管　理　與　系　統
第十二卷 第二期　民國九十四年四月
93 – 121頁

Journal of Management & Systems
Vol. 12, No. 2, April 2005
pp. 93-121

旅遊語意網整體服務系統之建置

The Construction of an Integrated Service System for Travel Semantic Web

陳鴻文 Hown-Wen Chen　　張隆池 Long-Chyr Chang
王治立 Chih-Li Wang　　葉木水 Mu-ShawnYeh
大葉大學資訊管理研究所

Institute of Information Management, Da-Yeh University

(Received August 9, 2004; Revised November 1, 2004)

摘要：如何有效的利用網際網路上的資訊及服務，已成爲現今網路技術中相當重要的課題。「語意網」的架構主張將網路上文件內容以「知識本體」方式來將有意義的內容結構化，以建立起一個資訊明確及知識可重複使用的網際空間，進而促使網路上的資源及服務更易取得和分享。本研究嘗試從服務整合的觀點，設計旅遊語意網整體服務系統架構，目的是提供使用者便利、個人化、且幾近自動化的旅遊行程規劃與旅程服務。此雛型系統由使用者介面模組、旅遊服務排程模組、旅遊服務提供模組及旅遊服務註冊中心所組成；系統運作方式是由使用者提出旅遊安排的個人喜好需求及資源限制條件，接著旅遊服務排程模組依據需求，和不斷地透過資訊檢索的動作，使用綜合評估來篩選景點，並由旅遊行程規劃器進行行程安排。一旦使用者確認該行程規劃後，行程服務預約機制將會自動進行相關行程服務的預訂事宜。最後，系統即可將完整的旅遊行程規劃暨服務預約狀態，透過使用者介面模組呈現給使用者。本研究的主要貢獻計有：(1)設計暨建立台灣地區城市旅遊知識本體之雛型；(2)使用知識本體的樹狀結構，有效提昇檢索的效能及準確性；(3)將貪婪法理論、排列組合與綜合評估法則進行整合，以達到客製化的旅遊行程安排；(4)與服務預約機制相結合，藉以達到旅遊整體服務的功效。

關鍵詞：旅遊語意網、知識本體、資料檢索、旅遊行程規劃、網路服務

Abstract：The issues of effectively using Web information and services is getting more important among Internet technologies. The Semantic Web was thus proposed to facilitate resources on Internet re-usable and sharable using the ontology approach. In this paper, a novel prototype system, “Travel Semantic Web of the Integrated Service System Architecture (TSWISS)”, was proposed to provide an integrated Web service for tourists making personalized travel plan more easily. TSWISS comprises four modules: user interface module, travel scheduler module, service provider module, and service registration center module. This research mainly focuses on three folds: the ways of defining and constructing Taiwan tour ontology, an efficient mechanism of information retrieval on Semantic Web, and a personalized trip scheduler. Relevant tour information was first collected to build domain and information ontology. Then, the constructed information ontology was translated into relation databases from original DAML format to make indexing more efficient. Furthermore, we utilized the top-down management of Web services to store the domain and information ontology in the registration center in advance. A novel information retrieval system was constructed using tag hierarchy within ontology and then was compared its performance with Google. Experimental cases were used to illustrate the results of the proposed system being more accurate and less redundant. Therefore, the novel system architecture based on Semantic Web could be expected to provide personalized travel plans for tourists more efficiently by using the designed ontology and proposed information retrieval mechanism.

Keywords **:** Travel Semantic Web, Ontology, Information Retrieval, Trip Planning, Web Service

1. 緒論

自政府實施「周休二日」政策以來，單年度（民國九十一年）國人國內旅遊次數累計達一億零六百萬人次以上，較上一年度成長了9.1%；而旅遊方式，則是以個別旅遊與自行規劃旅遊行程爲主，佔了89%，且其中利用電腦網路取得旅遊資訊也有明顯增加的趨勢，約佔13% (中華民國交通部觀光局，民91)。同時，截至民國92年6月爲止，台灣地區旅遊網站的設立家數更達600家 (科威資訊，民91)；民國92年電子商務（E-Commerce）市場規模亦達新台幣220.9億元，其中以旅遊產品爲銷售種類的最大宗，佔了48.5% (資策會網站，民93)。由此可知，旅遊產業結合網際網路與電子商務的模式，已形成了現代人生活中新的旅遊經濟型態。

不過，即使旅遊網站爲數眾多且提供各種類的旅遊資訊與服務，但從針對網路使用者所做的調查（葉禮宗，民91）可以發現，網路使用者對於旅遊網站設計最不滿意的部份，主要是「行程內容介紹太簡單」，而且大部份的網路使用者則普遍認爲，「行程規劃」是整個旅遊過程中最重要的核心部份。另外，根據一份訪查30個國內旅遊相關網站的研究（連惠英，民91）也指出，各

網站所提供的旅遊產品，一般是以內建一日遊、二日遊或三日遊等暨定路線的套裝旅遊產品爲主，普遍缺乏針對個別旅遊者，提供客製化的旅遊資訊服務的機制。由此，更加突顯出目前旅遊網站，對於旅遊行程資訊暨服務整合的不足和缺乏彈性。

另一方面，網際網路雖然徹底地改變了電子資訊的應用方式，每個人都可以輕易地經由全球資訊網取得與分享知識；但是它的成功與指數成長的趨勢，卻使得廣大的使用者於資訊的搜尋、存取、呈現以及維護上更加的困難。究其原因，主要癥結在於目前網頁內容的編寫語言，仍以HTML爲主；在此方式下表達出來的網頁內容，大多呈現非結構化，且標籤(tag)無法解釋網頁內容真正要表達的涵意。隨後W3C發表XML (eXtensible Markup Language) 語言，其最大的特性是提供了一個標示結構化的資料形式，讓使用者也可自定具有意義的標籤名稱。然而，利用XML進行知識分享有一個很大的問題，就是標籤名稱的命名仍無共同的標準，因而產生「同名異物」、「異名同物」的困擾。所以，若能在各個不同的資訊來源間，建立一個共通的標準，如此對於知識分享或知識交換的運作，勢必能更加地順暢。

有鑑於此，全球資訊網的創始人Tim Berners-Lee博士提出了新的網際網路架構－語意網(Semantic Web) (高虹，民91；Berners-Lee *et al.*, 2002)，主張將網上有意義的內容結構化，藉由共享的、通用的知識本體（Ontology）之建置，使得網路上的資源及服務更易取得和分享。然而目前有關語意網的研究，大都集中於知識本體與基礎語言如DAML+OIL的探究與設計，有關邏輯推理架構和應用的研究尚在起步中。本研究嘗試從服務提供的觀點，設計「旅遊語意網整體服務系統 (Travel Semantic Web of the Integrated Services System, TSWISS)」，以提供使用者客製化、幾近自動化的旅遊行程規劃與旅程服務。執行過程需以現有全文檢索功能爲基礎，建構出一套知識本體資訊檢索雛型系統；本研究並利用了綜合評估及與使用者互動的方式，來選擇最適合的網際網路旅遊服務，最後則實作出一個能夠針對個別旅遊者，提供個人化的旅遊資訊服務的整合機制。希望藉此能促進網路自動化服務的發展，以建立起一個資訊充份分享及知識可重覆使用的網際空間。

