

# 以知識本體為基礎 建構旅遊行程語意網路服務系統

邱英華<sup>1</sup>陳志豪<sup>2</sup>梁家松<sup>3</sup><sup>1</sup>南華大學資訊管理研究所<sup>2</sup>京元電子股份有限公司資訊技術處<sup>3</sup>南華大學資訊管理研究所

## 摘 要

由於網路的普及，國內旅遊業者大都有建置本身的網站，以提供旅遊行程資訊。然而，目前國內尚無完善的網站整合各家旅遊業者所提供的行程訊息，以致消費者通常要花費很多時間去搜尋、比較各家旅遊業者所推出的旅遊行程。為了解決此一問題，我們利用知識本體(Ontology)與語意網路服務(Semantic Web Services)技術來整合各家旅遊業者所提供的旅遊行程訊息，開發出以知識本體為基礎之旅遊行程語意網路服務系統(Itinerary Semantic Web Service System, ISWSS)。在 ISWSS 系統中，我們利用 OWL-S (Web Ontology Language for Service, 網路服務知識本體語言)來描述網路服務的知識本體，以達到網路服務自動化搜尋與調用的能力。另外，我們藉由推論引擎(Inference Engine)和匹配器(Matchmaker)來提升資訊檢索的效率與準確性，並且由 OWL-S 虛擬機器(Virtual Machine)來達到自動調用的功能，以便讓網路服務更為人性化。

**關鍵詞：**語意網路服務、知識本體、OWL-S、推論引擎、匹配器



# Using Ontology to Construct an Itinerary Semantic Web Service System

Yin-Wah Chiou<sup>1</sup>

Chih-Hau Chen<sup>2</sup>

Chia-Sung Liang<sup>3</sup>

<sup>1</sup> Department of Information Management, Nanhua University

<sup>2</sup> Information Technology Division, King Yuan Electronics Co., Ltd.

<sup>3</sup> Department of Information Management, Nanhua University

## Abstract

Since the Internet is popular, most tourism enterprises construct their own websites to offer the itinerary information. At present, however, there is no complete website to combine the itinerary information of every tourism enterprise. This problem causes consumers usually to spend much of time for searching and comparing the itinerary information. In order to solve this problem, we develop a system based on the *Ontology* and *Semantic Web Services* to integrate itinerary information which is provided by every tourism enterprise. We call this system as an *Itinerary Semantic Web Service System (ISWSS)*. In ISWSS, we use *Web Ontology Language for Service (OWL-S)* to describe the Web Service Ontology for reaching automatic discovery and enactment Web Service. The *Inference Engine* and *Matchmaker* are used to increase the efficiency and accuracy of information retrieval. In addition, the OWL-S Virtual Machine is employed to reach the automatic enactment for promoting the personalization of Web Services.

**Keywords:** Semantic Web Services · Ontology · OWL-S · Inference Engine · Matchmaker



## 壹、導論

近幾年來，國人旅遊風氣盛行，越來越多人是在假日期間安排旅遊活動。而隨著網際網路的發達，旅遊業者紛紛成立本身的網站，提供自身所推出的旅遊行程以供消費者查詢。以過去的使用者經驗而言，消費者必須至各家業者的網站逐一查詢與比較；在這過程裡，消費者無形中花費了過多的時間與精神於比對資訊與選擇決策等行為上。由於目前尚無完善的網站以整合、比對各家業者所提出的旅遊行程資訊，因此消費者較不容易從中比較、蒐集進而發現自己所需的資訊。綜觀上述，我們歸納出傳統旅遊業者提供網路服務時，消費者會遭遇到的問題如下：

- 消費者對於繁瑣的網路旅遊資訊進行蒐集與篩選會有所缺失與不足。
- 消費者得付出相當多的時間和成本來進行其本身與眾多業者所提供的旅遊行程做比較與評估。
- 消費者常因許多限制因素(如：時間、空間)，而無法獲得最佳的選擇訊息。

為了解決上述的諸多問題，我們透過網路服務(Web Services)的應用，藉由註冊機制(UDDI, Universal Description, Discovery, and Integration)來協調供需之間的互動，以便達到整合各家旅遊業者所提供的旅遊行程資訊。另外，我們亦透過知識本體(Ontology)的使用，並且利用語意網路服務(Semantic Web Services)的技術，使得網路服務可以達到自動發現、整合與自動調用的目的；讓電腦可以了解消費者的需求，依照消費者輸入

的資訊進行搜尋與比對進而呈現給消費者最合適的資料。

所謂語意網(Semantic Web)即是讓電腦能夠理解現有網路資料的內容形式；而知識本體則是概念化的一個形式規格用來說明特定領域知識的架構(Gruber, 1993)。因此，知識本體是實現語意網的重要關鍵技術。Eriksson(2007)提及語意網能夠使文件內容格式轉變成為讓電腦能夠更容易理解的資料型態，並且將資料意義定義的更為明確，以方便溝通和交換。將語意網用於網路服務中，可帶來清楚的描述資源及資源之間的關聯性，其所代表的意涵是網際網路不再只是單純的訊息、知識，而是具有推論能力的智慧架構。而語意網路服務既是整合了語意網與網路服務這兩種技術。

本文所提出的旅遊行程語意網路服務系統(Itinerary Semantic Web Service System, ISWSS)，其主要目的是透過語意網路服務的應用與知識本體的建構，來加強資料檢索的效能，讓 ISWSS 系統能運作地更為順暢，以期真正能提供符合消費者的需求及服務。本文的主要貢獻簡述如下：

- 整合國內各家旅遊業者所推出的旅遊行程，當註冊中心的資料數量越加龐大時，ISWSS 系統所能查詢到的訊息便越趨於完整、詳細。
- 透過知識本體與匹配器(Matchmaker)的應用達成更高的搜尋效率與準確性。
- 利用 OWL-S(Web Ontology Language for Service)虛擬機器讓系統能達到自動調用的功能，以便讓網路服務更為人性化。



關於本文其餘部分的組織結構，我們首先在第貳節文獻探討裡介紹 ISWSS 系統所涉及的相關理論與技術，包含：語意網路服務、知識本體、OWL (Web Ontology Language)、OWL-S、OWL-S/UDDI 匹配器、OWL-S 虛擬機器 (Virtual Machine)。其次，我們在第參節描述系統分析與設計，包含：需求分析、旅遊行程知識本體、ISWSS 系統架構、使用者查詢服務流程、服務提供者註冊流程。之後，我們在第肆節進行系統實作；使用的工具包含：Protégé、Microsoft SQL Server 2000、Tomcat、JSDK、WebLogic、JBuilder、Matchmaker Client - Version 1.2。最後，在第伍節，我們總結本文的重點並探討未來的研究方向。

## 貳、文獻探討

在本節，我們將探討本文所使用到的相關理論與技術，包含語意網路服務 (Semantic Web Service) 技術、知識本體 (Ontology)、OWL 網路知識本體語言 (Web Ontology Language)、OWL-S (Web Ontology Language for Service)、OWL-UDDI 匹配器 (Matchmaker) 與 OWL-S 虛擬機器 (Virtual Machine)。

### 一、語意網路服務

近幾年來，網路上不斷出現各式各樣的服務，如線上銷售、線上拍賣、部落格(blog)等等。然而，網際網路使用者在利用這些服務時，根據本身所需，仍然要花費大量的人工來將相關所需求的服務組合起來。為了降低使用者人力的描述了呼叫一個服務必須要有哪些條

花費，語意網 (Semantic Web) 與網路服務 (Web Services) 這兩種技術可以整合起來而形成語意網路服務 (Semantic Web Services)，它允許電腦可以在語意網中透過自動推理，整合查詢。使用者欲完成任務時，只要透過網路發出要求，語意網路服務軟體代理人 (Software or Intelligent Agent) 就會自動利用語意網推理規劃相關所需的各項服務，並利用網路服務標準界面找尋合適的服務，以完成使用者所交付的任務。

在網路服務中加入語意查詢的概念，經由語意描述網路服務，讓電腦能夠自動處理服務的訊息，是目前網路服務發展的方向。Burstein et al. (2005) 認為語意網路服務的發現和互動可分為下列三個階段：

1. **服務發現 (Service Discovery)**：它最主要的目的在於讓軟體代理人 (Software Agent)，或語意配置者可以找到一個真正符合所需要的服務。而利用 OWL-S 來描述這個服務可以達到服務自動發現的能力。
2. **服務整合 (Service Engagement)**：它的目的是整合相關的網路服務，找出每一個網路服務的位址，產生輸入和輸出的參數。而 OWL-S 則提供了足夠的資訊讓代理人可以找到所需要的服務及其參數或條件，並且讓代理人可以較為容易地組合這些服務來完成工作。
3. **服務調用 (Service Enactment)**：該階段最主要的目的是讓電腦程式或代理人可以自動地呼叫服務。OWL-S 則是件或需要有哪些參數，有了這樣描



述，就可以讓代理人自動地帶入參數完成呼叫。

語意網並非完全取代 WSDL(Web Service Description Language, 網路服務描述語言)，其間是互補的關係，透過語意達到查詢服務的完整性，利用 WSDL 來完成訊息的傳遞、操作和鏈結(戚玉樑, 2004)。

## 二、知識本體

知識本體源於哲學，意涵一個符合全體人類一致共同觀點的系統。知識本體到底是什麼，有許多研究者提出許多解釋。Swartout et al. (1997) 認為知識本體應用在人工智慧上是一組概念，用來描述知識領域、建立知識的陳述。Chandrasekaran et al.(1999)認為知識本體在電腦科學是指某一個領域知識(Domain Knowledge)中相關的術語(或詞彙)的集合。Gruber(1993)則認為知識本體是概念化的一個形式的規格說明，其可以說明知識的架構，表現出一個特定領域的知識核心，可以讓不同領域的知識互相分享。Fensel et al.(1999)指出知識本體是指某領域下之共有的認知，目的是為了以明確且正規的定義來表達共同概念化之事物，讓概念具有共通性並且讓定義出的概念必須要能夠清楚地人在人與電腦之間傳達。

