

行政院國家科學委員會專題研究計畫 成果報告

針對網路服務之模糊化搜尋機制

計畫類別：個別型計畫

計畫編號：NSC94-2416-H-009-020-

執行期間：94年08月01日至95年07月31日

執行單位：國立交通大學資訊管理研究所

計畫主持人：羅濟群

計畫參與人員：程鼎元、王勇璋、林；平祺

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 95 年 10 月 26 日

行政院國家科學委員會專題研究計畫成果報告

針對網路服務之模糊化搜尋機制

The Study of Fuzzy Matchmaking for Web Service

計畫編號：93WFA0605493

執行期限：94年8月1日至95年7月31日

主持人：羅濟群

國立交通大學資訊管理研究所

計畫參與人員：程鼎元、王勇璋、林平祺

國立交通大學資訊管理研究所

中文摘要

網路服務架構是一種新興的異質平台整合方式，當所有網路服務都須由外部取得時，服務需求者將會面對許多性質類似的服務來滿足各種需求，雖然候選服務的數量越多，越能增加服務應用的彈性。但如何快速地找到適合的服務，卻是使用者必須面臨的難題。

若想發揮 Web Service 的強大應用能力，關鍵就是如何正確快速地找到所需服務。為了改善 Web Service 的搜尋效能，目前發展的技術有 UDDI、Semantic Web 等。UDDI 類似一種目錄服務，藉由 UDDI，服務需求者可找到所需服務，形成提供服務的單一進入點。Semantic Web 是一種能夠精確描述資源及資源間關係的 Ontology，提供 Web 資訊共用和交換的基礎。

關鍵字：網路服務，模糊理論，matchmaking
Abstract

When web service becomes popular, there will be more different and similar services. Users will be confronted with choosing the appropriate service. How to match the suitable service is important for users to use web service effectively. Various technologies have been developed in order to improve the matchmaking of web service. For instance, web services use UDDI which provides standard programming interface and communication protocols to resolve interoperability problems associated with integrating heterogeneous web services. Further, the semantic web is the ontology which describes the relationships between resources precisely and provides the communication between machines.

The above technologies do not address a

critical issue, which is associated with contents of web services. However, the development of fuzzy theory for library and information science inspired us to use fuzzy logic to handle terms for describing possibility, probabilities and quantifier to web service.

Keywords: Web Service, Fuzzy, Matchmaking
前言

網際網路從無到有的普及發展係和其無所不在的特性，而 Web Service 的興起更是將網際網路的應用帶向另一個紀元，因為 Web 不僅可以作為一個資訊平台，也將成為一個服務平台，此處的 Service 並不是指一般網路上所提供的訂票、購票等服務，而是一種元件的服務，可以讓其他人來建構更強大的服務元件。網際網路發展的趨勢也正朝向異質系統的整合，而 Web Service 的興起不僅解決了許多中介媒體會碰到的問題。

Web Service 是利用 Web 作為資訊發佈者，而其主要的優勢在於它的使用十分簡單，而且透過 Web 通訊協定與符合開放式標準，再加上利用開放式標準的資料格式，利用 XML 語言作為資料的表示、儲存，使得 Web Service 有良好的互通性，在不同的平台上使用不同的程式語言所建置出來的系統也可以輕鬆的整合在一起，藉以達成異質系統整合的目標，克服目前分散式系統建置時各自使用不同機制，而造成整合上會產生困難。

一、研究目的

網路服務架構是一種新興的異質平台整合方式，因此 Web Service 的應用數量將會越來越多，內容也會越來越完整。當所有網路服務都須由外部取得時，服務的需求者則將會面對許多性質類似的服務，以滿足各

種服務需求，陳列在服務仲介者的查詢名單中的服務提供者會越來越多，雖然候選服務的數量越多，越能夠增加服務應用的彈性。目前網路服務的尋找方法係利用 UDDI 或 Semantic Web，目前所採用這些方法並沒有強調 Web Service 的真實內容，而只是列舉與服務相關的功能和資訊的描述；再加上整個搜尋的過程中，使用者必須要明確定義所輸入搜尋字串，使用者需求通常並沒有使用明確的量化模式來查詢。如何能夠快速地找到適合的服務，使用者必須面臨的難題。

要解決上述的該問題，則可以使用模糊理論的方法，模糊理論過去在圖書資訊學中供了相當有價值的研究，其過程係透過將想要處理的資訊模糊化，然後再引進適當的語言變數來控制模糊資訊的推論程序。因此本研究係是利用模糊理論來表達 Web Service 內容，模糊理論的發展使的人們解決問題以及產生新的思維，模糊理論的應用相繼在自動控制、醫療診斷、圖像辨識等各領域中應用；而今，此方法引用到 Web Service 中，不僅符合網路服務的展現方式，而且可以減少搜尋的空間。對於服務需求者來說，便能夠使用模糊的字串進行搜尋，更能夠貼近人類的思考模式、加速搜尋的效能。本研究中，找出一套模糊推論系統來表達網路服務內容，在內容分析與搜尋規則 Match Making 查詢方式以增進查詢的能力。

