

以過濾及評選模型(FLQR)提升網路服務之品質評估效能

Using the Filter Large Qualify Rest (FLQR) Model to Improve Quality Evaluation Performance of Web Services

戚玉樑 黃崇德

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
andy_huang@kpec.com.tw

Yu-Liang Chi Chung-De Huang

Dept. of Management Information System, Chung Yuan Christian University
200, Chung Pei Rd, Chung Li, Taiwan, R.O.C., 32023.
maxchi@mis.cycu.edu.tw
andy_huang@kpec.com.tw

摘要

網路服務技術(Web Services)是以文件為中心的應用系統整合方法，它利用訊息的封裝及傳遞，完成異質環境所需克服的互通性問題，因此已逐漸應用在電子商務中；另一方面，多元的服務提供者突顯了服務品質不確定的問題，因此如何濾選合適的服務，則是未來服務仲介者的重要課題。本研究著眼於後網路服務時代的「仲介」問題，提出兩階段的「先過濾再評選」(FLQR, Filter Large Qualify Rest)之改善模型：過濾機制是以減少候選服務之數量為標的，因此考量之因素如服務狀態、穩定性、價格、安全等非功能性因素；其次，評選機制是以服務本身的功能性因素之偏好來決定，並以加權法計算出推薦的序列。由實際的研究結果顯示，使用者透過服務品質的評選機制，將依照品質模型推薦服務的排序，因此可有效解決品質的不確定性。由於一般組織機構皆可建置其區域的服務仲介者，因此本研究的FLQR模型將特別適用於此類服務仲介者的改良。

關鍵詞：服務發現模式、服務品質評估、服務篩選。

Abstract

Web Services architecture (WSA) is an emerging technology for integrating Web-based applications. Due to services are all outsourcings, requesters may have a variety of choices to satisfy their criteria. That is, amount of candidate services advantage the flexibility in implementation. However, requesters encounter the decision issue of selecting a proper service. Motivated by the brokering problems, this study proposes a Filter Large Qualify Rest (FLQR) model to improve a discovery process within a service broker. The model is consists of three components: i). Filter Large is related to remove improper services such as unavailable, expensive, and inefficiency. ii). Qualify Rest is giving a quality evaluation model to compute the performance of each service. iii). Finally, a feedback mechanism offers learning technique

to complement FLQR model. In the end, this study utilizes a private UDDI registry center as a services broker. Empirical result suggests that the brokering process in WSA can be improved by using the FLQR model.

Keywords: Service Discovery Model, Service Quality Evaluation, Service Filter

壹、緒論

網際網路標準組織(W3C, World Wide Web Consortium)所發展的XML改變了電子商務的運作模式，它不但可做為共通性資料交換格式，也提供分散式環境整合的基礎。為使異質環境能藉訊息來溝通，SOAP的雛型在1999年即已制定，隨後納入W3C繼續修訂規格，並搭配描述服務用途的WSDL及以律定服務仲介機制的UDDI等規範，共同建立網路服務架構的規範。由於網路服務是以現有的技術為基礎，故近年來已引進至電子商務環境，而各種支援的工具，例如程式語言(如Java及.NET等)、應用程式伺服器軟體(如Apache Tomcat及BEA Web Logic等)及公用的服務仲介者(如Microsoft及IBM等)，均已普遍可見；簡言之，如同Curbera et al. (2002)、Dalal et al. (2003)等的評論，網路服務技術已成為分散式應用系統的重要架構及整合方法之一。

網路服務與傳統的應用系統在架構上的最大差異，是在Client與Server之間加入了仲介者(Broker)的角色，這項變革使得需求者與提供者都不須是特定的對象。因此，只要是內容符合需求者的要求，服務可能是根本事前不知道的來源，就類似查詢網頁一樣，只不過仲介者所媒介的是應用程式或系統。網路服務架構的確改善了使用跨平台應用程式的習性，但也因為服務提供者並不受仲介者的掌控，因此仲介問題充斥著許多不確定性。Ran (2003)的研究即曾指出，服務仲介機制並不負責個別網路服務的管理，因此目前在服務註冊中心(仲介者)內有高達48%的鏈結都已經不能使用。本研究為檢驗Ran的說法，我們以名稱stock為標的，並在技術模型(t-Model)中查詢，經偵測其服務端點之狀態如表1所示，其中之「不可用」的情況包含服務描述文件位置錯誤、服務不存在或主機未在連線，我們發現高達75%以上的stock服務並不能使用，由此可見使用網路服務在仲介問題上的嚴重性。

表1：各UDDI之服務狀態

服務註冊中心(仲介者)	服務數目	不可用數目	不可用比例
IBM UDDI Registry Center	28	24	85 %
Microsoft UDDI Registry Center	28	21	75 %

服務仲介雖然突破傳統整合須事前確定提供者的限制，但也相對造成難以掌握提供者狀態的困境，仲介機制至少面臨下列兩項問題：(1)對網路服務的現況缺乏進一步的服務控管與輔助機制，例如服務是否堪用、不穩定、或是負載過高導致效率低落等問題皆無法掌握。(2)缺乏有效的退場機制，使得實際不堪用的服務仍充斥於註冊中心。