藉由學者對於不同領域的知識本體所下的定義，我們可以發現知識本體是對於現存某個現象或領域的抽象模型，利用概念化的方式，透過共享與明確的方式呈現。Guarino(1995)提到知識本體的概念為將真實世界視為許多不同的領域層次式分類結構和少量約束的本

(domain)所組成，在不同的領域中，有著不同的領域知識，而根據知識本體的內涵有三種分類：術語知識本體(Terminological Ontologies)、資訊知識本體(Information Ontologies)、知識塑模本體(Knowledge Modeling Ontologies)。若是以主要用途來分類則有：上層知識本體(Upper Ontology)、領域知識本體(Domain Ontology)與作業知識本體(Task Ontology)、以及應用知識本體(Application Ontology)。

## 三、OWL

OWL(Web Ontology Language)是W3C(Word Wide Web Consortium,全球資訊網協會)所提出的一種知識本體描述語言,OWL的前身是DAML(DARPA Agent Markup Language) + OIL(Ontology Inference Layer),它繼承了DAML+OIL的框架和大多數語法特徵,同時針對不同的應用範圍做了擴充和限制。OWL包含了三個子語言:OWL Full、OWL DL和OWL Lite;其限制由少到多,其表達能力依次下降,但可計算性依次增強(McGuinness and Harmelen, 2004),其內容摘要如下:

1. *OWL Full*: 提供最豐富的表達能力和最大的 RDF (Resource Description Framework)語法自由度,支援 OWL 的全部語法結構,但沒有可計算性保證。OWL 允許本體擴大與定義辭彙的含義,但不能完全支援電腦自動推理。
2. *OWL Lite*: 它提供最小的表達能力和最強的語義約束,適用於只需要體,例如詞典。因為其語義較為簡



- 單, OWL Lite 比較容易被工具支援。
3. *OWL DL* (Description Logic): 它得名於它的邏輯基礎—描述邏輯。OWL DL 處於 OWL Full 和 OWL Lite 之間, 兼顧表達能力和可計算性。

#### 四、OWL-S

OWL-S 是使用 OWL 語言所描述的網路服務的知識本體, 它也是一種機器可理解的標記語言, 用來描述網路服務的屬性和功能。如圖 1 所示, 服務可分為三部份 (Martin et al., 2004): ServiceProfile(服務簡介), ServiceModel(服務模型), 與 ServiceGrounding(服務基礎)。我們分別說明如下:

1. *服務簡介* (Service Profile): 描述一個服務主要包含三方面的訊息: (1) 服務提供者的資料、聯絡方式等; (2) 服務的功能; (3) 服務本身所屬的

分類。

2. *服務模型* (Service Model): 主要是服務提供者用來描述服務的內部流程, 包含 Atomic Process、Composite Process、Simple Process。Atomic Process 是不可再分的過程, 可以直接被調用。Composite Process 是由幾個 Atomic Process 和 Composite Process 構成的流程。Simple Process 是一個抽象概念, 它不能被直接調用。
3. *服務基礎* (Service Grounding): 描述服務如何被鏈結以及協定、格式等。由於 OWL-S 並沒有定義語法描述內容, 因此它利用 WSDL 規範加以描述。

#### 五、OWL-S/UDDI 匹配器

由於 OWL-S 並沒有辦法完整描述服務的執行描述、介面等。因此它必須利用 OWL-S/UDDI 匹配器將 OWL-S 與

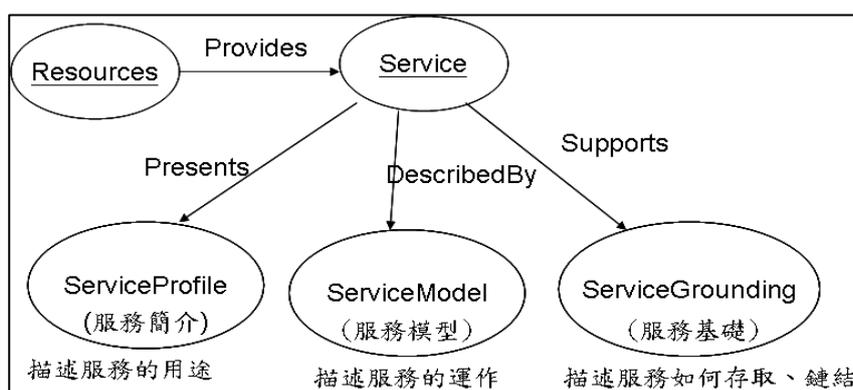


圖 1 OWL-S 整體架構

(資料來源: Martin et al., 2004)



UDDI 中的 *tModel*(技術模型)進行結合，才能完整的描述所發佈的服務。其中 *tModel* 是在 UDDI 登記中表示為服務類型的資料結構。每一個企業在 UDDI 將他們的 Web 服務作登記，並依照已經定義好的一個服務類型進行列表分類。企業可依搜索登記的列表來尋找服務提供者。*tModel* 是一個抽象的服務類型技術規範；它把服務類型的資訊組織起來並做成一個可訪問的登記資料庫 (Brittenham et al., 2001)。圖 2 是一個 OWL-S/UDDI 匹配器，其中包含了通訊模組、OWL-S/UDDI 轉換器、OWL-S 匹配引擎(Paolucci and Sycara, 2003; Sycara et al., 2003; Sycara et al., 2007)。

所有外部的請求都由通信模組進入 OWL-S/UDDI 匹配器中。其中包括服務提供者所發出的服務廣告，服務請求者的查詢和請求。通信模組將廣告 (Advertisements)和查詢(Inquiries)傳遞給 OWL-S/UDDI 轉換器，把用 OWL-S 描述的廣告和查詢轉換成 UDDI 格式，再傳給外部的 UDDI 註冊中心。另一方面，通信模組將請求 (Requests) 傳遞給 OWL-S 匹配引擎，而 OWL-S 匹配引擎利用知識本體匹配請求和服務的功能，找到最合適的服務。圖 3 是 OWL-S 匹配引擎的內部結構，它可分為四個主要元件，分別為廣告資料庫、知識本體資料庫、匹配引擎、推理機。廣告資料庫用

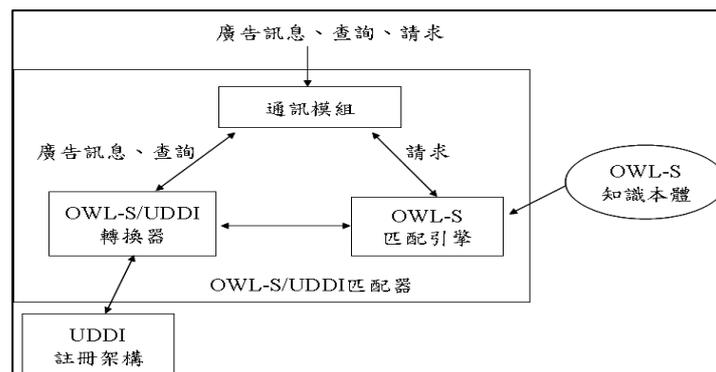


圖 2 OWL-S/UDDI 匹配器

(資料來源：修改自 Paolucci and Sycara, 2003; Sycara et al., 2003; 許慶堂，2006)

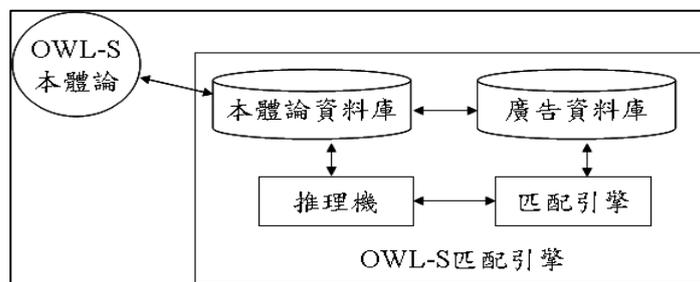


圖 3 OWL-S 匹配引擎

(資料來源：修改自 Paolucci and Sycara, 2003; Sycara et al., 2003; 許慶堂，2006)



來儲存服務提供者發佈的廣告訊息；知識本體資料庫用來儲存服務的知識本體；匹配引擎用來作為服務和需求之間的配對；OWL-S 的推理機，做為知識本體之間的推理。

服務提供者想發佈服務到註冊中心時，先以 OWL-S 描述服務的功能和一些相關的訊息，這些訊息傳到 OWL-S/UDDI 匹配器之後，經由通訊模組傳遞 OWL-S/UDDI 轉換器，再轉換成能夠發佈於 UDDI 上的格式進行發佈，而原本的廣告訊息會被送到廣告資料庫中儲存。當需求者想查詢服務時，透過 OWL-S 匹配引擎將需求傳遞給推理機，推理機利用知識本體資料庫的知識本體，與需求者提出的需求進行推理配

對。例如一個溫泉旅遊的服務，有一消費者想去台東知本的請求。如果只能進行關鍵字的查找，顯然這個服務和請求無法匹配。如果有了關於溫泉旅遊景點的知識本體，其中有一個知識說明知本是一處可以泡溫泉的地方，那麼我們就可以通過推理匹配這個服務和請求。

當 OWL-S/UDDI 匹配器從 UDDI 搜尋完服務之後，必需對所找尋到的服務加以整合(Kwon, 2003)。圖 4 為多個網路整合的關係，其中 Var 代表不同的參數 Var1 到 Var4 這四個參數代表網路服務的輸入參數。Var5 到 Var8 這四個參數則代表網路服務的輸出參數。虛線圍成的範圍是每一個網路服務所包含到的範圍，其中範圍之間會有重覆的存在。當一個