2. 相關研究

2.1 知識本體

「知識本體」一詞，最早是出現在哲學的領域中。簡單來說，知識本體就是清楚的描述一個領域內所表達的概念和與概念有關的特徵 (property) 及屬性 (attribute)，再加上屬性的限制(constraint)，和依此分類法所產生的實體 (instance) (Chandrasekaran *et al.*, 1999)。

Guarino對知識本體則提出下列的定義 (Guarino, 1998)：「知識本體是一個邏輯理論 (logical theory)，用來說明一系列字彙的特定意義 (intended meaning)。而爲了達成描述此一概念化範疇(conceptualization) 的目的，使用一系列的邏輯語言來表達，然而此語言基於其知識本體行爲

(ontological commitment) 的限制，必須從邏輯語言中找出適當的特定模型 (intended models) 來說明概念化範疇的特定意涵」。由此可知，知識本體應具備的特性有：可描述人世間所有事物，達到語法的互通性 (syntactic interoperability) 及可達到語意的互通性 (semantic interoperability)。

另外，根據Noy等學者的定義 (Noy *et al.*, 2004)：「知識本體」就是針對特定課題之一群知識項目的總和，包括了字彙、字彙間的語意關係，和邏輯推論規則。「知識本體」使得語意網既能表達資料和對資料進行推理的規則，也能接受任何現有知識表徵的推理規則；如此，使得網路代理者能經由自動推理來選擇下一個動作，同時具備回答問題的能力。

2.2 台灣旅遊知識本體庫的相關研究

目前國內將知識本體應用於旅遊方面的研究並不多，其中已有採用類似知識本體的形式，應用於旅遊行程安排 (梁書豪，民90)。而後繼的研究者皆以此爲範本，諸如將其概念應用於具有時間限制下的個人化旅遊行程規劃的研究 (吳俸昌，民91)；認爲要完成以代理人爲主的旅遊規劃系統，首先必須定義出一套完整的旅遊知識本體，使得不同的代理人能夠利用知識本體來互相溝通，藉以協助使用者完成旅遊行程的規劃；並首將多個旅遊相關知識本體彙整成旅遊知識本體庫（tourism ontology）引用至論文中。亦有研究者將此概念應用於對話介面人代理人的旅遊行程推薦的研究 (葉禮宗，民91)。

本研究以上述研究者所提之知識本體概念爲基礎，加以修改後應用於提出之旅遊語意網整體服務雛型系統，成爲更趨完備的旅遊知識本體，藉以達成自動化服務的目的。

2.3 旅遊排程資訊系統的相關研究

旅遊排程並非只是單純的問題，除了需逐一檢查具高複雜度的巨量空間路徑外，尚須考量空間內之各路徑與旅遊者所提出各種條件之組合程度何者爲最佳。爲能使各種自然資源與旅遊者所提出之眾多目標與條件都能被滿足，**綜合評估法**（連惠英，民91）則同時考量到旅遊排程問題的五個構面：

(1) 空間：受旅遊者偏好的影響，由旅遊起始點到旅遊終止點會存在眾多可能景點，也就是會形成由出發點到眾多可能景點間的旅遊路徑，這樣的空間路徑之列舉與追蹤問題，都是進行旅遊排程時需要考慮到的空間問題。
(2) 偏好：含旅遊者本次之遊憩偏好，包括希望的遊程安排鬆緊偏好、娛樂性、價格和距離因素之個人感受偏好，因關係到個人價值觀與個人感受問題，系統無法預設，需透過使用者介面由旅遊者直接告知。
(3) 時間和預算的資源限制。
(4) 各景點遊玩之金錢成本和時間成本。
(5) 景點遊玩的最短停留時間。

目前有關旅遊排程問題的研究，大致上可分爲四個方向：(1)是以走過所有被選定景點之旅遊推銷員問題（Traveling Salesman Problem, TSP）(Horowitz, *et al.*, 1990) 的概念來進行行程安排，所謂TSP是指在一有向性的G圖上，找一含“每一頂點”之循環周遊路徑 (tour)，周遊路徑的代價就是這一個周遊路徑上的每一個邊代價的總和，即是找出具最小代價的周遊路徑。所以TSP追求的是“每一頂點都恰只走過一次”的最短路徑，例如「旅遊行程安排及探勘分析之實作」 (陳肇男，民88)。(2)是遵循最小擴張樹（Minimum cost Spanning Tree, MST）方法，MST追求的是以最少的邊來連結所有的頂點且不造成迴路（cycle）(Horowitz, *et al.*, 1990)。(3)是採用Dijkstra最短路徑演算法（廖鴻圖等，民88）來解決行程規劃的問題，因其僅用$O(n^2)$的時間複雜度，即可有效率地把單一起點與其它所有頂點間的最短路徑都找出來，例如「旅遊代理人以協商之方式推薦旅遊行程」(梁書豪，民90)。(4)是以旅遊路由樹（Trip Routing Tree, TRT）演算法加以延伸，來進行旅遊行程安排。所謂TRT演算法是由出發點開始，以樹狀結構搜尋方式逐一測試可能的排列組合；若加入適當的限制，如有序搜尋且不需回頭搜尋，則此類改良式的TRT演算法，在避免重覆搜尋之下，則可使TRT之複雜度的指數時間$O(2^n)$降爲多項式時間$O(n^2)$，例如「智慧型旅遊路線排程系統」(連惠英，民91)。

但由於進行個人化之旅遊行程規劃時，時常需要考慮到旅遊因子眾多，諸如空間、資源、成本、偏好及遊賞時間等因素，故無法先行轉換成加權型最短路徑問題；例如排程中已選取一個溫泉景點後，爲了維持整個行程能確切符合使用者的喜好比例，之後對於溫泉景點之選擇喜好，勢必要調降。如何避免耗用大量時間和記憶體存放各種可能的行程（景點排列組合），而能即時有效地呈現給使用者可行之行程，則是值得本研究探索的課題。