目前許多關於網路服務仲介的研究均已關心此議題，且大多以如何掌握服務品質(QoS, Quality of Services)的角度切入，但由於品質評估的資源耗損將隨著網路服務數量的成長而加大，在實務應用上較不經濟，因此我們認為在實行服務品質評估前，應先行縮小評估的對象範圍。本研究對此議題的作法是由架構面著手，提出兩階段的先過濾再評選(FLQR)，並設計各階段相互的運作機制，以期服務仲介者可以應用於服務的篩選，另外也加入動態更新的回饋機制，以保持服務品質的及時性。本研究的架構如下：第二節簡要說明應用系統整合之演進、網路服務在仲介機制的問題，及目前關於服務品質評估的相關研究；第三節說明FLQR模型的設計理念及設計流程及以FLQR模型建構一個改良式的服務仲介者；第四節討論FLQR所能改善的服務仲介問題及實證過程中的經驗；第五節我們對本研究總結並提出應用的建議。

貳、文獻探討

一、系統整合與網路服務技術

異質環境的整合是為達成資源共用及資料共享之目的，Vinoski(2002)曾指出異質平台之間的溝通，必須是透過雙方都認可的通訊協定、作業方式、甚至於開發語言等，而異質平台的相互操控，則是需要透過雙方都同意的中介軟體(middleware)才可以進行。中介軟體近年來在應用系統整合上扮演很重要的角色，兩種主要的中介軟體規範(DCOM及CORBA)也普遍可見，例如Sullivan et al.(1999)指出DCOM及後繼者COM+是微軟所提出的分散式運算架構，但兩項架構僅支援與微軟平台相容的軟體；另外，Siegel(1998)及Vinoski(1997)曾指出CORBA為常用的系統整合規範，它的優點是不受作業平台或開發方式侷限，也是功能較COM+更強大的跨平台中介軟體規範。

近年來，由於XML的發展，使得傳統以中介軟體為主的系統整合，逐步以更具彈性的網路服務技術取代，Kou et al.(2000)即指出網路服務架構可溯源至XML規範，由於其Text-based的特性，目前已廣泛的應用在各領域中做為基礎的資料格式。Stal(2002)則指出近年來XML的衍生規範已朝向更低階的基礎架構，其中以XML Messaging為溝通基礎的網路服務架構，已成為異質環境整合新興方案之一。根據W3C(2004)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(requester, provider, and broker)。另外，SOA也常以三項規範來詮釋各角色間的互動或作業依據，此架構所涉及的技術規範包括：

- 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。
- 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。
- 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，亦即提供網路服務環境的仲介機制。

目前網路服務在跨平台系統上的應用很廣泛，例如Alameh(2003)透過服務的組合於地理資訊系統(GIS)上，建立開放性的聯合服務介面，提供完整的地理圖像描繪功能給使

用者。Medjahed et al.(2003)曾以網路服務組合的方式，整合分散各地的社會福利系統，達成單一服務窗口而減少行政手續的效益，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 et al.(2002)則將網路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(CMS)。另外，還有更多在不同應用領域的研究結論也對網路服務能創造更大效益的持正面態度[2, 4, 6, 7, 8, 17, 20, 26]。

二、服務仲介與品質問題

根據W3C的定義，網路服務實踐了程式對程式(A2A, AP to AP)及群組對群組(P2P, Peer to Peer)的運作架構，特別是P2P創造了服務來源的多元化，但也因為多元化，網路服務在品質上並不易掌控。Ran (2003)及Zeng et al. (2004)即指出，在網路服務環境下，提供功能性一致的服務將會有許多，但如何在他們之中選擇一個適合的服務，則是取決於非功能性的因素，例如效能或可靠性等，因此負責儲存服務資訊的註冊中心(仲介者)，應提供更進一步的管控機制。由於服務導向架構促使需求與供給之間更加的彈性，在Kou et al.(2000)、Vinoski (2002)、Stal (2002)、Ran (2003)及Menasce(2004)的研究中均分析與傳統系統的差異，我們整理歸納如表2：

表2：傳統資訊系統與服務導向架構的比較表

	傳統資訊系統	服務導向架構
鬆散耦合	-	●
平台中立	-	●
系統互控性	-	●
服務重覆使用	-	●
資訊正確性	●	-
服務可靠度	●	-
服務穩定性	●	-

● 符合需求

服務導向架構雖然具有鬆散耦合、平台中立、系統互控性及服務重覆使用的特性，但在服務的可靠度及穩定性上明顯需要改善，許多研究將這些非功能性的因素定位為服務的品質問題，例如Menasce(2002)將品質進一步區分為可用性、服務處理能力、執行時間、回應時間、安全性(如授權、不可否認性等)，Zeng et al.(2003)則認為服務品質的因素可考慮五項指標：價格、執行時間、可靠度、可用度、及信用等級，而服務品質的總合指標則為最後選擇服務的依據。

三、服務品質的評估

服務品質的內容包羅萬象,也常因為評估的角度不同而有不同的面向,Bevan (1999)的研究把品質分成內部、外部及使用等三部份。內部品質可以由靜態屬性來審視,如文件及程式。外部品質可在系統執行時由動態屬性來測量,如回應時間。使用品質的測量,主要視符合使用者需求的程度來定義,如使用者滿意度。Moorsel(2001)提出服務的定量評量指標,包括系統考量(QoS:System)、工作執行時期(QoS:Task)、使用者經驗(QoE)及企業經驗(QoBiz)等品質影響構面,而各構面分別有其組成的細項,例如QoS:Task包含有服務請求成功機率、否順利完成、回應時間及處理的能力等;Naumann et al.(1999)針對網路上異質系統整合的資訊品質也提出品質評量因素,所提到的資訊品質指標包括資訊評價、資訊價格、資訊可用度、資訊正確度、資訊穩定度等。