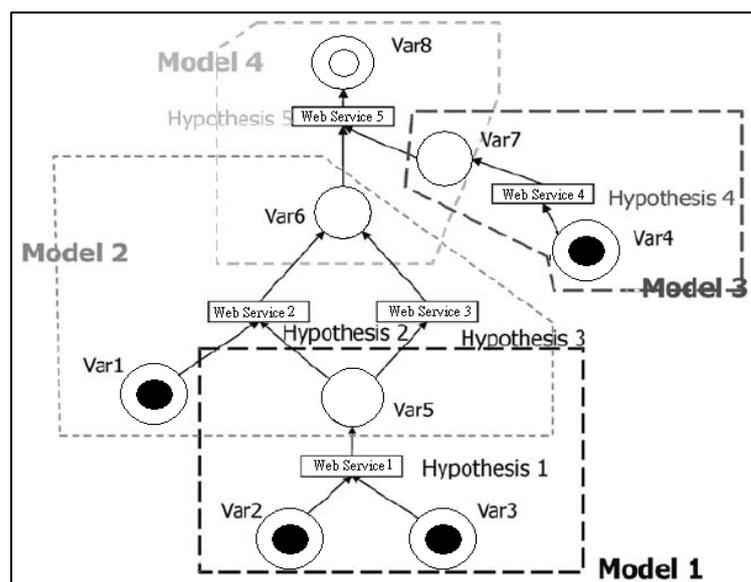


圖 4 多網路服務整合關係  
(資料來源：修改自 Kwon, 2003)



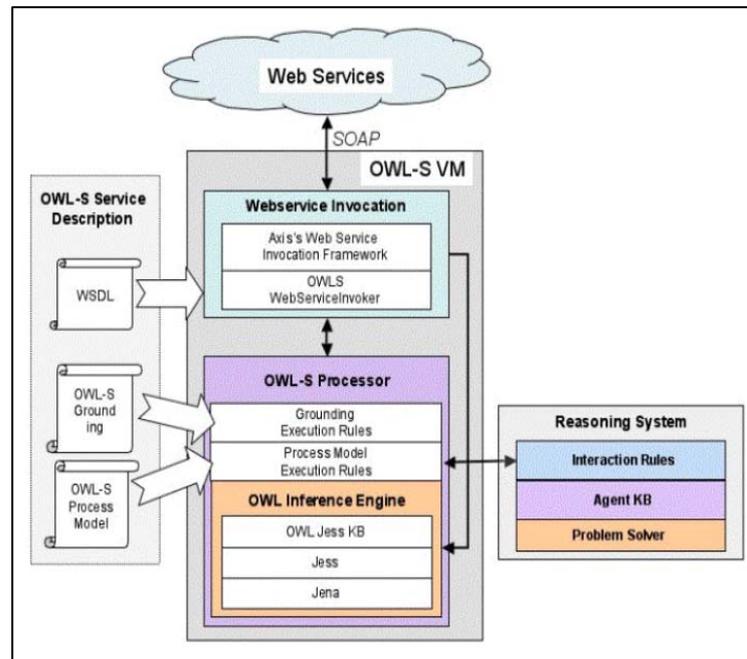


圖 5 OWL-S 虛擬機器

(資料來源：修改自 Sycara et al., 2004)

網路服務的輸出和其他網路服務的輸入相同時，我們就可以利用此參數將網路服務連結在一起，亦即網路服務輸出的參數可視為某網路服務的輸入參數，其中 Var5 到 Var7 這三個參數就是代表此類型的參數。服務與服務之間的連結，使用者並不知道其中有牽涉到哪些服務的整合，使用者只了解其本身只是利用一個網路服務來得到結果，如此可以節省許多人力、時間的支出。

## 六、OWL-S 虛擬機器

圖 5 為 OWL-S VM (Virtual Machine, 虛擬機器) 提供了一個完整的基於 OWL-S 的網路服務的調用環境。它包含下列三個主要部份(Sycara et al., 2004)：

1. *Web Service Invocation* (網路服務呼叫)：包含兩個主要部份，分別是 Axis's Service Invocation、OWL-S Web Service Invoker；主要是將服務轉換成 OWL-S 格式並傳給 OWL 推論引擎。
2. *OWL-S Processor* (處理器)：包含 Grounding Execution Rules、Process Model Execution Rules；主要是藉由網路服務呼叫與服務提供者溝通，來決定服務如何被執行。
3. *OWL Inference Engine* (推論引擎)：包含 OWL Jess KB、Jess、Jena 三個推理器，主要處理從其他的服務取得的 OWL-S 相關規定，並將訊息傳送給 OWL-S 處理器。



## 參、系統分析與設計

本文的核心在於如何有效利用網路服務整合行程資訊與透過語意網的技術提高搜尋效率以及建構旅遊行程知識本體，進而讓領域知識更加容易分享與使用。在本節，我們描述旅遊行程語意網路服務系統(ISWSS)有關的需求分析，接著我們呈現旅遊行程知識本體。之後，我們進一步介紹 ISWSS 完整的系統架構，並分別以使用者及服務提供者的角度來說明 ISWSS 系統的運作流程。

### 一、需求分析

本文所建置的 ISWSS 系統主要是以欲參加旅遊業者所安排旅遊行程的消費者為研究對象，因此我們必須先了解消費者選擇行程時所考量的需求，以便於分析與了解系統的功能與建置的目標。一般大眾出團旅遊、度假都會先行查詢旅遊業者所安排的行程，通常業者安排的行程資訊不外乎以下幾類：

- *旅遊行程資訊*：消費者在出外旅遊前，一定會先查詢旅行社提供哪些行程，這些行程通常包括地點、天數、價錢、住宿、航空公司、出發地點等。以往網路尚未普及時，消費者必須到旅遊業者所開的實體辦事處才能查詢到資訊，但隨著網路的發達，依消費者的習慣，大多數的人都會先到各家業者的網站進行查詢。然而，消費者在制定決策的過程中往往出現一個盲點，亦即所查詢到的資訊僅僅只限於此一家，

不然就是利用關鍵字在搜尋引擎上查詢 如此將會造成所查詢到的資料過於龐大且不符合消費者所需。為了解決此一問題，經由我們所開發的 ISWSS 系統查詢後，可自動的找尋最相關的訊息提供給消費者，以減少查詢的時間與增加查詢效率。此類的服務，我們可由業者公司的網站或書面資料取得。

- *旅遊資訊*：消費者選定行程之後，通常會查詢當地區域日常生活用品的規格以及當地的氣候、氣溫等，以了解當地一些生活訊息，而 ISWSS 系統亦提供此資訊給消費者。此類資訊，我們可由氣象網站或旅遊網站上得到，提供給消費者做為參考。
- *購買途徑*：購買方式可分為實體店家購買與網路購買。在實體店家購買上，提供其業者公司與分公司地址給消費者自行到其店家購買；而網路購買方面則是聯結到其公司的網址，由消費者點選訂購。

### 二、旅遊行程知識本體

透過知識本體的建置，能夠讓領域彼此之間的知識更容易分享。本文蒐集台灣各個旅遊業者所建置的網站資料，並參考由王治立(2004)所提出的城市旅遊知識本體(其範圍較小，不包含旅遊業者，而交通工具亦不涵蓋飛機)；我們針對旅遊業者與行程的內容特別加以擴充



與修改並建構出新的旅遊行程知識本體。另外，我們利用 UML 的 Aggregation

關係表示法呈現出旅遊行程知識本體，如圖 6 所示。

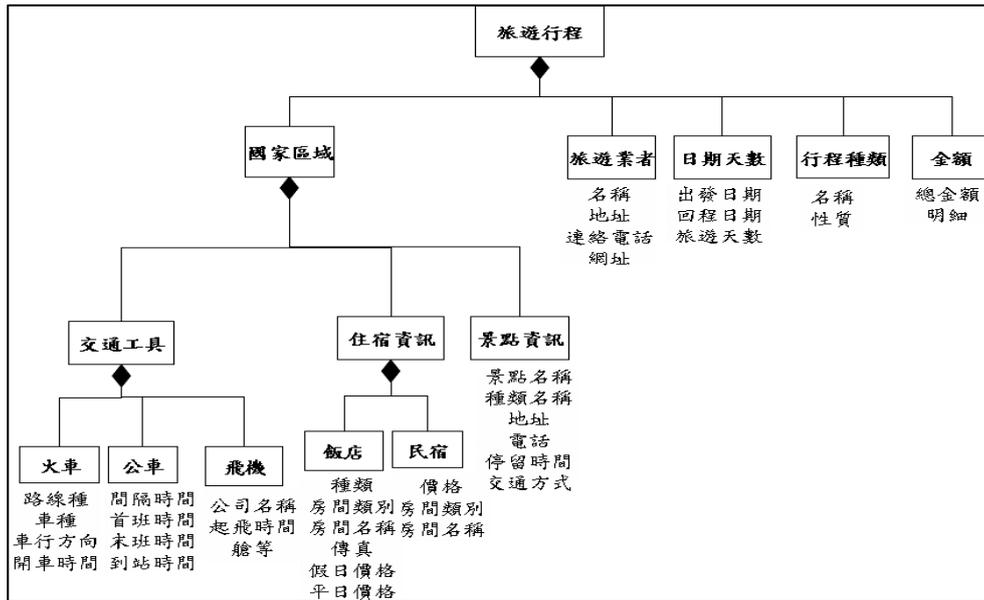


圖 6 旅遊行程知識本體

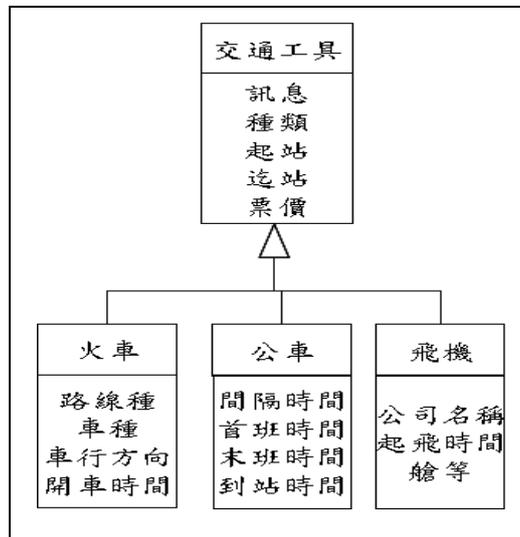


圖 7 交通工具知識本體



我們改良王治立(2004)所提出的交通工具知識本體成爲新的交通工具知識本體(包含飛機)，並以 UML 的 Generalization 關係表示法呈現交通工具知識本體，如圖 7 所示。「交通工具」的屬性包含訊息、種類、起站、迄站與票價。「火車」的屬性包含路線種、車種、車行方向、開車時間。「公車」的屬性包含間隔時間、首班時間、末班時間及到站時間。「飛機」的屬性則包括航空公司、起飛日期時間、艙等。而「交通工具」爲「火車」、「公車」、「飛機」的父類別，因此經由繼承關係，「火車」、「公車」、「飛機」將擁有父類別「交通工具」的所有屬性。