二、文獻探討

1. 背景介紹

本節針對 Web Service、UDDI、SOAP、Semantic Web 四個主題作為本研究使用技術與環境之背景介紹：

(1) Web Service

網路服務技術基本上是沿襲分散式資訊系統的理念，只是更專注在Web上。而何謂網路服務技術呢？根據W3C的定義[1]，一個網路服務是指一個應用程式可經由XML來描述查詢，以及利用URI來辨識

此應用程式的介面與鏈結方式，均已被完整定義且支援其他的應用程式，並能藉XML形式的訊息，經合乎網際網路的協定來直接驅動。網路服務技術我們可將其視為是以服務為導向的架構，亦被稱之為SOA(Service Oriented Architecture)。

(2) UDDI

UDDI是提供用戶端在網路上動態尋找其他Web Service的機制，UDDI建置於網路傳輸層和基於SOAP的XML訊息傳輸層之上。諸如Web服務描述語言[3] (Web Services Description Language, WSDL) 之類的服務描述語言提供了統一的XML詞彙與互動式資料語言類似來提供描述Web服務及其介面使用。可以透過加入分層的功能搭起整個基礎，比如使用Web服務流程語言的Web服務工作流程描述、安全性、管理和服務品質功能，從而解決系統可靠性和可用性問題。

(3) 簡單物件傳送協定(SOAP)

簡單物件傳送協定(Simple Object Access Protocol, SOAP)[4]是一個以文字為基礎的通訊協定，可以藉由包裝在其他通訊協定傳送，例如：HTTP、SMTP、TCP、UDP...等，以充分達到在異質環境中達成系統整合的目的。

(4) 語意網路(Semantic Web)

語意網路[6]的運作主要技術是建夠在XML之上，而RDF則是在語意網路中作為描述資源的功用，使得文件的語意能夠讓機器所了解。要如何建立起有用的語意網路，就需要使用本體論語言來描述文件，例如使用DAML+OIL來表達。以下就對RDF與Ontology Language分別作介紹：

A. 資源描述架構, RDF

W3C所大力倡導的資源描述架構，簡稱RDF(Resource Description Framework)[7]，作為處理Metadata的標準模式，使得在Web上的應用程式間可以相互讀取、理解所交換得到資訊。RDF可以視為一種知識表達的方法，該方法描述某一物件為資源時，是將該物件表示成屬性(Property)與內容值(Value)的關係，也就是物件、屬性、內容值三元表示法(Object-Attribute-Value Triples)。

在 RDF 中並未定義任何的推論機制，只是藉由知識表達的方法來描述 Metadata，進而建構整個知識庫，讓系統開發人員可以很容易的在該知識庫上建立所需要的推論引擎，下圖為一個 RDF 表示的例子。

```
<rdf: RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:foaf="http://xmlns.com/0.1/foaf#" >
  <rdf: Description rdf:about="">
    <dc: creator rdf:parseType="Resource">
      <foaf: name>Sean B. Palmer</foaf: name>
    </dc: creator>
    <dc: title>The SemanticWeb: An Introduction</dc: title>
  </rdf: Description>
</rdf: RDF>
```

圖 1：以 RDF 方式表示之範例

B. DAML+OIL

DAML+OIL 係是由 DAML-ONT (Ontology) 與 DAML、與 OIL (Ontology Inference Layer) 三種語言合併而成的一個模式系統，主要目的是為了使 Web 的內容更容易存取與了解。DAML+OIL 使用更多用於描述屬性和類型的辭彙等機制來表達語意，提供使用者(人或機器)可讀的文件，在資源管理系統的建置中，可以藉由 DAML+OIL 來描述類別與子類別之間的關係，以及類別與類別之間關係上的限制條件。DAML+OIL 的文件是建立在 RDF Triples 語法為基礎所建構出的知識庫。

2. 研究主題介紹

本節針對內容分析、搜尋規則、Fuzzy 三個主題作文獻回顧。

(1) 內容分析

A. OWL-S

OWL 在 web service 上的應用，為了讓使用者的 software agent 讀得懂所提供的服務內容以及知道如何執行它，因此我們使用 OWL-S 去描述一個 web service。OWL-S 所要達到的任務為以下三個項目：

(a) 自動化的 discovery

在傳統的網站，如果要找尋賣書的服務，可能要透過網路上的搜尋引擎，然後找到網頁手動輸入要找的書本。在 semantic web 的世界中，service provider 可以用 OWL-S 描述自己所提供的服務，然後發佈到提供註冊的 registry(如:UDDI)。requestor 向 registry 下 query，registry 會回覆能提供要求服務的 web service。