目前網路服務的品質評估大致是承襲傳統軟體品質評估的方法,並予以應用在服務仲介的機制上。Conti(2002)曾提出改良的搜尋模式,把擴展性、處理能力、可靠度、可用度等服務品質因子加到服務仲介者。Sheth(2002)的研究亦對可靠度、可用度、及系統的穩定性等,提出計算出系統的反應時間並即時偵測異常狀況等方法。綜合各學者對影響網路服務之品質因素可分成四大構面(整理如表3),包括服務描述、服務可信度、服務效能及服務安全。

表3：網路服務品質影響因素

影響構面	影響因素	說明
服務描述	價格 (Price)	使用服務之價格
	精確度 (Accuracy)	服務請求正確之程度
	信用評比 (Reputation)	信用評價排名
服務可信度	狀態 (Status)	服務之狀態
	可用度 (Availability)	服務可被使用的機率
	可靠度 (Reliability)	服務在預期內完成的機率
服務效能	處理能力 (Throughput)	服務可處理請求的數目
	執行時間 (Duration)	從發出請求到得知結果的時間
服務安全	安全性 (Security)	授權,不可否認

參、兩階段濾選模型(FLQR)之設計

為解決現階段服務仲介者在評選機制上的不足,我們提出兩階段的濾選模型(Filter Large Qualify Rest),以下簡稱為FLQR模型。由於隨著網路服務在應用上日益普及,在可見的未來,提供相同功能的服務將愈來愈多,Filter Large即為先行剔除不適用的服務,而過濾的條件以非功能性因素為主。另一方面,過濾是消極性的排除工作,但Qualify Rest則是積極性的評選機制,我們可藉品質評選來區隔網路服務的優劣。FLQR架構的概念如圖1:

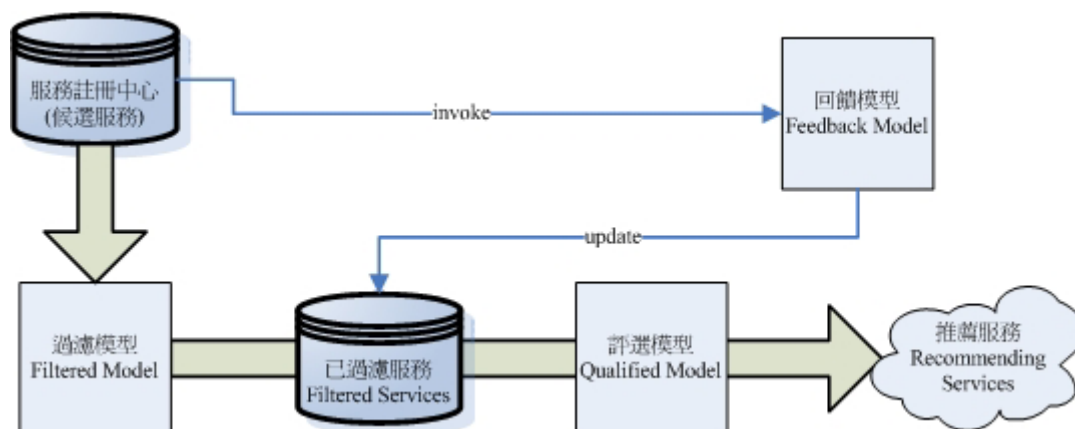


圖1：兩階段濾選模型示意圖

圖1的FLQR模型分為三個子模型，分別是過濾模型、評選模型及回饋模型，而作用的資料分別為：(1).候選服務：是指現存於服務註冊中心的已登錄的服務，可視為本研究的起始過濾來源；(2).已過濾服務：是指經由過濾機制篩選的堪用服務；(3).推薦服務：是指利用評選機制所做的排序的結果，可以作為服務選擇的依據。各模型說明於以下各節。

一、過濾模型(Filtered Model)

為剔除不堪使用的服務，過濾模型的設計分成兩階段實施如圖2：第一階段主要是決定以何種服務過濾策略來取得需求之服務，例如精確率策略可讓候選服務數量縮小，回收率則可讓候選服務數量擴大，而對應到實行階段，即是透過服務仲介者取得服務資訊，方法包含Keyword-Based、Frame-Based、或Deductive Retrieval等的方法，它們都能協助找出適合的網路服務，本研究主要是透過服務仲介者所提供之關鍵字為基礎來進行過濾策略第一階段之實現，以求得符合使用者需求之服務。第二階段篩選的目的在於取得可用的服務，我們以網路服務可測量之項目當作此階段篩選條件，對應到實行階段，服務狀態即為其中過濾條件，以期縮減網路服務之數量。由於第一階段的過濾策略涉及其他數理模型的訂定，因此本研究保留此設計並待後續擴充，目前僅針對第二階段實作。

過濾模型之結果應利用儲存模式交付後續之評選模型使用，儲存可利用類似註冊中心之資料庫方式，以加速處理之效率。本研究對服務狀態的資訊，是以開發偵測程式並以alive狀態為過濾條件，其偵測結果儲存於資料庫的欄位如下所示：

