C. Zhou, and R. J. Caudill, "Integration of disassembly leveling and bin

- assignment for demanufacturing automation,” *IEEE Trans. Robot. Autom.*, vol. 18, no. 6, pp. 867–874, Dec. 2002.
- [33]M. Gao, M. C. Zhou, and Y. Tang, “Intelligent decision making in disassembly process based on fuzzy reasoning Petri nets,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2029–2034, Oct. 2004.
- [34]Y. Tang, M. C. Zhou, and M. Gao, “Fuzzy-Petri-net based disassembly planning considering human factors,” *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 4, pp. 718–726, Jul. 2006.
- [35]Y. Tang and M. C. Zhou, “A systematic approach to design and operation of disassembly lines,” *IEEE Trans. Autom. Sci. Eng.*, vol. 3, no. 3, pp. 324–329, Jul. 2006.
- [36]Z. Li and M. C. Zhou, “Elementary siphons of Petri nets and their applications to deadlock prevention in flexible manufacturing systems,” *IEEE Trans. Syst., Man, Cybern.*, vol. 34, no. 1, pp. 38–51, Jan. 2004.
- [37]N. Wu and M. C. Zhou, “Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.
- [38]N. Wu and M. C. Zhou, “Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems,” *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 658–669, Oct. 2001.
- [39]Z. Li and M. C. Zhou, “Two-stage method to design liveness-enforcing Petri net supervisor for FMS,” *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 313–325, Nov. 2006.
- [40]Z. Li, H. Hu, and A. Wang, “Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 517–526, Jul. 2007.
- [41]J. S. Lee, M. C. Zhou, and P. L. Hsu, “A Petri-net approach to modular supervision with conflict resolution for semiconductor manufacturing systems,” *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 4, pp. 584–588, Oct. 2007.
- [42]Z. Li and M. C. Zhou, “Control of elementary and dependent siphons in Petri nets and their application,” *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 133–148, Jan. 2008.
- [43]Z. Li, M. C. Zhou, and M. D. Jeng, “A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions,” *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Jan. 2008.