### 三、ISWSS 系統架構

圖 8 爲 ISWSS 之完整系統架構，本系統包含下列七個主要元件：

1. 旅遊行程服務提供者：即提供給行

程服務提供者發佈旅遊行程服務的網站。行程服務提供者建立好服務描述文件後，透過 OWL-S/UDDI 匹配器將廣告訊息轉換成 WSDL 格式後，並向提供註冊服務的註冊中心(UDDI)網站發佈，以方便服務要求者的搜尋、取得與使用。

2. 使用者：即想要使用服務提供者在註冊中心所發佈服務的個人或企業。使用者透過 OWL-S/UDDI 匹配器可以利用儲存於匹配器中的知識本體和使用者的需求做比對，找出符合使用者需求的服務。
3. 註冊中心(UDDI)：儲存由 OWL-S/UDDI 匹配器轉換服務提供者提供的廣告訊息，提供一個發佈訊息的平台以供查詢。
4. OWL-S/UDDI 匹配器：接收來自服務提供者提供的旅遊行程資訊，將其轉換成 WSDL 格式後發佈於

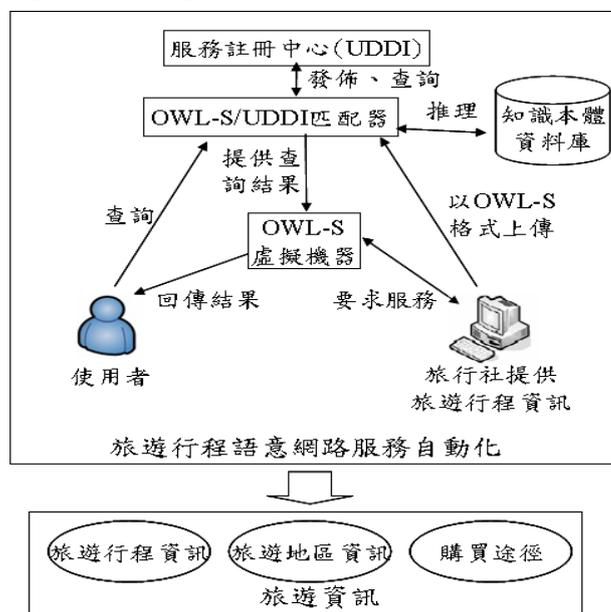


圖 8 旅遊行程語意網路服務系統



UDDI 中，並提供使用者查詢資訊的服務。依照使用者的查詢訊息，藉由匹配引擎的推理、匹配找出符合使用者所查詢的訊息，並將訊息傳遞給 OWL-S 虛擬機作為自動調用的依據。

5. **OWL-S 虛擬機器**：藉由 OWL-S/UDDI 匹配器所提供的參數訊息，自動找尋網際網路上相關的訊息，加以整合並提供給使用者最合適的服務。
6. **旅遊資訊**：透過 ISWSS 自動化網路服務的運作之後，使用者可以取得最為合適的資料，其中包括旅遊行程資訊、旅遊地區資訊以及購買途徑三項訊息。

7. **知識本體資料庫**：用來儲存知識本體的資料庫，可供 OWL-S/UDDI 匹配器進行配對與推理。

#### 四、使用者查詢服務流程

使用者透過 ISWSS 系統的服務查詢流程，可以查詢出合適的旅遊行程。如圖 9 所示，查詢服務介面是使用者使用本系統的媒介，使用者透過服務查詢介面，輸入欲查詢旅遊行程的相關字詞以進行查詢，可得出使用者所需的訊息。服務查詢介面在設計上，將行程可用來查詢的屬性都列出來，如行程的性質、行程的地點區域、行程的金額等。使用

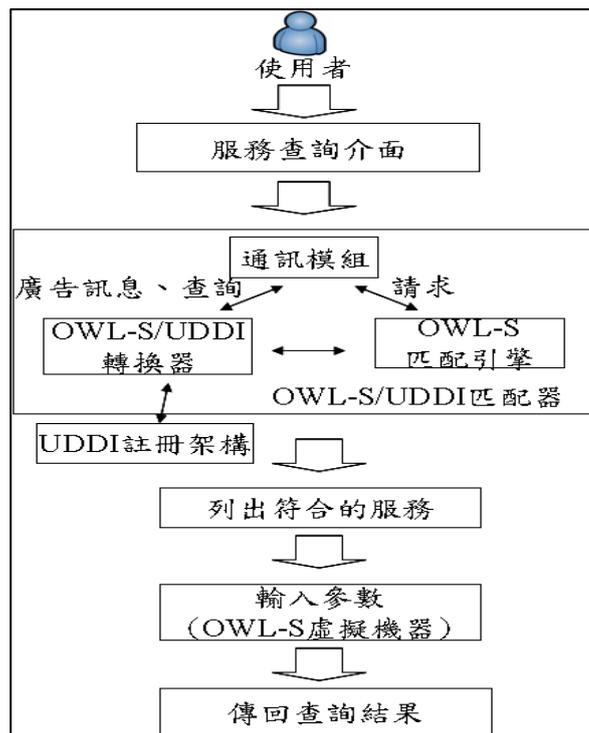


圖 9 使用者查詢流程圖



者可以選擇輸入哪些屬性的關鍵詞來查詢服務，而 OWL-S/UDDI 轉換器便會利用這些使用者輸入的關鍵詞，轉換成查詢的知識本體，最後將知識本體傳送到下一個部份。

OWL-S/UDDI 匹配器中的通訊模組接收到服務查詢介面所傳來的查詢知識本體後，便會將之傳到 OWL-S 匹配引擎進行配對；而匹配引擎則利用廣告資料庫和知識本體資料庫裡的資料做為配對的資訊，最後將符合的服務傳送到下一部份。當 OWL-S 虛擬機器接收到由匹配器傳送過來的旅遊行程參數後，系統會透過網路服務呼叫將其服務轉換成 WSDL 格式並傳送給 OWL 推論引擎。爾後 OWL 推論引擎會處理從其他服務所取得的 OWL-S 相關規定，並將訊息傳給 OWL-S 處理器；而 OWL-S 處理器則藉由網路服務呼叫與旅遊業者取得聯繫，來決定服務如何被執行。

## 五、服務提供者註冊流程

旅遊行程資訊係由旅行社提供，因此亦必須有一介面提供給旅行社業者上傳的訊息，其流程如圖 10 所示。業者藉由上傳介面將所要發佈的旅遊行程訊息輸入，系統會以 OWL-S 格式將旅遊行程訊息上傳至 OWL-S/UDDI 匹配器，再由匹配器內部的 OWL-S/UDDI 轉換器將 OWL-S 檔案轉成 WSDL 格式並發佈於 UDDI 註冊中心。發佈的廣告訊息必須包含旅行社的名稱、地址、行程訊息、連結等詳細資料以方便日後查詢與調用。隨著 UDDI 註冊中心資料量的增加，系統所能查詢到的資料便更加齊全，亦能滿足更多符合使用者需求條件以便提供使用者選擇。