- 服務識別代號(S_{id})：服務在註冊中心的唯一識別序號(service key)。
- 找尋次數(N_{total})：找尋服務所發生的總次數。
- 找尋成功次數($N_{success}$)：找尋服務的成功次數。
- 執行完成次數($N_{complete}$)：服務可順利執行的次數
- 平均執行時間(T_{exe})：可執行服務的平均時間

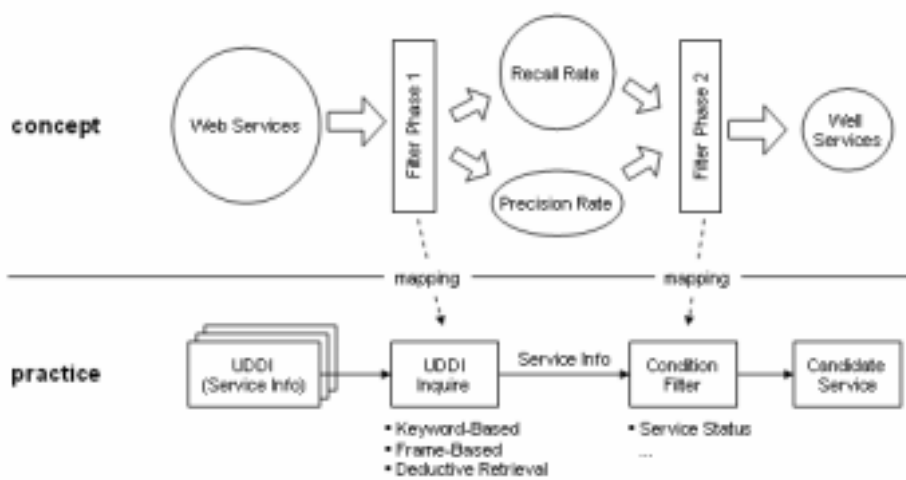


圖 2：過濾模型示意圖(概念與實行)

二、評選模型

評選模型是對前述模型已過濾所產出的結果，接續以準則項目逐一評選，此機制中，必須要先定義品質模式，其內容為評估進行時所需取得的要度量項目。另一方面，由於影響品質的項目種類很多，本研究在評估方法上採用多屬性決策的精神，亦即以加權方式及偏好權重來運算，以加權方式及偏好權重來運算，即簡單加權法，其包含決策過程簡單及方便使用之優點，決策者會將候選服務的分數由各準則計算值與其相對權重相乘，以得到每一個候選服務的加權評估值，依據分數的高低來進行網路服務優劣比較，以獲得品質參考指標，並做為服務排序的依據。在FLQR的設計模型中，簡單加權法亦可由其他方法替代，如多準則決策法。

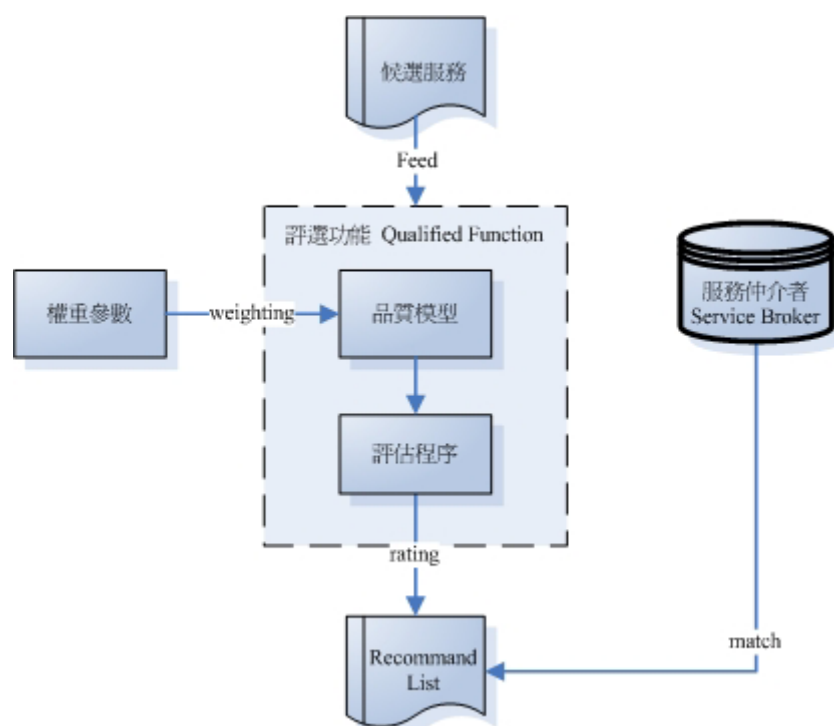


圖3：服務評選模型示意圖

如圖3的服務評選模型示意圖，其運作流程如下：(1).依查詢條件由品質資料庫取得候選服務；(2).取得使用者輸入之權重分配並與品質模型合併進行服務評估運算；(3)依品質總指標產出排序之推薦服務，並至服務仲介者取得原服務之登錄資訊。

本研究所使用的品質評估項目係參考Sheth(2002)、Zeng et al. (2003)、Stal (2002)及Ran (2003)等學者之研究建議所選定，並考量實際執行時的限制後，以可用度、可靠度、及平均時間等三項做為評估項目如表4。