(b) 自動化的 invocation

我們可以把 web service 看成是一些 remote procedure calls 的集合，當我們的得知遠端 OWL-S 的位置以及 input/output 為什麼，我們就可以透過工具來執行。執行方式我們可以想像成類似 IDL 的執行方式。



圖 2：Web Service Client 端與 Server 端呼叫

(c) 自動化 composition 與 interoperation

Service providers 之間可以透過 OWL-S 做服務的整合以及服務之間的互動。舉例來說，假設有一家旅行社提供查詢旅遊的 service，這個 service 可以搭配航空公司的查詢機票的 service 做結合。兩家公司也可以透過 web service 的互動取代傳統的 EDI。

(2) 搜尋規則(Match Making)

A. LARKS

LARKS 的過濾方式可以區分為：

- (a) 內文比對(Context matching)
- (b) 特性比對(Profile matching)
- (c) 相似比對(Similarity matching)
- (d) 比對(Signature matching)
- (e) 限制比對(Constrain matching)

B. 個案為基礎的推理, CBR

所謂個案為基礎的推理 (Case-Base Reasoning, CBR) 指的是組織過去解決問題的經驗，以案例的方式儲存於知識庫中。並利用這些知識來解決類似的新問題。

(3) 模糊集合理論(Fuzzy Set Theory)

在一般集合的觀念裡，如果是依類別的概念建立集合，元素與集合間存在著唯一且明確的歸屬關係。但是在現實世界中的元素與集合間卻往往無法用這樣唯一且明確的歸屬關係來界定。在模糊理論中，除了不採用類別的觀念去定義元素與集合的關係，讓相同的元素歸屬在不同的命題集合以外，並提出以歸屬函數在 0~1 間連續取值，來定義元素對集合歸屬程度。以下針對模糊集合理論中常使用的歸屬函數、基本運算單元、語言變數及 Hedge 分別作介紹。

A. 歸屬函數(Membership Function)

在整個集合中每一個元素對於某一模糊集合的歸屬程度都賦與一個介於0和1之間的數值，將此對應關係稱為歸屬函數或特徵函數(Characteristic Function)。歸屬函數形式會隨著使用者定義不同，須符合兩限制：

- (a) 歸屬函數的值域須在[0, 1]上；
- (b) 歸屬函數必須是一對一函數。

以下介紹四種在 Fuzzy Set Theory 中常被使用的歸屬函數：

- (a) 梯形(Trapezoidal)歸屬函數

梯形歸屬函數具有四個參數 $\{a, b, c, d\}$ ，函數如下：

$$\mu_A(a) = \begin{cases} \frac{x-(c-b)}{b} & c-b \leq a \leq c \\ 1 & c \leq a \leq d \\ \frac{-x+(d+b)}{b} & d \leq a \leq d+b \\ 0 & a > d+b, a < c-b \end{cases}$$

- (b) S 歸屬函數

S 歸屬函數是具有三個參數 a, b, c 的平滑歸屬函數，如下：

$$\mu_A(a) = \begin{cases} 0, & a \leq b \\ 2\left(\frac{a-b}{d-b}\right) & b \leq a \leq c \\ 1-2\left(\frac{a-b}{d-b}\right)^2 & c \leq a \leq d \\ 1 & a > d \end{cases}$$

- (c) Z 歸屬函數

Z 歸屬函數是具有三個參數 a, b, c 的平滑歸屬函數，如下：

$$\mu_A(a) = \begin{cases} 0, & a \leq b \\ 2\left(\frac{a-b}{d-b}\right)^2 & b \leq a \leq c \\ 1-2\left(\frac{a-b}{d-b}\right)^2 & c \leq a \leq d \\ 1 & a \geq d \end{cases}$$

- (d) 高斯(Gaussian)歸屬函數

高斯歸屬函數由兩個參數定義 $\{m, \sigma\}$ ，如下：

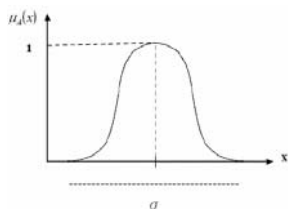


圖 3：高斯歸屬函數

B. 基本運算單元

知悉模糊集合中的歸屬函數後，對應模糊集合中的運算也可以定義，類似普通集合

中運算的聯集、差集與補集三種運算：

- (a) 聯集(Union)運算

兩個模糊集合的聯集是兩個集合中取較大者，定義如下：

$$\forall x \in U : \mu_C = \max[\mu_A(x), \mu_B(x)]$$

- (b) 交集(Intersection)運算

兩個模糊集合的交集是兩個集合中取較小者，定義如下：

$$\forall x \in U : \mu_C = \min[\mu_A(x), \mu_B(x)]$$

- (c) 補集(Complement)運算

補集運算相當於否定，兩個模糊集合的聯集定義為：

$$\forall x \in U : \mu_{\bar{A}} = 1 - \mu_A(x)$$

C. 語言變數

為了使電腦能夠運算在字詞而不是數值，美國加州柏克萊大學的查德教授(Professor A. Zadeh, UC Berkeley)在 1973 年提出語言變數的觀念，所謂語言變數就是變數的值是自然語言中的字詞或句子。有語言變數 [21] 就可以將自然語言轉譯成邏輯數值型的敘述，建構模糊推理。語言變數可分為三類：