我們將 ISWSS 系統完整的處理流程呈現於圖 11。在此圖，當箭頭行經到一個菱形決策符號時，就形成了轉換。

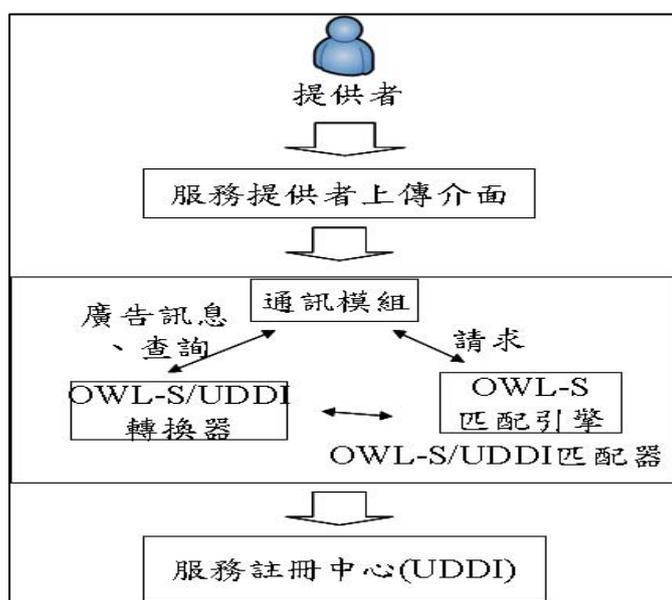


圖 10 服務提供者上傳流程圖



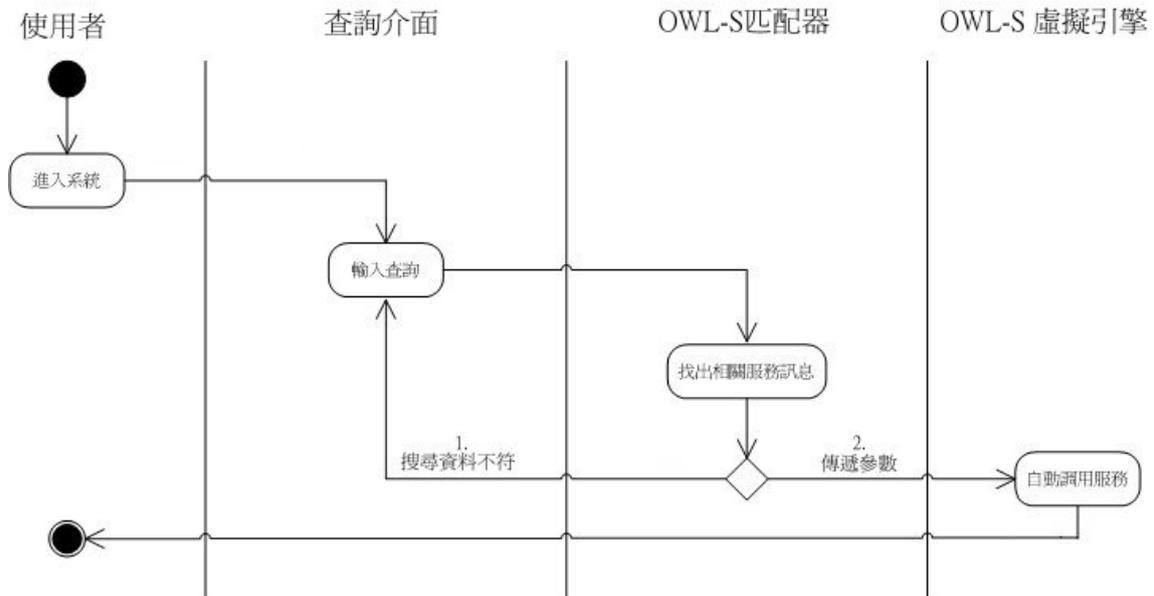


圖 11 系統處理流程活動圖

當系統發生轉換時，此時會有多個路徑可供選擇，一般只可以選擇其中一條路徑來繼續之後的作業步驟。這個地方會選擇何種路徑的判斷依據，我們稱為成立條件；所謂成立條件就是當事情發生時所要做出的決定。舉例來說，一個活動箭頭指向菱形決策符號時，若是 1 則回到原有的活動，若是 2 則是繼續之後的活動，因此菱形決策符號即是判斷式，用來決定進入 1 或是 2 的路徑。因此，從圖 11 中我們可以清楚得知，當使用者進入系統後，輸入查詢訊息之後就開始了整個搜尋運作。

藉由使用者介面來和 OWL-S 匹配器溝通，當系統從 OWL-S 匹配器中找出服務後，輸出參數到 OWL-S 虛擬機器，此時系統會啓用 OWL-S 虛擬機器選擇自 UDDI 所提供的連接方式與供應商連接，而當使用者對於找出的結果不甚滿

意時，也可以透過人為選擇改用其他條件來重新搜尋。此時，就可以重新設定找尋條件，並展開搜尋程序，直到使用者找到適合的服務，最後到整個活動的結束節點(亦即圓圈環繞黑色圓形實心符號所示)。

## 肆、系統實作

在本節，我們首先探討如何建置旅遊行程知識本體，接著談到系統的開發步驟與應用實例。本文建構的知識本體乃利用由史丹佛大學醫學資訊中心所研發的 Protégé 3.2.1 作為知識本體的編輯工具。本系統的資料庫則是採用 Microsoft SQL Server 2000 資料庫來儲存旅遊行程以及廣告訊息的資料。我們以 Tomcat 5.54、JSDK 1.4.1 作為系統運作平台，並利用 WebLogic 作為發佈網路服務



的伺服器。使用 JSP 作為網頁撰寫語言，以 Tomcat 搭配 JSP 作為網頁後端連結的伺服器。透過 JavaBean 的配合運用，讓 JSP 能夠使用於應用服務上。系統介面的撰寫工具則是利用 JBuilder，由於 WebLogic 能夠透過 JBuilder 來發佈服務以及其功能較為完整。OWL/UDDI 匹配器是利用由美國卡內基美隆大學(CMU)所開發的 Matchmaker Client - Version 1.2 來做為 WSDL 與 OWL-S 之間的轉換工具。

### 一、旅遊行程知識本體之建構

知識本體的建立階段可以利用各種知識本體的編輯工具來建立(Noy and McGuinness, 2007)，我們選擇史丹佛大學醫學資訊中心所研發的 Protégé 3.2.1(<http://protege.stanford.edu>) 作為知識本體的編輯工具。因 Protégé 3.2.1 是目前較受歡迎的知識本體編輯工具，它支援 OWL-DL 語法，除了有多種平台的版本之外，還可以外掛的方式載入像是

推論引擎(Racer)、視覺化階層架構圖(OWLvizTab)等延伸功能，能加快知識本體建構的速度與效率。當知識本體建置完成之後，可以利用 Protégé 3.2.1 檢視知識本體內容的一致性(Consistency)、並利用 Protégé 3.2.1 外掛的推論引擎來發掘知識本體中的一些隱藏的概念(Implicit Concepts)，顯示建置後的本體能夠由電腦進行推論的價值(蘇嶸學，2006)，並將知識本體儲存於知識本體資料庫(Ontology Database)中以供查詢。

在以 Protégé 初步建置完成後(如圖 12)，我們可以透過 Racer Pro 來驗證整體建置過程中是否有邏輯錯誤存在。Racer System 公司的 Racer Pro 軟體為德國漢堡大學所開發，它可以與 Protégé 結合做為驗證知識本體一致性及簡單推論的工具。而透過 Protégé 呼叫 Racer Pro 驗證的畫面如圖 13 所示。在 Connected to Racer 的對話視窗中，可清楚看到各項類別關係與邏輯的驗證檢查結果。圖 14 為利用 Protégé 的外掛程式 OWLVizTab 以