3. 系統設計與研究方法

3.1 台灣地區城市旅遊知識本體之定義

要完成一個以語意網爲基礎的自動化旅遊規劃暨服務系統，首要的就是建構完整的旅遊知識本體。透過旅遊知識本體的建立，使得系統能夠瞭解旅遊者的旅遊需求，準確地規劃出符合旅遊者需求的旅遊行程暨服務。在本研究中，將知識本體的內容區分爲領域知識本體 (domain ontology) 及資訊實體 (information ontology) 兩類，前者明確定義了某一領域中詞彙及詞彙間的關係；而資訊實體則遵循領域知識本體的定義，以類似DAML知識本體語法來實質描述特定網頁或文件之內容。而圖1呈現旅遊語意網中，台灣地區城市旅遊知識本體的樹狀結構，涵蓋了住宿、景點、交通領域知識本體，清楚表達了旅遊領域中的相關資訊。

在知識本體建構部份，自網際網路蒐集有關旅遊語意網的相關資料，至針對資料進行分析與歸納的過程，本研究皆採用人工的方式。以下將以交通工具知識本體爲例，簡述本研究知識本體表示的方式。

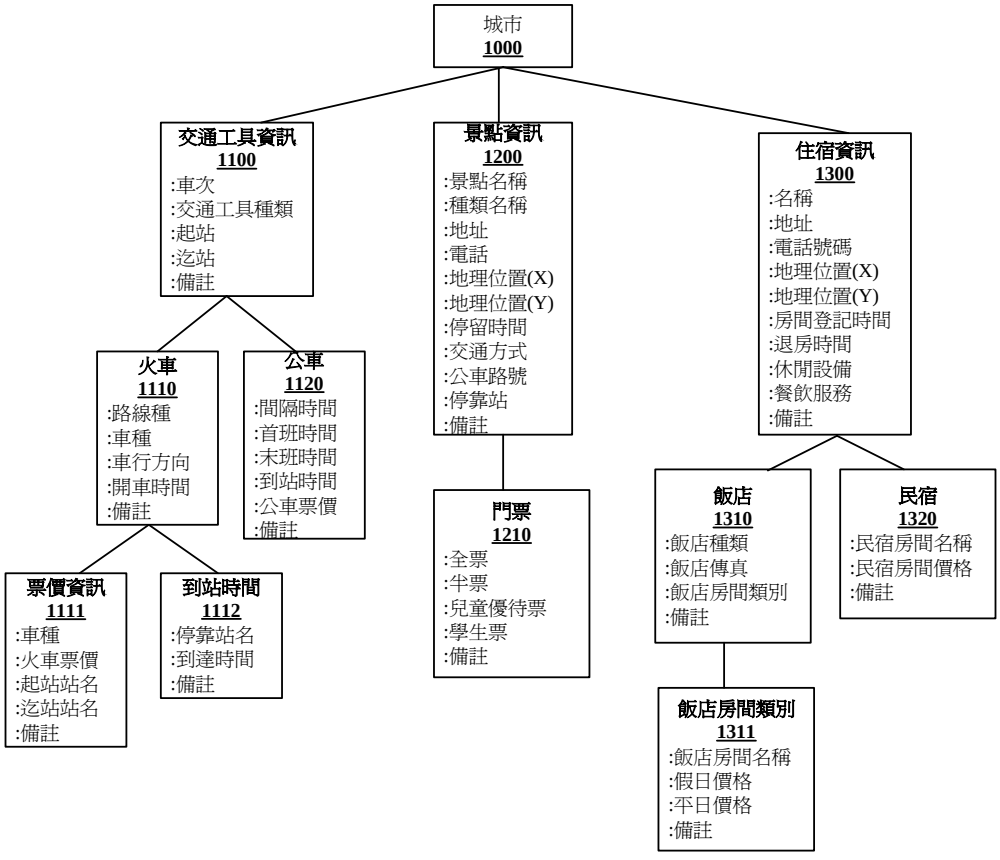


圖1　本研究採行之城市旅遊領域知識本體架構

3.1.1 領域知識本體

領域知識本體明確地定義了某一領域中關鍵字詞及其間關係。如圖1所示，矩形代表領域中的類別，而每一類別包含兩部份：類別名稱與屬性。類別名稱在其對應編號下加入底線，而屬性則是用來說明類別的意義，以「:」作爲開頭。在類別間則可以存在繼承與參考的關係。

由於本研究對於知識本體的描述採用DAML語言。因此，利用protégé2000的圖形化介面，如圖2所示，建構出知識本體的架構後，再轉換以DAML語法格式來呈現。如圖3所示爲景點結構定義部份。