- (a) 量化(Quantification)變數
- (b) 頻率(Usuality)變數
- (c) 可能(Likelihood)變數

D. Hedges

在自然語言中，形容詞常常用來修飾名詞，例如：非常便宜，“非常”就是修飾便宜的形容詞。在模糊理論裡，我們將這樣的形容詞稱之為 Hedges，也就是模糊數值的修飾詞：亦即歸屬函數的值若是加了 Hedges 修飾之後，將會有系統的將原歸屬函數的值做轉換，也將會產生不同的模糊集合。Hedge 可分為以下五種類型：

- (a) 集中度的 Hedge

例如，很多。原有歸屬函數的值經由集中度 Hedge 修飾後會變小。以數學式 $\mu_{A'}(a) = \mu_A(a)^p$ 表示，其中 $p > 1$ ； A' 是模糊集合 A 經集中度 Hedge 修飾模糊集合。

- (b) 擴張度的 Hedge

例如，有點。原有歸屬函數的值經由

擴張度 Hedge 修飾後會變大。以數學式 $\mu_{A'}(a) = \mu_A(a)^{1/p}$ 表示之，其中 A' 是模糊集合 A 經擴張度修飾後的模糊集合。

(c) 強化對比的 Hedge

例如，極端。原有歸屬函數的值經由對比強化 Hedge 修飾後，原有值小於 1/2 會變小，大於 1/2 會變大。以數學式表示為

$$\mu_{A'}(a) = \begin{cases} 2^{p-1} \mu_A(a)^p & \mu_A(a) \leq 1/2 \\ 1 - 2^{p-1} (1 - \mu_A(a))^p & \mu_A(a) > 1/2 \end{cases} \text{ 其中 } p > 1; A'$$

是模糊集合 A 經集中度的 Hedge 所修飾後的模糊集合。

(d) 不明確的 Hedge

Hedge: 例如，不常。和對比強化 Hedge 相反。一般用下列數學式表示

$$\mu_{A'}(a) = \begin{cases} \sqrt{\mu_A(a)/2} & \mu_A(a) \leq 1/2 \\ 1 - \sqrt{(1 - \mu_A(a))/2} & \mu_A(a) > 1/2 \end{cases}$$

其中 A' 是模糊集合 A 經集中度的 Hedge 修飾後的模糊集合。經由不明確的 Hedge 修飾後的模糊集合將更加模糊化，一般常運用在將明確的集合轉成模糊集合。

E. 可能性模糊關係語言, PRUF

在本研究中採用一種自然語言的意義表達語言，叫作可能性模糊關係語言 (Possibility Relational Universal Fuzzy, PRUF)，這種語言的框架是模糊集合論的創始人 L.A. 扎德所提出來的。是用來表示自然語言中的模糊查詢，PRUF 主要由一系列的轉換規則和推論規則所組成的，在 PRUF 中最主要的轉換規則有下列幾個類別：

(a) 修飾規則(Modification Rules)

$$P^+ \triangleq N \text{ 是 } m \tilde{F} \text{ 的翻譯寫為 } N \text{ 是 } m \tilde{F} \rightarrow \prod_{(X_1, \dots, X_n)} = F^+$$

(b) 合成規則(Composition Rules)

合成規則是用在翻譯 $P = q^* r$ 這類命題，其中 $*$ 是合成運算，例如：and、or。

(c) 量限定規則(Quantification Rules)

量限定規則也稱為量詞規則，用來翻譯經過模糊量詞修飾過的命題，這種命題可以寫成

$$\tilde{Q} = \left\{ \left[\Pr(\tilde{F}/\tilde{G}), \mu_{\text{most}}(u, v) \right] \mid u \in \tilde{F}, v \in \tilde{G} \right\} \text{ 其中在}$$

\tilde{G} 中的部份 \tilde{F} 記為 $\Pr(\tilde{F}/\tilde{G})$ ，且 $\Pr(\tilde{F}/\tilde{G}) = \frac{\tilde{F} \cap \tilde{G}}{\tilde{G}}$

(d) 質限定規則(Qualification Rules)

$$\text{cons}\{N \text{ is } F \mid N \text{ is } G\}$$

$$\triangleq \text{Poss}\{N \text{ is } F \mid N \text{ is } G\}$$

$$= \sup_{u \in U} \left\{ \min(\mu_F(u), \mu_G(u)) \right\}$$

其中 Poss 是指可能性，sup 是指支持度。例如：“X 很便宜”是非常可能的。

三、採用之研究方法與步驟

1. 研究方法

本研究採用研究方法是建立一系統平台進行試驗，提供網路服務查詢環境，透過模糊推論引擎及模糊規則庫的推論，到網路服務資料庫中查詢滿足使用者需求的條件，經過解模糊化後，推薦適合的服務給使用者。