表4：品質評估項目

評估項目	說明
可用度(Availability) (Q_{av})	可用度為服務可被呼叫的機率。 $Q_{av} = N_{success} / N_{total}$
可靠度(Reliability) (Q_{re})	可靠度為服務可完成工作的機率。 $Q_{re} = N_{complete} / N_{success}$
執行效能(Performance) (Q_t)	平均執行時間為反向指標，須先以施以標準化(M_t)再求取效能比。 $Q_t = 1 - M_t(T_{average})$
品質總指標 (Quality Index)	$Q_i = W_i Q_{av} + W_j Q_{re} + W_k Q_t$ $W_i + W_j + W_k = 1$ W表加權值

各評估項目最終須歸納為品質總指標，但由於執行效能是以平均時間為主，執行時間愈久所代表之意義為效能愈低，因此 $T_{average}$ 須先以標準化處理(N_t 即為計算該服務與其他服務的最大平均執行時間之比值)，最後 Q_t 再以反向指標取得統一的度量基準。本研究允許使用者在品質偏好賦予相對的權重值(Weight)，或使用簡單加權法計算候選服務的品質總指標，但加權僅為項目乘積分配的比率，各權值總和仍為1。

三、回饋模型

回饋模型是服務品質的學習機制，我們認為服務註冊中心應將使用之服務資訊及狀態記錄傳給FLQR品質評估模型，以做為更新某項服務的品質指標。回饋模型如圖4，主要工作包含：

- 透過回饋機制記錄服務使用情形，如執行時間、呼叫次數等資訊，並更新資料庫之品質資料。
- 若服務無法使用，透過回饋機制傳送訊息給服務品質評估代理人，作為優先修改依據。

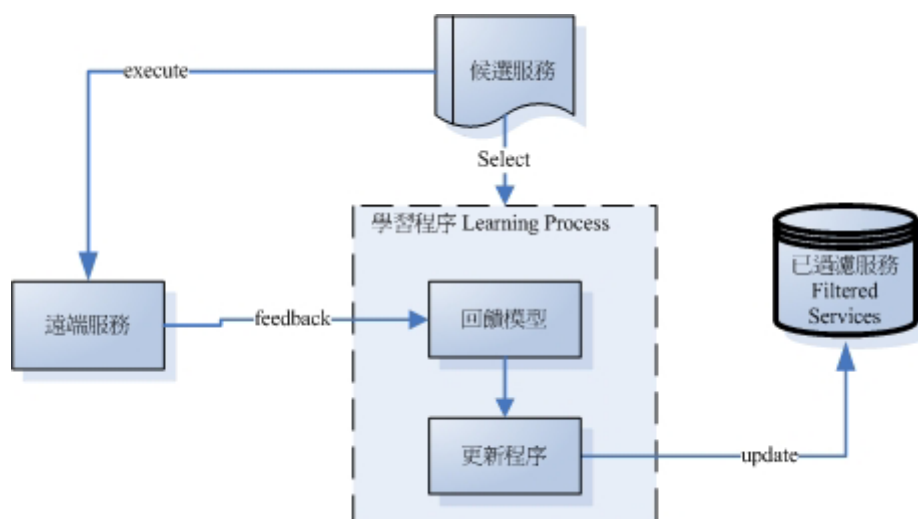


圖4：學習模式示意圖

當推薦的網路服務被需求量選用執行時，回饋機制同時也被起動，並依下列更新規則修正資料庫的品質資訊，規則如下：

$$T_{average} = \frac{T_{average} * N_{total} + T_{duration}}{N_{total} + 1}$$

$$N_{total} = N_{total} + 1$$

$$N_{success} = N_{success} + 1$$

$$N_{complete} = N_{complete} + 1$$

透過上述規則，FLQR模型可達到學習回饋之目的，此階段監測服務之使用情形亦可做為下一階段服務評選之參考。

四、濾選模型的系統建置

依照前述FLQR模型架構，本服務濾選系統須建置於服務註冊中心，主要功能包括使用者查詢介面、服務過濾功能、服務評選功能、及服務使用介面，我們採用的開發工具包括Java、Application Server、Database、及UDDI註冊中心建置套件等，所涉及之參考規範標準包括UDDI 3.0、WSDL 1.2、及WSIF2.0等，其整體系統之架構如圖5所示，各系統功能之建構流程說明如下：