3.1.2 資訊知識實體

如圖4所示以DAML語言所描述之景點資訊爲例，其中資料儲存格式皆是依據上述領域知識本體所定義之資料結構，而再增加實際資料的描述。

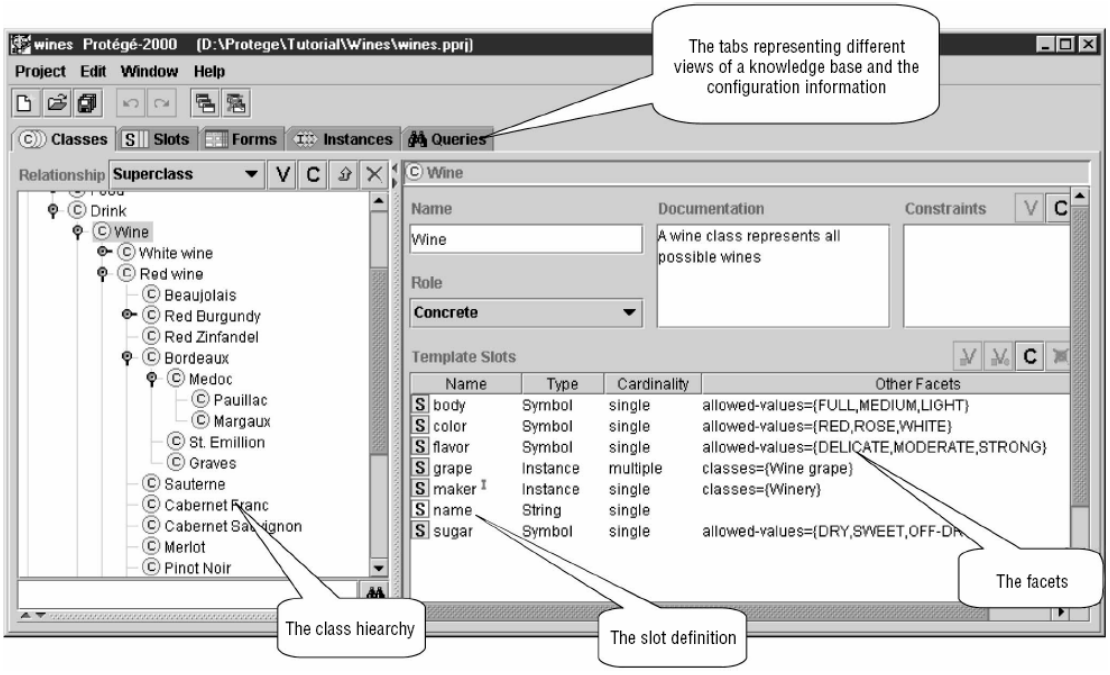


圖2 Protégé-2000操作介面範例

```
<rdf:RDF
   xmlns:XMLSchema ="http://www.w3.org/2000/10/XMLSchema#"
   xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:daml_oil ="http://www.daml.org/2001/03/daml+oil#"
   xmlns:landscape ="http://landscape#"
   xmlns ="http://landscape#">
<daml_oil:DatatypeProperty rdf:ID="公車路號">
   <daml_oil:domain rdf:resource="#景點資訊"/>
   <daml_oil:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</daml_oil:DatatypeProperty>
<daml_oil:DatatypeProperty rdf:ID="地理位置(X)">
   <daml_oil:domain rdf:resource="#景點資訊"/>
   <daml_oil:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</daml_oil:DatatypeProperty>
<daml_oil:Class rdf:ID="門票資訊">
</daml_oil:Class>
<daml_oil:DatatypeProperty rdf:ID="兒童優待票">
   <daml_oil:domain rdf:resource="#門票資訊"/>
   <daml_oil:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#integer"/>
</daml_oil:DatatypeProperty>

</rdf:RDF>
```

圖3 以DAML語法之部份領域知識本體（景點結構定義）

```
<rdf:RDF
xmlns:XMLSchema ="http://www.w3.org/2000/10/XMLSchema#"
xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:daml_oil ="http://www.daml.org/2001/03/daml+oil#"
xmlns:landscape ="http://landscape#"
xmlns ="http://landscape#">
<landscape:景點資訊 rdf:ID="landscape0616_00097">
<landscape:交通方式>公車</landscape:交通方式>
<landscape:地址>台北市文山區新光路二段30號</landscape:地址>
<landscape:景點名稱>木柵動物園</landscape:景點名稱>
<landscape:種類名稱>自然風景</landscape:種類名稱>
<landscape:附註說明>台北市立動物園創設於1915年，舊址圓山。</landscape:附註說
明>
<landscape:電話>(02)2938-2300</landscape:電話>
<landscape:門票 rdf:resource="#landscape0616_00098"/>
</landscape:景點資訊>
<daml_oil:Ontology rdf:ID="">
</daml_oil:Ontology>
```

圖 4　以 DAML 語法表示之部份交通資訊知識實體

3.2 系統架構

旅遊語意網整體服務系統架構TSWISS，由四個部份所組成：使用者介面模組、旅遊服務排程模組、旅遊服務提供模組及旅遊服務註冊中心 (如圖5所示)。

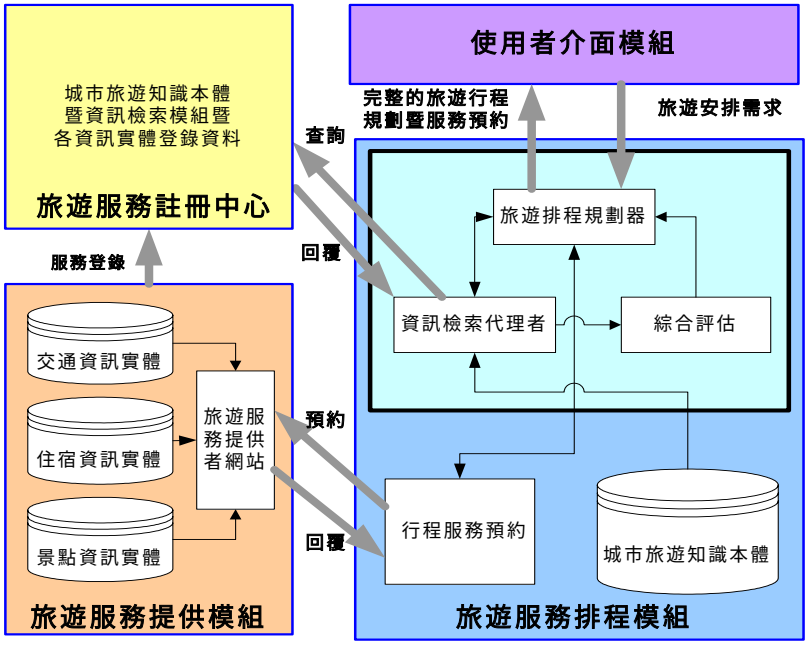


圖5　旅遊語意網整體服務系統架構

系統運作方式是由使用者透過介面模組提出旅遊安排的個人喜好需求及資源限制條件，接著旅遊服務排程模組依據使用者需求，使用綜合評估篩選景點，並由「旅遊行程規劃器」進行行程安排。如系統不能規劃出合理的行程，則透過「使用者介面模組」回應失敗的原因，並要求使用者調整資源限制後，再次重新進行規劃。在進行行程規劃的過程，「旅遊服務排程模組」需不斷地透過資訊檢索的動作，以獲取「旅遊服務提供模組」所囊括之網路上相關服務知識本體的摘要或詳細資訊。

接下來，分別向「旅遊服務註冊中心」的資訊檢索機制下達查詢的請求。經處理後，「旅遊服務註冊中心」回覆符合旅遊者需求景點詳細資訊給旅遊服務排程模組」；一旦使用者確認該行程規劃後，「旅遊服務排程模組」中的「行程服務預約」機制將會嘗試進行相關行程服務的預訂事宜，如旅館訂房、火車票預訂與景點門票訂購等。即系統將再次要求「旅遊服務註冊中心」的服務式知識本體回傳一份用來描述具體服務之WSDL (Web Services Description Language) 的文件，並經由「行程服務預約功能」自動將服務要求轉換成對應之SOAP (Simple Object Access Protocol) 指令後，即可溝通旅遊服務提供者來進行行程服務預約事宜。最後，系統即可將完整的旅遊行程規劃暨服務預約狀態，透過「使用者介面模組」呈現給使用者。如圖6清楚顯示了台灣地區城市旅遊知識本體的運作方式。