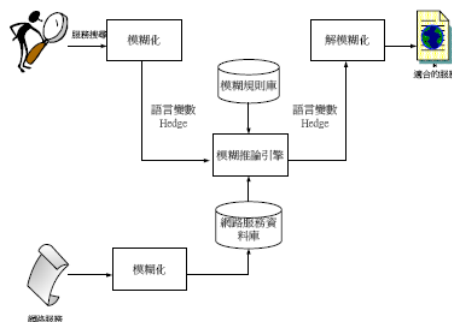


圖 4：本研究架構

A. 系統平台描述

系統平台包括模糊推論引擎、語義庫、分類單元、服務資料庫以及查詢介面等元件。

B. 實驗環境描述

實驗環境是模擬航空業者提供查詢機票票價與訂票服務，使用者利用 Web Browser 連線，輸入查詢條件，例如：便宜的機票、舒適的機票...等自然語言查詢，有別於傳統關鍵字查詢搜尋方法，更能貼近使用者意向，推薦給最符合使用者需要的網路服務。

2. 研究步驟

本研究以所設計系統平台為基礎，輔以其他程式語言及軟體工具支援發展，從網路服務資訊的建置及內容等建立、利用模糊理論將處理資訊模糊化，提供使用者查詢所需

的服務。

實作步驟如下所描述：

步驟 1：首先，所有 Web 服務被他們的 OWLS 描述的服務提供者記錄在 UDDI 中。

步驟 2：模糊的分類者詢問在 UDDI 裡資料庫記錄的 Web 服務。

步驟 3：模糊的分類者使用 OWL 解釋不同的資料庫裡的每個 Web 服務。

步驟 4：模糊的分類者詢問 Fuzzy Engine 以計算不同 Fuzzy 項的模糊值。

步驟 5：模糊的分類者問模糊發動機可計算模糊價值那兒模糊適合不同模糊條件確定。模糊的分類者透過 OWL 解釋不同的資料庫並且透過模糊邏輯描述 Web 服務的抽象描述。模糊分類把 Web 服務歸類為不同的模糊的條件的代表。因此模糊的分類者記錄 Web 服務適合在 UDDI 的各種各樣模糊條件的模糊價值搜尋。

步驟 6：當查詢被處理時，模糊的介紹分析質問句子，並且請求 UDDI Registry 和模糊的引擎獲得結果。研究步驟如下圖：

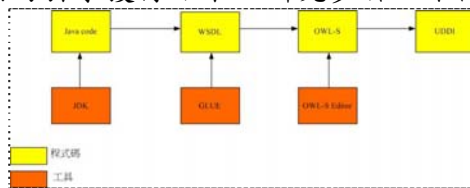


圖 5：實作研究步驟圖

安裝 JDK，再利用文字編輯器撰寫 java code。我們以提供輸入你的名字，然後回傳”hello name”作為例子，所寫的 java code 如下：

```
String HelloName(String yourname){  
    {  
        String hello = "hello" + yourname;  
        return hello;  
    }  
}
```

圖 6：Code Example

首先 compile 成 class 檔，再利用 Glue deploy 在 Glue server 上，Glue 會自動把 java code 轉成 WSDL。我們再利用 OWL-S Editor 編寫我們需要的 OWL-S，最後把 OWL-S profile 註冊到 UDDI Server。

A. 設計網路服務

我們使用的工具用法介紹如下：

(a) GLUE

首先撰寫提供 service 的 java code，例

如：可以查詢機票的程式碼，然後 compile 成 class 檔。再撰寫一支 java code 把要提供的 web service deploy 到 GLUE server 上，GLUE 會自動把 java code 轉成 WSDL 文件。requestor 知道 WSDL 的 location 之後即可以透過 GLUE client 執行 WSDL。

(b) OWL-S Editor

可以透過兩種方式撰寫 OWL-S 文件，第一種方式是 OWL-S editor 的 import WSDL 的功能，只要告知 WSDL 的 location，OWL-S editor 會自動把 WSDL 轉成 OWL-S。第二種方式先撰寫 Service Model 的 atomic 或 composite process，再把與 WSDL 的 mapping 關係寫在 grounding 部分。最後再把 OWL-S profile 撰寫出來，OWL-S profile 是來提供做 web service discovery。