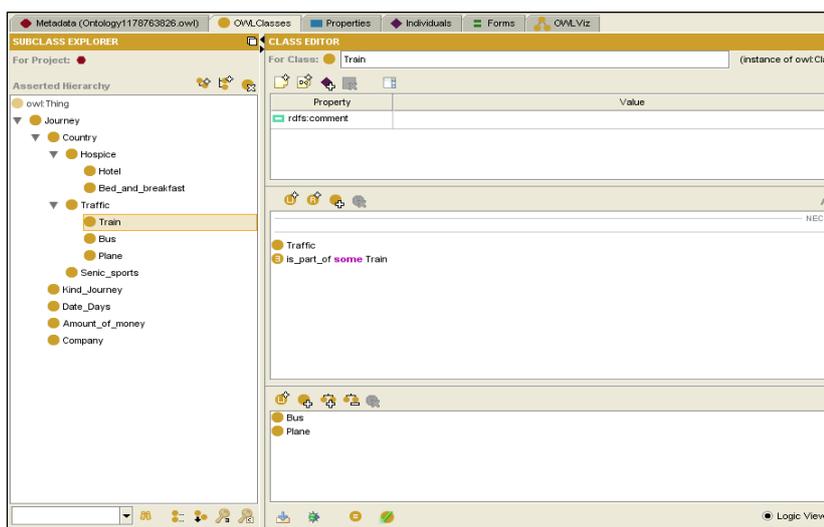


圖 12 利用 Protégé 建構旅遊行程知識本體



圖形化的方式呈現旅遊行程知識本體。本文對於知識本體的描述則採用 OWL 語言。因此，利用 Protégé 的圖形化介面

建構出知識本體的架構後，再轉換以 OWL 語法格式來呈現，如圖 15 所示。

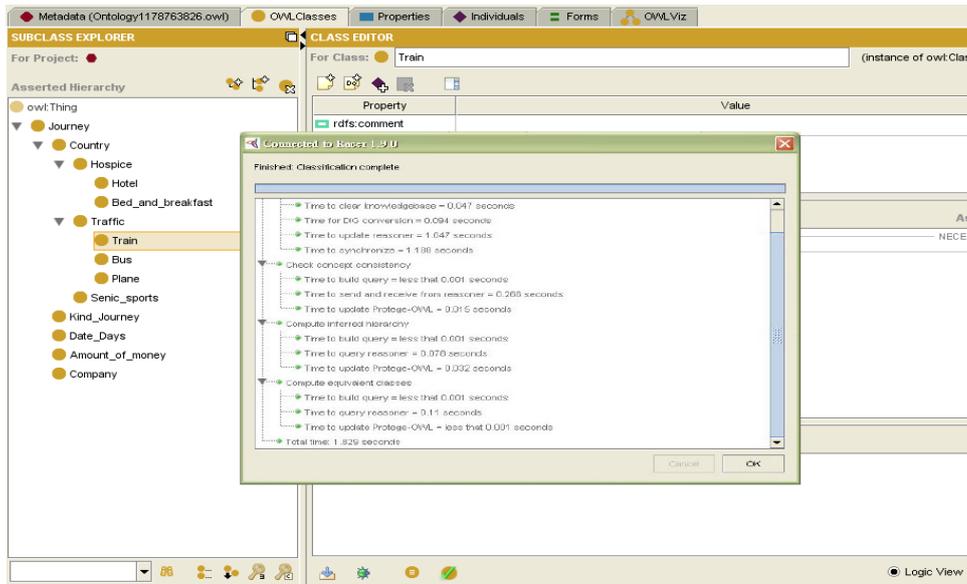


圖 13 以 Racer Pro 驗證旅遊行程知識本體

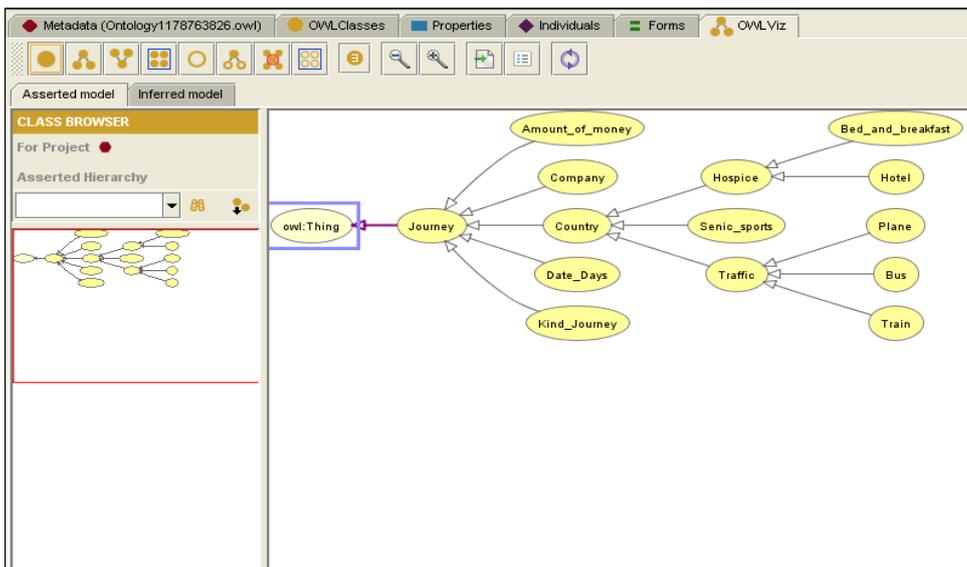


圖 14 利用 OWLVizTab 以圖形化來呈現旅遊行程知識本體



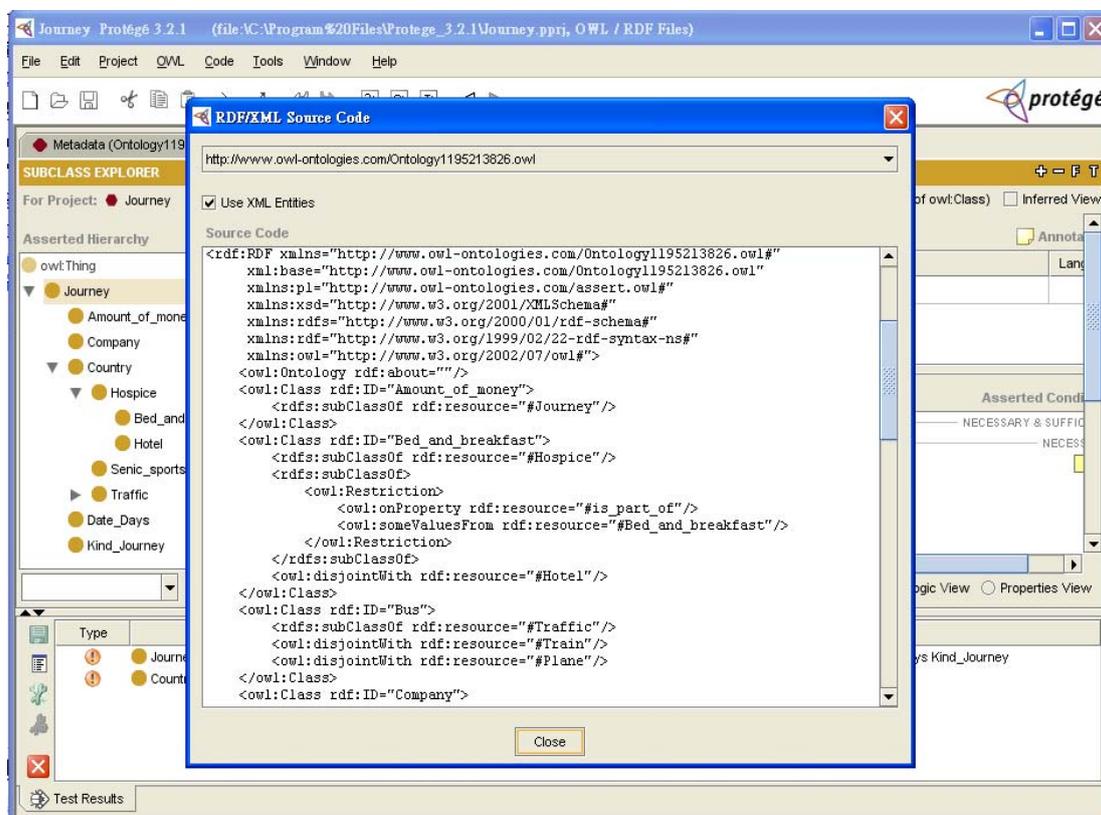


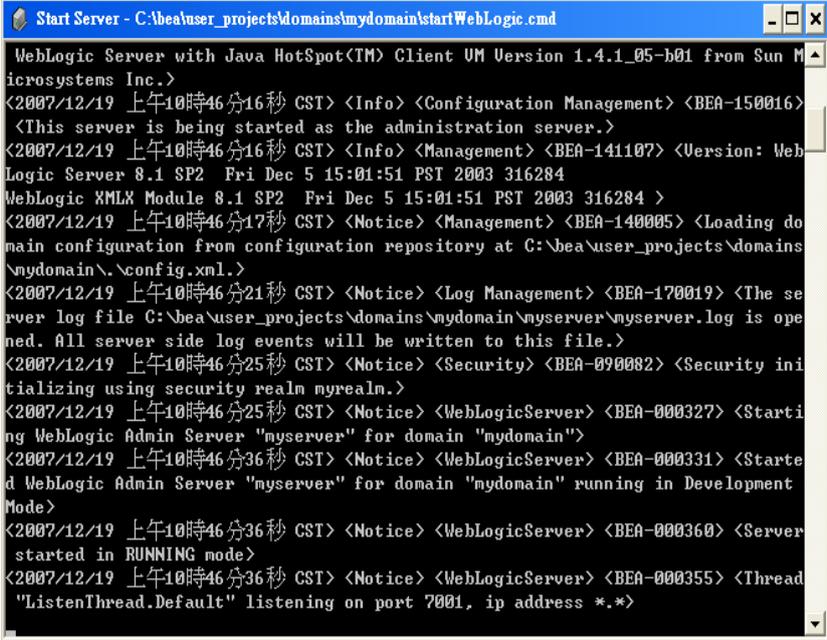
圖 15 以 OWL 語法描述旅遊行程知識本體

## 二、服務發佈與資料格式轉換

我們利用 Tomcat 5.5.4 做為網頁連結伺服器結合後端 SQL Server 2000 資料庫用來儲存知識本體和廣告訊息，來進行配對之用。關於網路服務的發佈，本系統利用 BEA WebLogic 來做為服務發佈的網路服務伺服器，如圖 16 所示。BEA WebLogic 是一個功能豐富、基於標準的應用伺服器(Application Server)，是由 BEA 公司所開發，目前最新版本為 WebLogic 9，它為企業建構可靠、可伸縮和可管理的應用程式提供了一個堅實的

基礎。借助於全面的功能、對開放標準的服從、多層的架構和對基於元件開發的支援，WebLogic 為開發和部署業務驅動的應用程式提供了必需的底層核心功能(王文傑等, 2003)。圖 17 是利用 Tomcat Administration 來連結 SQL Server2000 內的 juddi 資料庫。juddi 資料庫主要用來儲存服務提供者所發佈的服務，以及提供給使用者透過匹配器來進行服務的查詢與比對。





```
Start Server - C:\bea\user_projects\domains\mydomain\startWebLogic.cmd
WebLogic Server with Java HotSpot(TM) Client VM Version 1.4.1_05-b01 from Sun M
icrosystems Inc.)
<2007/12/19 上午10時46分16秒 CST> <Info> <Configuration Management> <BEA-150016>
<This server is being started as the administration server.>
<2007/12/19 上午10時46分16秒 CST> <Info> <Management> <BEA-141107> <Version: Web
Logic Server 8.1 SP2 Fri Dec 5 15:01:51 PST 2003 316284
WebLogic XMLK Module 8.1 SP2 Fri Dec 5 15:01:51 PST 2003 316284 >
<2007/12/19 上午10時46分17秒 CST> <Notice> <Management> <BEA-140005> <Loading do
main configuration from configuration repository at C:\bea\user_projects\domains
\mydomain\.\config.xml.>
<2007/12/19 上午10時46分21秒 CST> <Notice> <Log Management> <BEA-170019> <The se
rver log file C:\bea\user_projects\domains\mydomain\myserver\myserver.log is ope
ned. All server side log events will be written to this file.>
<2007/12/19 上午10時46分25秒 CST> <Notice> <Security> <BEA-090082> <Security ini
tializing using security realm myrealm.>
<2007/12/19 上午10時46分25秒 CST> <Notice> <WebLogicServer> <BEA-000327> <Starti
ng WebLogic Admin Server "myserver" for domain "mydomain">
<2007/12/19 上午10時46分36秒 CST> <Notice> <WebLogicServer> <BEA-000331> <Starte
d WebLogic Admin Server "myserver" for domain "mydomain" running in Development
Mode>
<2007/12/19 上午10時46分36秒 CST> <Notice> <WebLogicServer> <BEA-000360> <Server
started in RUNNING mode>
<2007/12/19 上午10時46分36秒 CST> <Notice> <WebLogicServer> <BEA-000355> <Thread
"ListenThread.Default" listening on port 7001, ip address *.*>
```