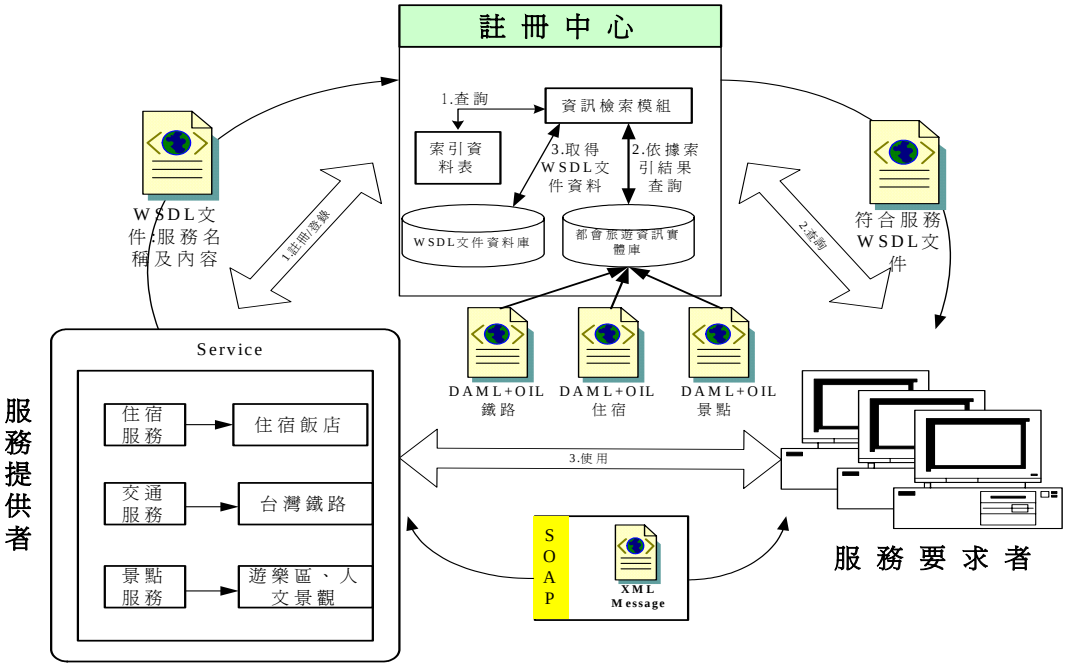


圖 6　台灣地區城市旅遊知識本體的資料運作流程

3.2.1 使用者介面模組

使用者介面模組主要是負責與使用者互動，包括「旅遊需求介面」負責接收使用者提出的旅遊服務需求，和「系統顯示介面」以將回應旅遊服務結果給使用者。其中旅遊需求介面由基本資源需求、住宿需求、交通需求與景點類型的需求四部份所組成。圖7即描述了本系統提供使用者表達的旅遊行程需求。

3.2.2 旅遊服務排程模組

旅遊服務排程模組負責完成使用者所提出的旅遊服務需求。旅遊服務排程模組主要分成行程規劃暨服務預約兩部份，前者是依據使用者所提出的旅遊服務需求，藉由綜合評估法則刪除不符合需求的景點，再利用旅遊排程規劃器來進行旅遊排程規劃的工作。第二部份則依據使用者確認本系統所規劃的旅遊排程後，進行旅遊行程中各項服務的預約事宜，如火車票的訂購、旅館的預約、景點門票的預訂等等。接下來，將詳細說明旅遊服務排程模組中的各項功能。

3.2.2.1 旅遊景點選擇

面對諸多可供遊覽的景點，旅遊者並不一定每個景點都要去遊玩。所以，當使用者提出了旅遊需求之後，旅遊服務排程模組首先針對旅遊景點進行分析，以找出符合旅遊者需求的景點，以便能夠在時間及預算的限制下，安排符合旅遊者個別需求之行程。爲了有效分析旅遊景點，本研究採用了2.3節所描述的綜合評估概念。表1爲本系統內部所使用的評估規則。

圖8的景點選擇演算法，主要是在開始進行排程之前，系統先行計算在符合預算、時間及使用者偏好的情況下，某城市內可以選擇的可能最大景點數目。主要功能是依照權重來刪除權重較小的景點後，找出通過剩餘景點的「近似最短路徑」行程，並且計算此行程所需花費的經費與時間。所謂「近似最短路徑」的決定，是採用貪婪式經驗法則 (greedy heuristics) 的作法，以期大幅節省運算的時間。依照權重再刪除景點，直到找出合乎預算及時間限制的行程。比如說本來此城市有20個景點，經過逐步減少個數的計算過程後，發現爲了符合預算、時間及使用者偏好的條件下，最多只有13個景點可走，下一步就只需排列組合此13個景點來找出最佳行程，故可以大幅降低排列組合的複雜性。

3.2.2.2 景點行程的安排

依據旅遊天數、欲停留的城市與旅遊類型偏好，以及時間、預算的限制，系統已篩選出可能符合旅遊者需求的景點。此時，旅遊者可藉由排列組合的方式與綜合評估法則來解決景點路徑安排的問題。本系統行程安排的主要步驟爲：

步驟1：先找符合條件下之「近似最短路徑」(採用圖8之景點選擇演算法)。

步驟2：開始排列近似最短路徑之所有景點，以產生新行程並判斷是否合乎條件。

步驟3：如果預設時間到達或已找到預期的行程數量就可停止。

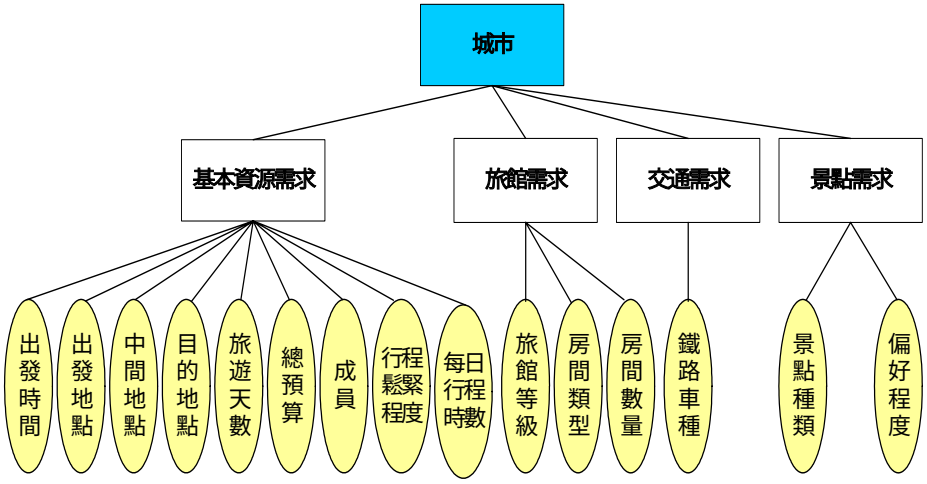


圖7　本系統提供之使用者表達旅遊行程需求的種類

表 1　本研究採行的綜合評估法則

規則類別	代號	規則內容概說
正規法則	R1	景點拜訪時間以十分鐘爲單位正規化。
目標法則	R2	依景點偏好程度比例來規劃遊玩景點的數目。
	R3	旅遊者可決定每日遊玩時間，但預設爲8小時。
	R4	依旅遊者之行程鬆緊程度（鬆弛、適中與緊湊）來設定景點參觀的時間，分別爲爲最短停留時間的1.6倍、1.3倍與1.0倍時間。