OWL-S 描述一個 web service 的哪些，有三個部份如下：

(1). what it does

OWL-S 使用 ServiceProfile 去描述這個服務是做什麼服務，例如：查詢機票。ServiceProfile 提供三種資訊，分別為 1. 什麼組織提供這個 service。2. 這個 service 的功能。3. service 一些特殊的特徵。第一種資訊包含聯絡者的名稱、e-mail、電話。第二種資訊包括 service 的 input、output、precondition、effect。舉查詢機票的 service 為例，input 為飛機的起飛時間地點，output 為可訂購的飛機票，precondition 為可用的信用卡帳號，effect 為信用卡刷了一筆資料。第三種資訊為 service 的一些特徵，例如：UNSPSC 的分類、service 的 quality rating、地理上的限制等。

(2). how to access it

OWL-S 利用 ServiceModel 去描述 client 如何跟 service 互動，分類為 atomic process、simple process、composite process。舉例來說查詢機票為一個 atomic process，composite process 為多個 atomic process 所組成，例如：可能為查詢機票、機票訂購、飯店預訂所組合成的一個 process。process 描述兩種資訊，第一種為資訊的流動，輸入哪些資訊以及產生哪些資訊。第二種資訊為對世界產生的影響，

例如：effect。

(3). how it works

OWL-S 利用 ServiceGrounding 去描述如何把抽象的資訊轉換成實際的 message 交換。因為 OWL-S 沒有提供像 XSD 這樣的語意，所以利用一些抽象的 class 去描述 binding 的資訊。這些抽象的 class 透過對 WSDL 的描述來提供 binding 的語意。OWL-S 的 inputs/outputs 與 WSDL 的 input messages/output messages 在有些情況下是一對一的關係，有些情況不是。兩者的對應如下圖：

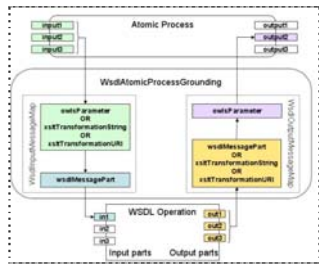


圖 7：OWL-S 與 WSDL 的 mapping

由上圖可知，多個 Atomic Process inputs 可以對應至一個 WSDL input message，而一個 Atomic Process output 可以對應多個 WSDL output messages。

B. 架設服務資料庫(UDDI Server)

使用 Apache 開放式軟體的 jUDDI 專案建立服務資料庫。

C. 利用 OWL-S 建立服務內容

利用 OWL-S editor 建立 OWL-S 以及將 OWL-S 的資訊新增至 jUDDI 中。Semantic web 是為了發佈、查詢、執行 web service 過程能夠自動化，減少人力的涉入，使用一種 Ontology Language 叫做 OWL-S 的語言來描述 web service。OWL-S 可以分成三個部份，分別為 service profiles、service models、service grounding：

(a) service profiles

提供人類可以閱讀資訊，例如：公司資訊，聯絡者，產業別服務提供者、該 service 所提供的功能、額外的資訊，例如：QoS。

(b) service models：提供 web service IOPE 的資訊。

(c) service grounding：介紹如何 Binding 方法，描述與 WSDL mapping 關係。

本研究中撰寫 OWL-S 時所使用的工具是 OWL-S editor²，它是 protégé³ 的 plugin。OWL-S 的執行我們所使用的工具是 OWL-S API⁴。OWL-S API 為 java API 可供撰寫 java 程式去執行 OWL-S。OWL-S 與 UDDI 的 mapping 關係圖如下：



圖 8：OWL-S 與 UDDI 的 mapping

OWL-S 有些 element 可以直接存進 UDDI，而有些無法直接存的 element 則存進 tModel。在 OWL-S 存進 UDDI 之後，即可以靠 tModel 的查詢做語意的推論。

D. 模糊系統處理 Web Service 資訊

Fuzzy classifier 的目標是為了把 web service 的資料表達成 fuzzy value，以及分類為數個 fuzzy sets。執行的步驟如下：

- 首先從 web service 的資料庫抓資料，因為資料庫的欄位名稱沒有統一，不同名稱的欄位可能表示的意思相同，例如一家航空公司某個欄位叫 price，另一家航空公司有個欄位叫 cost，兩個欄位其實都代表飛機票價，因此我們透過 OWL 建立一顆 Ontology tree，來得知兩個欄位表示的語義是相同的。這個部分所使用的工具為 OWLJessKB。
- 計算 fuzzy value，透過 Fuzzy Engine 得知合適的 fuzzy membership function 來算 fuzzy value。舉例飛機票價為 15000，利用相關的 membership function 來計算出與 cheap、medium、expensive 三個

² <http://owlseditor.semwebcentral.org/>

³ protégé 是由 Stanford 大學所開發可以用來撰寫

OWL-S 的編輯工具 (<http://protege.stanford.edu/>)