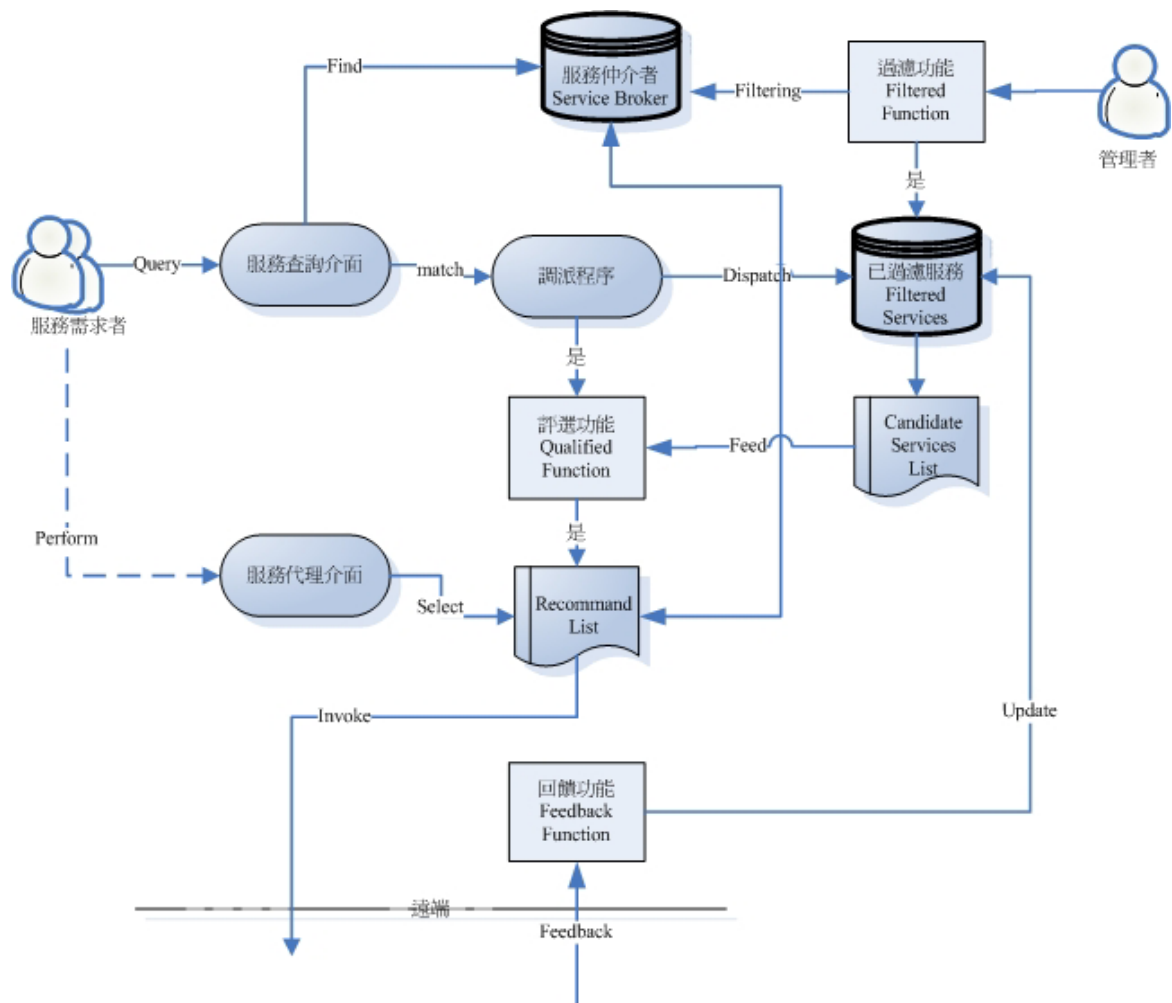


圖5：服務濾選系統架構

- 建置服務註冊中心：以 IBM ettk 來建置服務註冊中心，其性質為私有註冊中心，易於我們掌握管理。
- 服務查詢介面：參考 UDDI 規範，設計簡易的服務查詢介面，包含服務查詢位置及服務名稱，使用者依需求填寫後即可送出查詢請求。
- 服務過濾功能：開發應用系統實作服務的過濾，我們以「服務狀態」為篩選條件，以服務存取點(Access Point)為服務之主要作業位置，透過偵測方式來取得服務狀態，服務狀態以「●」表示此存取點為可連線，以「X」表示此存取點為中斷連線或是其他不可連線之情形。
- 服務評選功能：使用三個品質項目：可用度、可靠度、及平均時間。另外以下拉式選單供使用者賦予偏好之權重值。
- 服務代理者介面：當選用某一服務時，服務代理執行 SOAP 請求至對應的遠端服務，亦即透過此介面即可完成服務之執行及結果回應，本功能亦同時驅動回饋機制的運作。

在系統初次使用時，管理者應對服務仲介者所管轄的服務，實施全面性的過濾工作，偵測應選擇不同時間點並建議並至少實施3次以上，以建立初始的品質資訊，另可視系統的負載狀況，訂定後續固定的過濾工作。每次偵測工作應對個別服務執行5次以

上，而可連線之服務，須繼續完成執行成功次數及時間的記錄，各項數據依過濾模型之資料庫欄位予以登錄更新。評選工作是由服務需求者在下達服務查詢時發起，此時透過服務仲介者取得原登錄服務之Service Key，經由本研究之調派程序向資料庫調用相關服務的資料，並交付評選功能運算，產生推薦的排序名單。服務需求者可繼續在此平台上，參考推薦的排序名單，並選擇執行某一服務，而回饋機制同時利用執行的機會更新資料庫的記錄。

肆、驗證與討論

一、驗證分析

為驗證本研究之設計，我們以不同的開發方式製作具有相同功能的網路服務，並配置於不同的伺服器上，各服務之名稱、技術模型(tModel)、對應主機及狀態如表5，本研究共設計九個網路服務，其中技術模型以tModel-1至tModel-9表示，並發佈至服務仲介者的資料庫中。下列三項情境分別說明FLQR的一般使用情形、回饋機制、及偏好權重的功效。