目的: 搜尋近似最大可容納的景點數目

步驟1: 預算經費先扣除必要之火車費用。

步驟2: 排除不符合使用者偏好類型的景點。

步驟3: 選取所有景點中權重最高者，視爲必選景點。

步驟4: 逐一選取加入新景點，其方法爲與已選定景點所形成路徑頭或尾點距離最短者，再串聯加入以形成「近似最短路徑」。

步驟5: 估算時間資源及預算，若合乎預算即可停止。

步驟6: 若超出預算，則需刪除一個景點；刪除考慮因素依序爲

6.1: 刪除何類景點，方符合使用者偏好景點比例。

6.2: 刪除此類景點中權重最小者。

步驟7: 原路徑在刪除景點後之所有景點，再重新依照**步驟4~步驟7**形成新路徑。

圖8　景點選擇演算法

步驟4：如果沒找到任何行程，或找到的行程數量不足，則再從行程景點中刪除一個權重最低的景點，並從**步驟1**重新開始，直到停止。

3.2.2.3 行程服務預約

行程服務預約的功能，主要是使用者確認了旅遊排程規劃器所安排的行程之後，自動爲旅遊者進行行程中的各項服務預約的動作，包括了火車票的訂購、旅館的訂房及遊玩景點門票的預購等。此行程服務預約會依照之前資訊檢索代理者回傳的URL及預約所需資料格式，再搭配使用者事先輸入之個人資訊，如身份證字號、信用卡卡號等，自動完成服務預約的工作。

在語意網的架構之下，每一旅遊服務提供者皆有各自獨立的服務資訊實體，以便提供自動化的服務。因此，行程服務預約會依據使用者所確認的旅遊行程，經由旅遊服務註冊中心的WSDL文件資料庫查詢結果，再向特定的旅遊服務提供者提出行程服務預約的動作；當旅遊服務提供者接收到行程服務預約的請求之後，訂票代理人確認是否可提供適合的服務給使用者，並回覆適當訊息給服務預約代理人，以完成行程服務預約的事宜。以台中二日遊爲例子，假設4位已成年的旅遊者希望在2004年7月5日出遊，由台北出發，目的地爲台中的二日遊行程，出發時間是早上8:00，搭乘的火車爲莒光號。當行程服務預約機制接收到其行程服務預約的流程，如圖9所示。

3.2.3 旅遊服務註冊中心

註冊中心分爲都會旅遊資訊實體庫、階層編號表、旅遊詞彙庫、WSDL文件資料庫與資訊檢索模組五大部份。如圖10所示，分別說明如下：

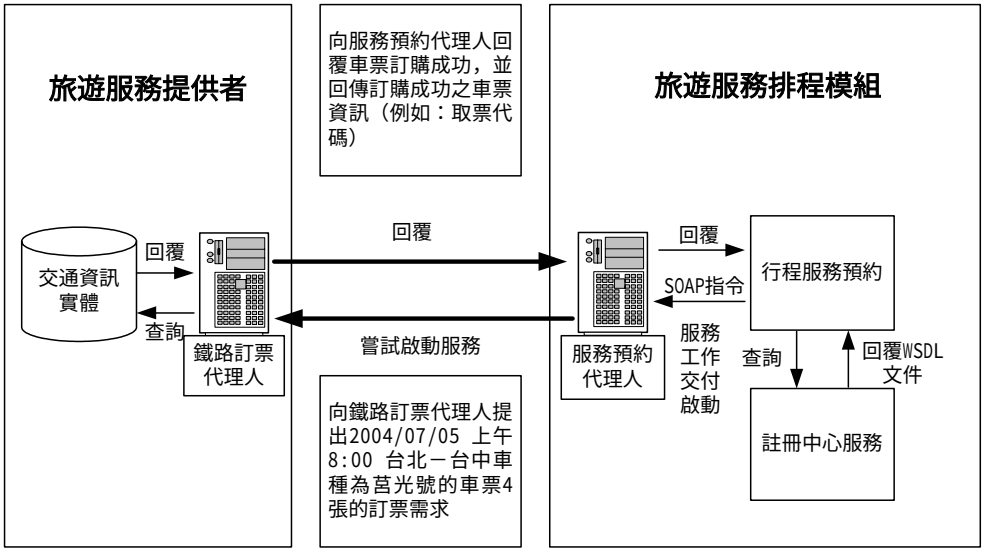


圖9 行程服務預約流程圖

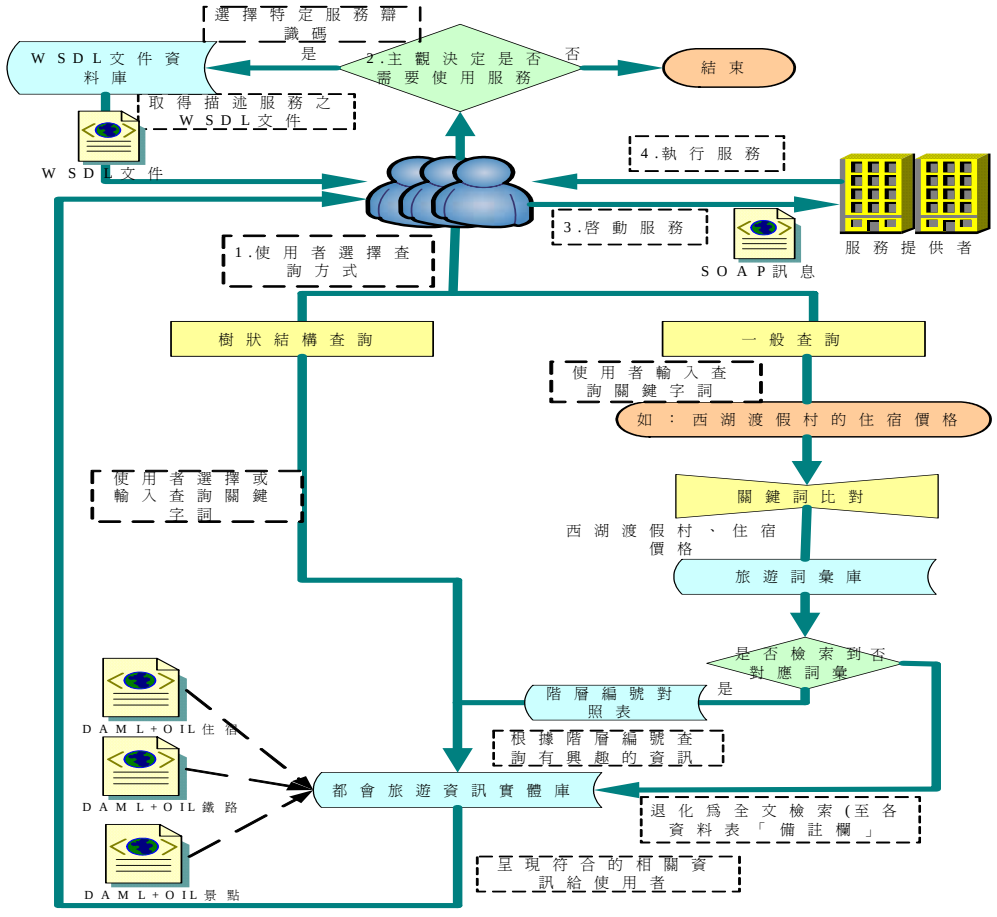


圖10　資訊檢索模組運作流程圖

(1) 都會旅遊資訊實體庫

參與旅遊語意網的企業或廠商，需按照事先定義的領域知識本體規格，自行完整描述本身所提供服務之資訊實體，這些資訊實體的主要內容將經由主動註冊，或由註冊中心利用網路爬蟲(crawler) 代理者取回，再經過資訊實體的剖析處理，轉換爲一般關連式資料庫格式後，被儲存在都會旅遊資訊實體庫，以供資訊檢索模組查詢。