⁴ <http://www.mindswap.org/2004/owl-s/api/>

fuzzy terms 的關係，所求出來的值可能為 0.8cheap、0.5medium、0.2expensive。

(c) fuzzy classifier 把算出來的 fuzzy values 儲存在 UDDI server 提供 UDDI 查詢用。

E. 建立推論規則庫

將各類經過模糊化的網路服務內容存於資料庫，做為網路服務搜尋資料來源。

F. 建立推論引擎

將網路服務使用者的輸入模糊化後，再根據模糊規則，進行模糊推論，得出的推論將從現有的網路服務資料庫中，找出最適合的服務來滿足使用者的需求。

JESS (Java expert system shell) 為 java 環境上所提供的一種 Rule-Based system。expert system shell 為 inference engine 再加上一些 expert system 的功能。Rule Engine 可以視為 expert system shell 上可以撰寫程式的特殊環境，Jess 可視為此類 Rule Engine。你只要在 Jess 所提供的命令列打進指令，即可出現所要獲得的結果。Jess 包含以下三個部份：

(a) Inference Engine

為 rule engine 的核心，推論引擎所做的事情如下：

決定哪些已經 apply working memory 的 rules 被 activate；列出哪些即將執行已經被 activate 的 rules；Inference engine 可以分成三個部份：Pattern matcher、Agenda、Execution engine

(b) Rule Base

存放所有可被系統認知的 rules，這些 rules 通常以 string 的資料型態存放在 txt 檔，然後利用 rule compiler 去處理這些 rules。可以把這些 rules 組合成一種特別的資料結構叫 Rete network，這樣可以加快 rule compiler 的處理速度。

(c) Working Memory

我們也可以稱它為 fact base，存放 rule-based system 運作所需要的所有資訊。

JESS 的架構圖如下：

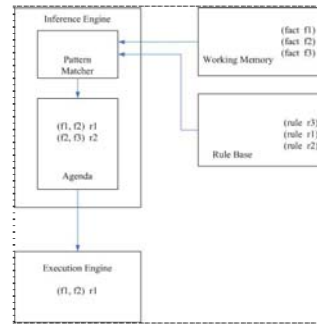


圖 9：JESS 架構圖

Jess 利用 Jess facts 當作它的 working memory。如果要利用 Jess facts 存放學生資料首先先定義 template 如下：

```
(deftemplate student(slot studentid)(slot name)(slot phone))
```

假若要輸入一筆為學號 001、名字 netlab、電話 12345 的資料，我們可以利用 assert function 來輸入，Jess 利用 Jess Rules 做推論，Jess Rules 利用 if...then... 做描述來做推論。假設我們想用 Jess Rules 制訂一條規則，假如 facts 有學號 001、名字 netlab、電話 12345 的這筆資料，則輸出 "Welcome, netlab"。

G. 解模糊化

此區塊在於將模糊推論所得的推論結果量化。因為我們要實際應用於實務系統，就必須將模糊推論所得到的輸出語言變數的語詞轉化成一個明確的輸出值。也就是在一個定義於字集 U 的模糊集合 A 中，找出最適合代表模糊集合 A 的明確值 $y^* \in U$ 。因此，網路服務搜尋所得的推論結果，將還原成原本的數值形式，譬如：房間舒適程度高，表示房間大於 10 坪，距市區距離小於 5 公里。

H. 實作網路服務搜尋

Client 端系統提供一個 Web 的查詢介面提供給使用者查詢在 Registry 中的 Web Service，此部份實作利用 UDDI4J 撰寫。

使用者可以上網以瀏覽器輸入模糊語意的字串(不是以傳統關鍵字查詢)進行查詢，利用 Fuzzy Discovery 查詢在 UDDI DB Server (Registry) 中服務業者的 Services。經 JESS 推理引擎推論後得到的 QoS 將會儲存在 UDDI Server 中，此時可以做為 Service Discovery 的參考值，回傳的服務

資訊應該提供：公司名稱，網址，公司服務描述，服務內容描述…等資訊(Service Description)。

四、實證結果與討論

1. 實證結果

上述所討論之本研究過程係建置一個網路服務的查詢環境，以航空公司所提供的訂票服務為模擬環境。將航空公司的資料建置新增至 UDDI Server 中，並提供一般 UDDI 查詢。