圖 16 啟動 WebLogic 來發佈服務

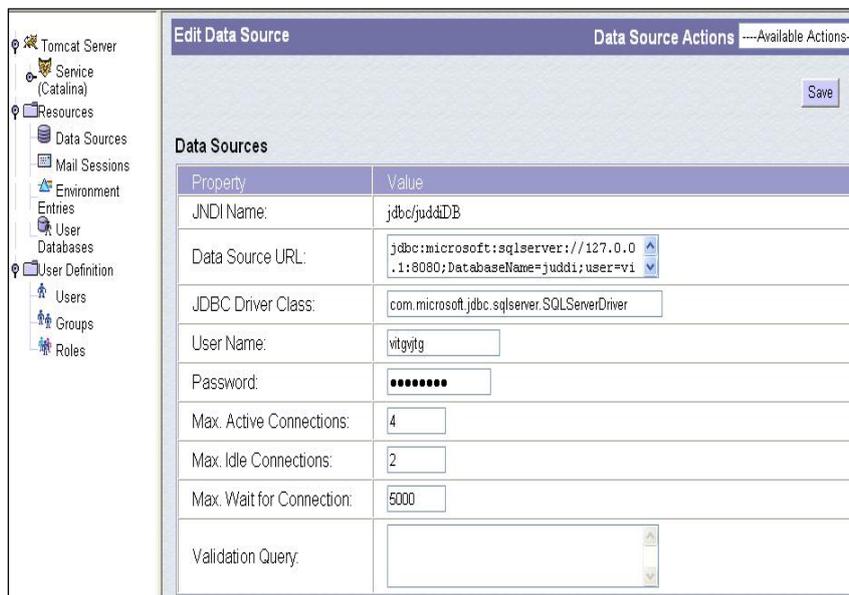


圖 17 利用 Tomcat Administration 連結 juddi 資料庫



The screenshot shows a web browser window titled "MyIE2 - [旅遊行程語意網路服務系統]". The address bar shows "http://127.0.0.1:83.jsp". The page content is a form titled "旅遊行程語意網路服務系統" with the subtitle "行程提供者輸入介面". The form fields are as follows:

服務名稱	日本北海道甜蜜度假之旅
服務描述	造訪日本唯一僅存古色古香，並富有歐洲風情之【小樽市】，在此您可漫步於羅曼蒂克的古老街道，隨後可參觀玻璃硝子場觀賞各式各樣的彩繪玻璃。
公司名稱	鳳凰國際旅行社股份有限公司
公司電話、傳真、E-mail	電話: 886-2-2537-8111 傳真: 886-2-2536-1790 E-mail: info@phoenix.com.tw
公司地址	台北市長安東路一段25號4、5樓
公司網址	http://www.travel.com.tw/
行程類型	度蜜月
行程天數	5
金額	35000

圖 18 服務提供者發佈服務介面

我們以旅行社業者與一般消費者的觀點，來說明旅遊行程語意網路服務系統(ISWSS)的應用。例如，鳳凰旅行社欲將日本北海道蜜月行程五日遊，金額為單人四萬元以下等資訊的行程訊息發佈於註冊中心。旅行社業者透過註冊介面(如圖 18)，輸入行程相關訊息。輸入完成後，系統將旅遊行程資訊加以轉換成 OWL-S 格式傳給本系統的 OWL-S 匹配器，再藉由匹配器轉換成 WSDL 格式，並將服務發佈於註冊中心以供查詢。提供服務的企業或廠商在進行註冊的過程時，需提供描述服務資訊的名稱及 WSDL 文件，並儲存於註冊中心之 WSDL 文件資料庫內。因此服務需求者可依據資訊檢索模組所得的查詢結果，取得所需要服務的 WSDL 文件，並得以進一步使用該服務。

旅遊行程提供者可以透過網際網路，利用瀏覽器新增廣告訊息。例如，圖 19 便是利用 Matchmaker Client - Version 1.2 並透過 WEB 介面新增一則廣告訊息所呈現的 OWL 格式檔案，並將 OWL-S 格式的廣告訊息傳送到 OWL-S/UDDI 匹配器(Sycara et al., 2007)。

當服務提供者將廣告訊息上傳至 OWL-S/UDDI 匹配器之後，透過通訊模組將服務傳送至 OWL-S/UDDI 轉換器，藉由轉換器將 OWL 格式轉換成 WSDL 格式，然後發佈於 UDDI 註冊中心。由於 OWL-S 規範中並沒有定義語法成分來描述具體的消息，而是利用 WSDL 規範。圖 20 為透過 OWL-S/UDDI 匹配器來將 OWL 格式與 WSDL 之間進行概念的結合。



```

<owl:imports rdf:resource="C:\Documents and Settings\Administrator\OWL ontologies"/>
</owl:Ontology>

<profile:Profile>
  <profile:serviceName>日本北海道甜蜜度假之旅</profile:serviceName>
  <profile:textDescription>
    造訪日本唯一僅存古色古香，並富有歐洲風情之【小樽市】，
    在此您可漫步於羅曼蒂克的古老街道，隨後可參觀玻璃硝子場
    觀賞各式各樣的彩繪玻璃。
  </profile:textDescription>
</profile:Profile>

<profile:contactInformation>
  <actor:Actor rdf:ID="鳳凰國際旅行社股份有限公司">
    <actor:name>鳳凰國際旅行社股份有限公司</actor:name>
    <actor:title>日本北海道甜蜜度假之旅</actor:title>
    <actor:phone>886-2-2537-8111</actor:phone>
    <actor:fax>info@phoenix.com.tw</actor:fax>
    <actor:email>886-2-2536-1790</actor:email>
    <actor:physicalAddress>10441 台北市長安東路一段25號4、5樓</actor:physicalAddress>
    <actor:webURL>http://www.travel.com.tw</actor:webURL>
  </actor:Actor>
</profile:contactInformation>
</profile:Profile>
<profile:serviceCategory>
  <addParam:ServiceCategory rdf:ID="度蜜月">
    <addParam:taxonomy>http://www.travel.com.tw/</addParam:taxonomy>
    <addParam:value>provider</addParam:value>
    <addParam:code>1022</addParam:code>
  </addParam:ServiceCategory>
</profile:serviceCategory>
<!-- Descriptions of the parameters that will be used by IOPEs -->
</profile:Profile>
</rdf:RDF>

```

圖 19 服務提供者透過 WEB 介面來發佈旅遊廣告訊息

```

<contacts>
  <contact useType='日本北海道甜蜜度假之旅'>
    <personName>鳳凰國際旅行社股份有限公司</personName>
    <phone>886-2-2537-8111</phone>
    <email>info@phoenix.com.tw</email>
    <address>
      <addressLine> 10441 台北市長安東路一段25號4、5樓</addressLine>
    </address>
  </contact>
</contacts>

<businessServices>
  <businessService serviceKey=''>
    <name>日本北海道甜蜜度假之旅</name>
    <description>造訪日本唯一僅存古色古香，並富有歐洲風情之【小樽市】，
    在此您可漫步於羅曼蒂克的古老街道，隨後可參觀玻璃硝子場
    觀賞各式各樣的彩繪玻璃。</description>
    <categoryBag>
      <keyedReference keyValue='AnonProfile1195223229406' tModelKey='UUID:74194360-DA9E-41D7-A0CA-020629DC0A33' />
      <keyedReference keyValue='http://travel.com.tw/trip/trip_list.php?gdate=JPN070109A' tModelKey='UUID:9B609CA0-D' />
      <keyedReference keyValue='C:\Documents and Settings\Administrator\OWL ontologies' tModelKey='UUID:5384C24R-Q48' />
      <keyedReference keyValue='provider' keyName='1022' tModelKey='UUID:4614C240-B483-11D7-8BE8-000629DC0A53' />
    </categoryBag>
  </businessService>
</businessServices>
</businessEntity>

```

圖 20 廣告訊息中的 OWL-S 與 WSDL 結合



### 三、以 OWL-S/UDDI 匹配器達成資料的搜尋

圖 21 為使用者查詢服務的介面，使用者可透過該查詢介面，輸入欲查詢的資料，系統藉由 OWL-S/UDDI 匹配器中的匹配引擎從 UDDI 註冊中心找出合適的服務以及相關的基本資料、WSDL 呼叫路徑。例如，林先生與林太太想上網查詢日本北海道蜜月行程五日遊，但是預算有限，經費設定在五萬元以下。以過去的經驗，林先生可能採用的搜尋方式有很多種，如在 YAHOO 或 GOOGLE 搜尋引擎上鍵入「旅行社」或「北海道蜜月旅行」關鍵字搜尋相關網頁資訊，結果出現了超過十幾個頁面以上的網頁，此現象乃因為國內旅行社數量過多，所提供的資料亦過於繁雜，讓關鍵字搜尋的效率不能達到使用者的需求。如此不但無法找出真正使用者的需求，且容易花費過多的時間在找尋、比對資訊。然而，透過我們所建置的 ISWSS 系統的使用，林先生於查詢介面上可填入國家為日本，行程種類為蜜月旅行，經

費在四萬到五萬元之間等條件，再經由匹配器從 UDDI 註冊機制找尋相關廣告訊息，透過匹配引擎配對後，找到鳳凰旅行社所提供的服務最能符合林先生的要求，並將參數傳遞給虛擬機器，表 1 為傳統查詢與 ISWSS 語意查詢所得到的檢索結果比較。

在圖 21 的使用者查詢介面中，資訊檢索畫面的內容是按照旅遊行程知識本體的樹狀結構來設計，使用者可以按照系統設定好的資料選項，在黑色方格中由樹狀架構的第一層依序到第三層選擇所需要查詢的關鍵字詞，以作為輸入。於第一層查詢中，使用者先填入第一層查詢所需填選的項目，並在國家城市中選擇第二層展開後，可以選擇三種查詢，分別為交通工具、住宿資訊以及景點資訊，如圖 22 所示。選擇其中一項之後，第三層樹狀結構展開(如圖 23 所示)，便可選擇於第二層結構所選擇層級的屬性，如此方可查詢到更符合使用者需求的條件。

圖 21 服務查詢介面



表1 傳統查詢與ISWSS語意查詢檢索結果比較

資訊檢索工具	檢索結果內容
Google與YAHOO搜尋引擎 (傳統查詢)	只要符合「旅行社」或「北海道蜜月旅行」的關鍵字詞網頁內容，都包含在檢索結果。
旅遊行程語意網路服務系統 (ISWSS語意查詢)	系統針對「北海道蜜月旅行」這組關鍵字詞與知識本體比對後所做的檢索，檢索結果呈現使用者真正關心的旅遊資訊。