表5：技術模型之設計

網路服務名稱	tModel	對應於主機說明	狀態
SP1-StockQuote	StockQuoteModel-1	於Server-1之網路服務	●
SP1-Stock	StockQuoteModel-3	於Server-1之網路服務	×
SP1-GetQuote	StockQuoteModel-4	於Server-1之網路服務	●
SP1-DelayedStockQuote	StockQuoteModel-6	於Server-1之網路服務	●
SP2-StockQuote	StockQuoteModel-2	於Server-2之網路服務	×
SP2-Stock	StockQuoteModel-5	於Server-2之網路服務	●
SP2-GetQuote	StockQuoteModel-7	於Server-2之網路服務	×
SP3-DelayedStockQuote	StockQuoteModel-8	於Server-3之網路服務	×
SP4-StockQuote	StockQuoteModel-9	於Server-3之網路服務	×

Case1：服務濾選

在過濾階段時以服務狀態為篩選條件，因此只有可用服務在「已過濾資料庫中」，我們獲得如表6之四項服務，亦即為評選程序之候選服務。在進入服務評選時，我們須指定服務需求之偏好權重，例如可用度為0.3、可靠度為0.4、平均時間為0.2，經計算及標準化後可獲得各項目之分數，再經權重計算獲得品質之總指標值，我們依指標值排列如下，第一項之SP1-StockQuote即為推薦之服務。

表6：服務評選

Service Name	Availability	Reliability	Time	Score
SP1-StockQuote	0.912	1	0.347	0.825
SP2-Stock	0.913	0.976	0.378	0.822
SP1-GetQuote	0.967	0.933	0.010	0.739
SP1-DelayedStockQuote	0.75	0.833	0.199	0.664

Case2：濾選系統之回饋機制

回饋機制涉及動態更新服務之品質資訊，表7承上例，其中SP1-StockQuote為推薦之服務，經使用者執行本服務後，各項數據更新於品質資訊之原始記錄中，例如本次服務之執行時間耗時1.602秒，故原服務平均使用時間由0.804秒增長為0.812秒，而服務被呼叫之次數、呼叫成功次數、執行成功次數均予累加1。因此，本服務經正規化及原權重比例下之計算調整，總指標值由0.825微降為0.824。

表7：服務評選

Service Name	Time(sec)	Invoke	Invoke Success	Execute Success	Score
SP1-StockQuote	0.812	92	84	84	0.824

Case3：品質偏好權重之調整

由於本研究設計允許使用者賦予各項目之權重，因此服務品質排序必然受到影響，表8為兩種不同偏好權重的比較，例如上半部份是以Availability= 0.8為主要權重，因此得到最高品質總指標為SP1提供之SP1-GetQuote服務。而表8下半部份是以Duration=0.8為主要權重，因此得到最高品質總分為SP2提供之SP2-Stock服務。調整權重僅影響該次服務查詢的排序，但不涉及資料庫的品質資訊異動，因此權重調整視為需求者的個人偏好。

表8：品質偏好權重之調整

Rank	Service Name	Score
1	SP1-GetQuote	0.868
2	SP2-Stock	0.865
3	SP1-StockQuote	0.864
4	SP1-DelayedStockQuote	0.703
<i>Availability= 0.8 , Reliability= 0.1 , Duration= 0.1</i>		
1	SP2-Stock	0.492

2	SP1-StockQuote	0.469
3	SP1-DelayedStockQuote	0.317
4	SP1-GetQuote	0.198
<i>Availability= 0.1 , Reliability= 0.1 , Duration= 0.8</i>		

二、討論分析

本研究所提出的FLQR模型是對於原服務註冊中心在品質評量上之輔助措施，具體的改善如(1).增加濾選機制減少評選範圍；(2).增加品質項目的訂定及權重分配，並以品質總指標做為服務選用依據；(3).以動態回饋機制保持品質項目的異動記錄。表9為原服務仲介者與加入FLQR模型輔助之比較：

表9：原服務仲介者與FLQR模型輔助之比較

	原服務仲介者	FLQR模型服務仲介者
服務資訊查詢	有提供	延用原服務仲介者
服務狀態篩選	無	有提供
品質項目參考	無	可自行加入品質項目
使用者參與評選	無	以權重指定方式
服務選擇問題	存在	以服務排序方式改善
服務使用追蹤	無	以服務代理者管理
品質指標修正	無	以服務代理者管理

本研究與過去研究的最大差異是在增加過濾及回饋機制上，過去網路服務品質之研究中，都以直接偵測並計算網路服務之品質項目來進行實驗，並未排除不可使用的網路服務，因此，不可使用之網路服務也會被納入服務評選之範圍，不但增加了評選機制運算、分析之時間，也會造成評選後，挑選到不可使用之服務機率提高。過濾機制強調如何先行剔除不具評選意義的服務，以減少評選機制運算、分析之時間，例如表6原在服務仲介者中有九項服務，但經過濾機制並以service key調派後，合格的候選服務僅餘四項，因此可減少需評選的來源。在新增的回饋機制中，服務代理者利用使用者執行服務之際，進行服務追蹤及修正其使用之情況，可確保品質資訊之及時無誤，此回饋階段得以調整服務的可用度，以減少挑到服務不可使用之機率，達到提升服務評選的實際效率。最後在評選機制上，本研究提供的品質項目大致與過去研究的相近，但增加了使用者的參與，亦即允許對不同品質偏好賦予權重，以解決不同情境下對品質偏好不同之問題。