系統起始畫面為提供輸入查詢的 Web Based 的畫面，系統畫面如下圖所示，我們可以從 BusinessEntity 的 name 開始進行查詢：



圖 10：提供 Web Services 查詢介面

A. 結果-1



圖 11：查詢得到的 Business entity 資訊
點選 BusinessName 獲得 Contact Information



圖 12：點選某廠商後得到的連絡資訊

B. 結果-2

點選 BusinessKey 獲得 Service 的 Data



圖 13：點選某廠商後得到的商務資訊

2. 討論

本研究成果分成以下兩部份探討，其中分別針對內容分析與搜尋規則進行探討。

A. 討論-1：搜尋規則

可以使用相同字義的形容詞來查詢，例如：輸入 cheap 時，是和 low-priced 是相同意義的。

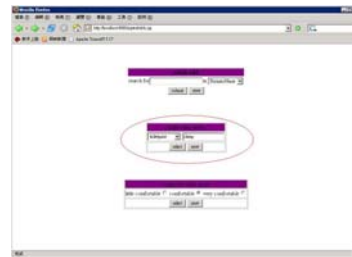


圖 14：同義字辨析

同時，也可以允許打錯字，例如少一個字元(chea)時，亦可以查詢到與 cheap 相同意義的查詢條件。

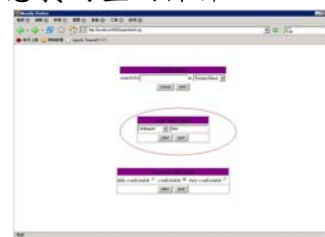


圖 15：錯字辨析

當使用者不小心輸入錯字時，將會將該字串更正後，再進行查詢，下圖是顯示查詢結果。

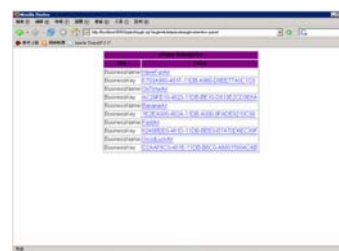


圖 16：查詢結果

B. 討論-2：內容分析

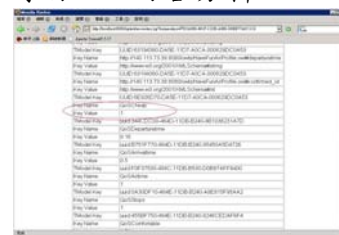


圖 17：HaveFunAir 服務資訊

以”便宜”這個關鍵字做為查詢條件為例，我們將便宜經過前述的解模糊化等步驟，計算出屬於便宜這個關鍵字的服務品質(Quality of Service)，當使用者想要找尋便宜的機票時，系統就會將已經分類、存在於UDDI Server 中的資訊回應給使用者。本研究設計是將 QoS 值越高放在越上面，通常使用者想要得知的資訊亦是由上往下檢閱。

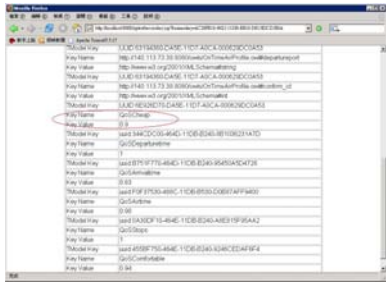


圖 18：OnTimeAir 服務資訊

五、結論與未來方向

由於目前網路服務眾多，搜尋機制又多依賴框架方法，如 UDDI，或語意網路，缺乏一種將網路服務內容真實呈現的表達方法，希望未來在服務提供者身份確認與交易的安全性能夠有更好的保障。

六、計畫自評

本研究提出利用模糊理論處理服務內容，透過語言變數來處理使用者搜尋，藉此建立一個網路服務模糊搜尋系統，對參與人員不止學習到相關的技術，在標準規格的研習上也有更深刻的學習，除此之外，希望在未來的查詢方法上都能夠有新的方法加入。本研究建立一個網路服務模糊搜尋系統，來解決使用者在面對許多網路服務時，能夠使用更貼近人類的思考模式，有效找到適合的服務。

七、參考文獻

[1] D. Booth and etc... “Web Services Architecture: 1.4 What is a Web service?,” W3C, 11 Feb., 2004, online source: <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis> .

[2] Edited by L. Clement, A. Hatley, C. von Riegen, and T. Rogers, ” UDDI Version

3.0.2,” OASIS, 19th Oct., 2004, online source: <http://uddi.org/pubs/uddi-v3.0.2-20041019.htm>

[3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, “Web Services Description Language (WSDL) 1.1,” W3C, 15th Mar., 2001, online source: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[4] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, and H. F. Nielsen , “SOAP Version 1.2 Part 1: Messaging Framework,” W3C, 24th June, 2003, online source: <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>

[6] Scientific American, "The Semantic Web", 17 May 2001

[7] Dan Brickley, Brian McBride, “RDF Vocabulary Description Language 1.0: RDF Schema”, 10 February, 2004

[8] “APPENDIX A – Fuzzy Sets, Logic and Control”http://vlab.unm.edu/documents/GA_Book_Appendix_A_Go.doc

[9] Graham Klyne, Jeremy J.Carroll, “Resource Description Framework (RDF): Concepts and Abstract Syntax”,W3C, 10 February 2004

[10] JESS IN ACTION, Ernest Friedman-Hill.

[11] “模糊邏輯及其應用”，全華，李允中等著，2004

[12] “知識工程中自然語義的模糊表達”，亞東，陳國權編著，1992

[13] ” 認識 Fuzzy”，第三波，王文俊等著，1994

[14] “模糊數學原理及應用”，儒林，張先迪、區奕勤編著，1991