(2) 階層編號表

本系統針對都會旅遊資訊實體庫內所儲存的關鍵字詞，及知識本體所定義的標籤字詞，皆各給予一個階層編號，此階層編號的用意在於使系統能輕易的取得關鍵字詞在知識本體樹狀結構的位置。階層編號的編碼採用四個數字，可表示出樹狀的階層爲四層：第一層爲城市；第二層爲交通工具資訊、住宿資訊、景點資訊，以下類推。諸如：「公車」類別的階層編號爲「1120」，代表「公車」的樹狀結構的位置順序爲「城市」－>「交通工具資訊」－>「火車」。

階層編號表的資料結構，如圖11所示，共分爲三個欄位：第一個欄位序號，用來記錄各個關鍵字詞的階層編號在編號對照表的起始點與結束點，這是因爲一個標籤字詞或關鍵字詞，可能會出現在台灣地區城市旅遊知識本體樹狀結構的不同位置，如圖1中的「到站時間」就可能擁有編號「1120」和「1112」。爲了表示出不定個數的對應編號，本研究採用將這些對應編號連續紀錄在編號對照表中，而利用序號來標示其起點和終點位置。第二個欄位爲階層編號，代表關鍵字詞所屬的編號；第三個欄位屬性，則代表關鍵字詞出現在資料庫表格內的欄位順序。需注意的是屬性詞彙將沿用類別名稱的階層編號；換言之，圖1中位於同一矩形內所有的詞彙階層編號皆相同。此階層編號表之使用方式在後續(5)資訊檢索模組中有更進一步的說明。

(3) 旅遊詞彙庫

「旅遊詞彙庫」的資料結構，分爲四個欄位：「關鍵字詞」即爲有關都會旅遊資訊實體的詞彙，可能爲知識本體樹狀結構的標籤或資訊實體內的詞彙；「起始編號」爲詞彙的階層編號在「階層編號表」的起點序號；「結束編號」爲詞彙的階層編號在「階層編號表」的終點序號；最後「標籤屬性」則代表詞彙在都會旅遊資訊實體的屬性，「1」代表爲標籤詞彙，「0」則代表詞彙爲「資訊實體類別」。

透過此「旅遊詞彙庫」的輔助，在資訊檢索模組對於服務式資訊實體庫進行檢索之前，大部份情況即可將輸入關鍵字組轉換成唯一的階層編號，且對應到同一株知識本體樹狀結構中。根據資訊檢索的慣例，上層標籤及描述可用來協助確定符合檢索條件的資訊實體，亦可明確界定此次檢索的範圍，僅侷限於最低階(內層)標籤之實質內容資訊，如此運作方式預期足以提高資訊檢索的速度與效率。

另外，在每一個資料表中皆設定了一個「備註欄位」，主要用來儲存除了關鍵字詞以外的說明資訊。若關鍵字詞無法在上述的「標籤類別」與「資訊實體類別」檢索到符合的詞彙，系統此時將退化成全文檢索功能，轉而至各資料表的「備註欄位」檢索符合的資訊。

(4) WSDL文件資料庫

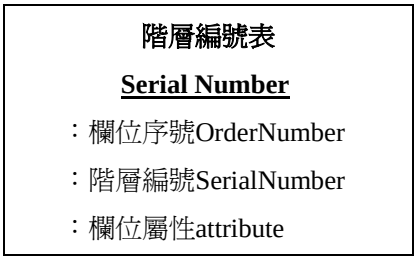


圖 11　階層編號表資料結構

當提供服務的企業或廠商在進行註冊的過程裡，將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。故需求者可依據資訊檢索模組所的查詢結果，取得所需要服務的WSDL文件，並得以進一步使用該服務。

(5) 資訊檢索模組

首先，使用者需要選擇資訊實體檢索的方式，分爲「樹狀結構查詢」與「一般查詢」。在「樹狀結構查詢」方面，使用者進行資訊實體檢索的過程，完全依照都會旅遊資訊實體的樹狀結構。因此，在資訊實體檢索系統後端程式接收到屬性值，可立即得知標籤在樹狀結構的階層編號，而不需再到「旅遊詞彙庫」進行比對，直接進行關鍵字詞的檢索即可。至於「一般查詢」方面，使用者輸入需要查詢的關鍵字詞之後，進行關鍵字詞比對。其方式是將使用者所輸入的關鍵字詞與旅遊詞彙庫內所定義的詞彙，依序進行兩種類別的比對及篩選動作。第一種爲「標籤類別」比對，先將關鍵字詞與標籤屬性爲「1」的詞彙進行比對及篩選；若沒有符合，則進行第二種「資訊實體類別」比對，將字詞與標籤屬性爲「0」的詞彙進行比對篩選。

進行「一般查詢」時，由於同一個關鍵字詞有可能出現在整個旅遊資訊實體樹狀架構的不同位置；因此，一個關鍵字詞可能會對應到好幾個階層編號，而這些階層編號將事先存放在「階層編號表」內，以備之後需要時可方便取得。當系統接收到使用者查詢的關鍵字詞後，首先，系統會到「旅遊詞彙庫」取得關鍵字詞階層號的起始與結束編號；再根據起始與結束編號到「階層編號表」，取得代表關鍵字詞的所有階層編號及其儲存在資料表的屬性。

然而，當使用者輸入兩組以上的關鍵字詞時，在取得關鍵字詞的階層編號之後，系統必須要能判斷各組關鍵字詞之間，是否存在階層關係或相鄰關係。若在「旅遊詞彙庫」中進行兩種類別的資訊檢索後，仍然無法檢索到符合關鍵字詞的資訊時，則至「都會旅遊資訊實體庫」內的各個資料表格的「備註欄位」，進行全文檢索，以搜尋符合使用者需求的資訊。