伍、結論

網路服務架構可達成企業初步的資源共用與資料共享，但由於服務仲介者無法有效掌握服務提供者的軟體品質，因此商務應用時必須面臨服務仲介的問題。本研究提出兩階段的「先過濾再評選」構想(FLQR模型)，所包含之重點如下：(1).提供服務仲介者過濾選機制，在實務上可剔除不能用之服務。(2).評選服務提供品質項目及權重的計算，使候選服務能依品質模型予以推薦排序。(3).為使FLQR模型能動態更新品質記錄，回饋機制能將歷次使用資訊予以學習，並回饋至服務仲介者的品質庫中。本研究最後亦實際建置一個FLQR模型的服務仲介者，在實證中已初步達成我們設定的目標，因此FLQR模型確可做為建置服務仲介者(註冊中心)的改良參考。

參考文獻

- [1].Alameh, N.: "Chaining geographic information Web services," IEEE Internet Computing, 2003, 7(5), pp.22 -29.
- [2].Benatallah, B., Sheng, Q.Z., and Dumas, M.: "The Self-Serv Environment for Web Services Composition," IEEE Internet Computing, 2003, 7(1), pp.40-48.
- [3].Bevan, N.: "Quality in use: meeting user needs for Quality," Journal of System and Software, 1999, 49(1), pp.89-96.
- [4].Chiu, D.K.W., Cheung, S., Karlapalem, K., Li, Q., and Till, S.: "Workflow View Driven Cross-Organizational Interoperability in a Web-Service Environment," Lecture Notes in Computer Science, 2002, 2512, pp.41-56.
- [5].Conti, M., Kumar, M., Das, S.K., and Shirazi, B.A.: "Quality of Service Issues in Internet Web Services," IEEE Transactions on Computers, 2002, 51(6), pp.593-594.
- [6].Curbera, M., 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, 2002, 6(2), pp.86-93.
- [7].Dalal, S., Temel, M., Little, M., Potts, M., and Webber, J.: "Coordinating business transactions on the Web," IEEE Internet Computing, 2003, 7(1), pp.30-39.
- [8].Florescu, D. and Kossmann, D.: "An XML Programming Language for Web Services Specification and Composition," IEEE Data Engineering Bulletin, 2001, 24(3), pp.48-56.
- [9].Kou, W., Lauzon, D., O'Farrell, W., See, T.L., Wee, D., and Tan, D.: "End-to-end E-commerce Application Development Based on XML Tools," IEEE Data Engineering, 2000, 23(1), pp.29-36.
- [10]. Medjahed, B., Rezgui, A., Bouguettaya, A., and Ouzzani M.: "Infrastructure for E-Government Web Services Middleware for Web Services," IEEE Internet Computing, 2003, 7(1), pp.58 -65.
- [11]. Menasce, D.A.: "QoS Issues in Web Services," IEEE Internet Computing, 2002, 6(6), pp.72-75.
- [12]. Menasce, D.A.: "QoS-Aware Software Components," IEEE Internet Computing, 2004, 8(2), pp.91-93.

- [13]. Moorsel A., Metrics for the Internet Age: HP Labs., "Quality of Experience and Quality of Business," Technical Report HPL-2001-179, 2001.
- [14]. Naumann, F., Leser, U., and Freytag, J.C.: "Quality-driven Integration of Heterogeneous Information Systems," Proc. of the 25th Very Large Databases Conference, Edinburgh, Scotland, 1999, pp.447-458.
- [15]. Rabhi, F.A. and Benatallah, B.: "An Integrated Service Architecture for Managing Capital Markets Systems," IEEE Network Magazine, 2002, 16(1), pp.15-19.
- [16]. Ran, S.: "A Model for Web Services Discovery with QoS," ACM SIGecom Exchanges, 2003, 4(1), pp.1-10.
- [17]. Shegalov, G., Gillmann, M., and Gerhard, W.: "XML-Enabled Workflow Management for e-Services across Heterogeneous Platforms," The VLDB Journal, 2002, 10(1), pp.91-103.
- [18]. Sheth, A., Cardoso, J., Miller, J., and Kochut, K.: "QoS for Service-Oriented Middleware," Proc. of the Systemics, Cybernetics and Informatics Conference, Orlando, 2002, pp.528-534.
- [19]. Siegel, J.: "OMG overview: CORBA and the OMA in enterprise computing," Communications of the ACM, 1998, 41(10), pp37-43.
- [20]. Stal, M.: "Web Services: Beyond Component based Computing," Communications of the ACM, 2002, 45(10), pp.71-76.
- [21]. Sullivan, K.J., Marchukov, M., and Socha, J.: "Analysis of a Conflict between Aggregation and Interface Negotiation in Microsoft's Component Object Model," IEEE Transactions on Software Engineering, 1999, 25(4), pp.584-599.
- [22]. Vinoski, S.: "CORBA: Integrating Diverse Applications within Distributed Heterogeneous Environments," IEEE Communications Magazine, 1997, 35(2), pp.46-55.
- [23]. Vinoski, S.: "Web Services Interaction Models - Part 1: Current Practice," IEEE Internet Computing, 2002, 6(3), pp.89-91.
- [24]. World Wide Web Consortium (W3C): "Web Services Architecture," [Online] available at <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, 2004.
- [25]. 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, 2004, 30(5), pp.311-327.
- [26]. 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, 2003, 13(2), pp.141-152.