圖 22 第二層展開頁面

圖 23 第三層展開頁面



#### 四、服務的整合及服務自動調用

使用者將欲查詢的資料透過服務查詢介面輸入之後，透過 OWL-S/UDDI 匹配器中的匹配引擎來進行搜尋合適的資料，接著整合相關的網路服務，產生輸入與輸出的參數。OWL-S 虛擬機器藉由所接收到的訊息，找尋出服務提供者所提供的服務出來加以整合，並呈獻給使用者完整的服務訊息以及相關的連結。

由於現今 OWL-S 虛擬機器所擁有的應用軟體與相關資料並不多，大多處於理論階段。雖然目前美國卡內基美隆大學對於 OWL-S 虛擬機器有較多的研

究探討，但其真正應用於實際系統上的相關資料卻較缺乏，因此我們運用記錄於廣告資料庫中的廣告相關訊息，找出每一個網路服務的 URI 的方式來達到 OWL-S 虛擬機器的功能與作用，完成服務的連結與調用。

由前面所舉的實例，系統會顯示鳳凰旅行社以及多家符合林先生所查詢的行程訊息給林先生作為參考，並顯示旅行社的聯絡方式、連結網址等資訊，如圖 24 所示。當中的匹配程度計算，是利用我們所提出的一套匹配度計算公式(如表 2 所示)，來進行相似度的計算。當使

旅行社	旅遊行程名稱	出發日	星期	天數	航班	費用	連結	匹配度
鳳凰國際旅行社股份有限公司	日本北海道甜蜜度假之旅	2007/10/18	四	6	CI 中華航空	35,000	<a href="#">連結</a>	70
雄獅旅遊	札幌白色燈節.熱氣球.騎馬.三大螃蟹.美人湯饗樂5日	2007/10/22	一	5	CI 中華航空	24,900	<a href="#">連結</a>	64
雄獅旅遊	醉愛北海道~三大螃蟹吃到飽~溫泉美食饗樂5日【千歲/函館】	2007/10/24	三	5	UNI立榮航空	21,900	<a href="#">連結</a>	61
山富旅遊	北海道【北斗星號列車V.S.函館山百萬夜景】5日【帶/函】	2007/10/25	四	5	CI 中華航空	27,600	<a href="#">連結</a>	55

圖 24 使用者查詢輸出結果

表 2 匹配度計算公式

計算公式	$T = A * 10 + B * 1$
符號說明	A：透過知識本體比對後關鍵字相符合次數。 B：利用關鍵字詞比對後相符合次數。 T：加總次數。



用者輸入關鍵字查詢時，系統會先行透過知識本體資料庫來比對是否有符合的字詞，其必須完全符合才能進行加分動作，每符合一筆便增加 10 分，接著利用傳統關鍵字詞加以比對，符合一字便增加 1 分，以此加總分數提供給使用者參考。

除了提供行程資訊外，使用者連結進服務之後，系統也會提供相對應的旅遊資訊及購買途徑的相關訊息給使用者。一般的行程資訊通常包括地點、天數、價錢、住宿、航空公司、出發地點等。而旅遊資訊則是如當地區域日常生活用品的規格以及當地的氣候、氣溫、簽證、匯率、急難救助等相關訊息，此類訊息我們可由氣象網站或旅遊網站上得到，提供給消費者做為參考。而購買途徑大致上分為實體店家購買與網路購買。實體店家購買上，提供其業者公司與分公司地址給消費者自行到其店家購買，網路購買方面則是聯結到其公司網址，由消費者點選訂購。

ISWSS 系統是從服務提供者提供的訊息中，透過關鍵字詞及知識本體的比對，來對於使用者所輸入的關鍵字進行百分比的評估。所輸入的關鍵字會先從知識本體來判斷是否有完全相符的字詞，如果有完全符合的字詞則分數會提高。

## 伍、結論與未來展望

由於網路資訊的龐大與氾濫，使用者透過傳統關鍵字搜尋引擎通常無法於第一時間獲得符合本身需求的訊息，旅遊資訊亦如此。有鑑於此，我們建置了 ISWSS 系統讓使用者在查詢旅遊行程時，減少時間的耗損以及增加使用者獲

得訊息的滿意度。我們透過知識本體來整合、歸納各家旅遊業者所推出的旅遊行程資訊，並結合語意網路服務的應用，讓網路服務能夠達到自動發現、自動整合以及自動調用的功能，因而改善了傳統搜尋方式所帶來的資料過於龐大且繁雜、相關性低、搜尋時間過長的問題。總之，本系統可運用於現今的旅遊業，整合旅遊行程的資訊，方便使用者的查詢，也可以為企業帶來更多的經濟效益。

雖然我們所開發的 ISWSS 系統能夠達到整合旅遊行程資訊的功能，透過語意網的應用提高使用者搜尋效率，但仍然存在某些條件限制。例如，現今網路服務的發佈依然不夠普遍，旅遊業者所推出的旅遊行程資訊不一定會發佈於註冊中心供人連結及取得，原因在於業者不一定願意提供其旅遊行程的相關資訊給予整合，當中可能牽涉到知名度等相關問題。另外，由於虛擬機器的相關研究尚未普及，因此未來虛擬機器的實際應用如果有更多的探討，相信能夠讓網路服務自動化更趨於完美。未來如果能夠針對這幾方面問題做更進一步的研究與發展，勢必能夠改善消費者在做搜尋與比較各個旅遊行程資訊時所耗費的時間，進而更增加搜尋的效率。

## 參考文獻

- [1] 王文傑、陳盈潔編譯，”建置 J2EE 應用系統/活用 BEA WebLogic 伺服器”，台灣培生教育出版股份有限公司，2003 年。
- [2] 王治立，”旅遊語意網整體服務系統之建置”，大葉大學資訊管理研究所碩士論文，2004 年。



- [3] 許慶堂, "基於詮釋網路服務之整合型購物決策資訊架構", 南華大學資訊管理研究所碩士論文, 2006年。
- [4] 戚玉樑, "網路服務技術導論", 台北, 全華科技圖書有限公司, 2004年。
- [5] 蘇嶸學, "以本體為基礎的內容感知系統應用於電子型錄管理", 中原大學資訊管理研究所碩士論文, 2006年。
- [6] Burstein, M., Bussler, C., Zaremba, M., Finin, T., Huhns, M. N., Paolucci, M., Sheth, A. P., and Williams, S., "A Semantic Web Service Architecture," *IEEE Internet Computing*, September/October, 2005, pp.72-81.
- [7] Brittenham, P., Cubera, F., Ehnebuske, D., and Graham, S., "Understanding WSDL in a UDDI Registry: How to Publish and Find WSDL Service Descriptions," IBM developer Workers, [http://turtle.ee.ncku.edu.tw/~lok/course\\_reports/Web%20Services/WSTK3.0/document/UnderstandingWSDLinUDDI.pdf](http://turtle.ee.ncku.edu.tw/~lok/course_reports/Web%20Services/WSTK3.0/document/UnderstandingWSDLinUDDI.pdf), August, 2001.
- [8] Chandrasekaran, B., Josephson, J. R., and Benjamins, V. R., "What Are Ontologies, and Why Do We Need Them?," *IEEE Intelligent Systems*, 1999, pp.20-26.
- [9] Eriksson, H., "The Semantic Document Approach to Combining Documents and Ontologies," *International Journal of Human Computer Studies*, 2007, pp.624-639.
- [10] Fensel, D., Benjamins, V. R., Motta, E., and Wielinga, B., "UPML: A Framework for Knowledge Sytem Reuse," In *Proceedings of the 16th International Joint Conference on AI*, Sweden, 1999, pp. 16-21.
- [11] Gruber, T. R., "Ontolingua: A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition* 5(2), 1993, pp. 199-200.
- [12] Guarino, N., "Formal Ontology, Conceptual Analysis and Knowledge Representation," *International Journal of Human-Computer Studies*, 43, 1995, pp. 625-640.
- [13] Kwon, O. B., "Meta Web Service: Building Web-Based Open Decision Support System Based on Web Services," *Expert System with Application* (24), 2003, pp.375-389.
- [14] Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K., "OWL-S: Semantic Markup for Web Services," W3C Recommendation, <http://www.w3.org/Submission/OWL-S/>, 2004.



- [15] McGuinness, D. L. and Harmelen, F. V., "OWL Web Ontology Language Overview," W3C Recommendation, <http://www.w3.org/TR/owl-features/>, 2004.
- [16] Noy, N. F. and McGuinness, D. L., "Ontology Development 101: A Guide to Creating Your First Ontology," <http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf>, 2007.
- [17] Paolucci, M. and Sycara, K., "Autonomous semantic web services," *IEEE Internet Computing*, September/October, 2003, pp.34-41.
- [18] Swartout, B.R., Patil, K., and Knight, T.R., "Toward Distributed Used of Large-Scale Ontologies," *Ontological Engineering*, AAAI-97 Spring symposium series, 1997, pp.138-148.
- [19] Sycara, K., Paolucci, M., Srinivasan, N., Kawamura, T., Payne, T. R., and Srinivasam, N., "OWL-S/UDDI Matchmaker," <http://www.daml.ri.cmu.edu/matchmaker/>, 2007.
- [20] Sycara, K., Paolucci, M., Ankolekar, A., and Srinivasan, N., "Automated Discovery, Integration and Composition of Semantic Web Services," *Journal of Web Semantics*, 2003, Volume 1, Issue 1, pp. 27-46.
- [21] Sycara, K., Martin, D., McGuinness, D. L., McIlraith, S., and Paolucci, M., "OWL-S Technology for Representing Constraints and Capabilities of Web Services," <http://www.w3.org/2004/08/ws-cc/dmowls-20040904>, 2004.