4. 系統實作與成果

本研究建構之系統，是在Windows2000 Server下，以Tomcat3.2、Apache1.3.27、JSDK2.0及JDK1.3.1作爲系統運作的平台；旅遊資訊實體格式轉換的部份，採用惠普公司的Brian McBride所開發的JENA API作爲DAML的剖析工具；至於使用者介面的設計，則以FrontPage2000、JAVA Servlet網頁語言搭配JavaScript來開發。另外，則使用MS SQL Server 2000資料庫來儲存資料。

本研究將實作分爲三部份，分別爲旅遊資訊實體的建構與轉換、線上資訊實體檢索系統的建置、及客製化旅遊排程暨服務預約。

4.1 旅遊資訊實體的建構與轉換

由於網路上缺乏旅遊相關的資訊實體，因此在本研究中，將充份利用網路上各搜尋引擎的搜

尋功能，來蒐集旅遊排程中所需要的各類資訊：交通工具、住宿及景點，再輔以人工的方式對於各項資料進行分析。接著，利用 protégé2000 來建構知識本體及資訊實體，最後則利用 protégé2000 提供的 DAML+OIL 外掛模組，將各知識本體以 DAML 語言格式呈現。以圖 12 爲例，即以此法自動產生之 DAML 語言格式來呈現台北木柵動物園之資訊實體。

由於服務提供者是將資訊實體以 DAML 檔案格式提供給註冊中心，而本系統的都會旅遊資訊實體庫，卻是以資料庫的格式來儲存資訊實體，以便提高日後資訊檢索的效率。因此，在 DAML 檔案格式與資料庫格式之間，必需進行格式轉換的動作。至於此動作，本研究是利用惠普公司的 Brian McBride 所開發的 JENA API，來剖析 DAML 檔的內容；這是因爲 JENA 提供了許多的函式，讓研究者可以輕易地操作與讀取 DAML 檔內的類別、屬性及屬性值。

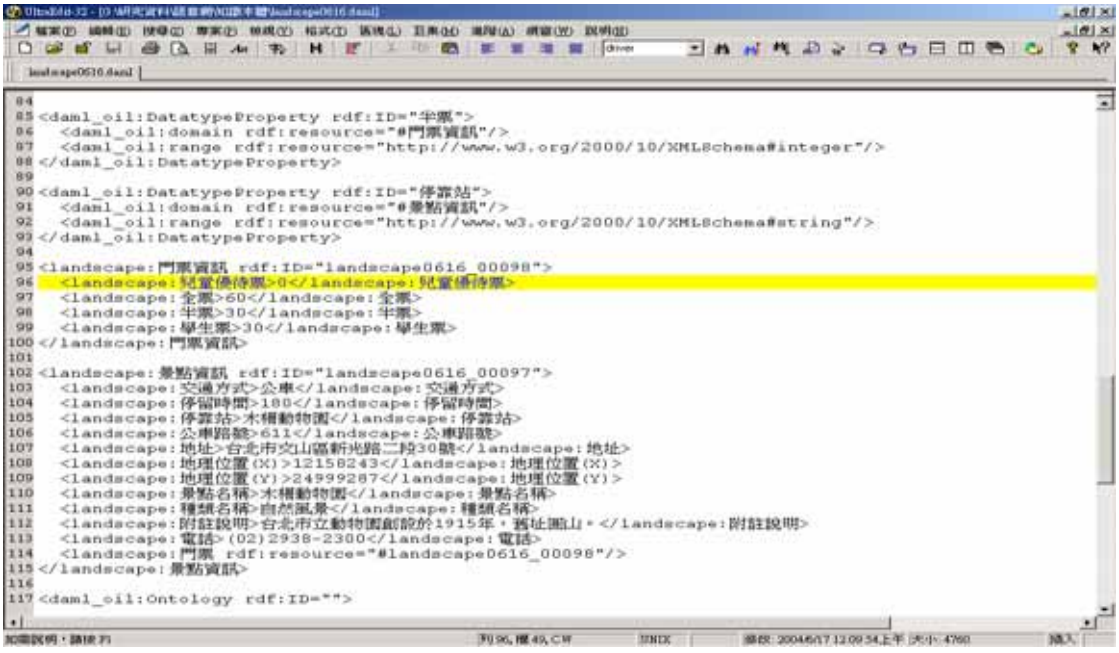
以下將以「交通工具資訊」類別之處理爲例，說明如何將 DAML 檔中的資訊實體轉換到資料庫。假設目前 DAML 檔儲存三筆記錄，如圖 13 所示，第一筆爲「車次:101、起站：Kee Lung、迄站：Ping Tung」，第二筆爲「車次:1003、起站：Sung Shan、迄站：Kaohsiung」，第三筆爲「車次:1009、起站：Kee Lung、迄站：Kaohsiung」。

圖 14 則是轉換程式的部份程式碼。首先，須宣告虛擬的「DAMLModel」用來暫存 DAML 檔的文件結構與資訊實體(instance)，之後使用 read()函式將 DAML 的文件結構檔(travel-daml.daml) 與資訊實體 (instance) 暫存於「DAMLModel」中。model.listDAMLClasses()可協助取得 DAML 檔內所有的類別後，再利用 getDefinedProperties()取得個別類別內所擁有的屬性，皆暫存於「暫存器-itp」中。

接著將 DAML 檔裡某一標籤屬性結構及其值儲存於「屬性存取器 (PropertyAccessor)-pa」，利用 getDAMLValue() 函式，將標籤值暫存於 DAMLDateInstance 中，再透過 getValue() 即可取得該標籤屬性內所包含的屬性值，最後則透過 SQL 語法將該屬性值儲存於相對地資料欄位中。圖 15 爲圖 13 資料轉換後之對應資料庫的內容。

4.2 線上資訊實體檢索系統

線上資訊實體檢索系統是供使用者操作的介面部份，可分爲兩部份：「樹狀結構查詢」與「一般查詢」。圖 16 上半部畫面呈現的是「樹狀結構查詢」的使用者介面；資訊檢索畫面的內容是按照旅遊領域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，由樹狀架構的第一層依序到第四層選擇所需要查詢的關鍵字詞，以作爲輸入。各層的資料選項分爲兩部份：該類別的屬性選項與次層的類別選項。在系統執行時，首先將只呈現第一層的查詢選項，然後按照使用者輸入檢索條件的過程，系統將會逐層顯示該層的屬性，以方便使用者輸入或選取查詢的條件。使用者若是點選項目，則系統直接取得對應之階層編號及欄位屬性，可逕自「都會旅遊資訊實體庫」內進行資訊檢索的工作。



```
<daml_oil:DatatypeProperty rdf:ID="半票">
  <daml_oil:domain rdf:resource="#門票資訊"/>
  <daml_oil:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#integer"/>
</daml_oil:DatatypeProperty>

<daml_oil:DatatypeProperty rdf:ID="停靠站">
  <daml_oil:domain rdf:resource="#景點資訊"/>
  <daml_oil:range rdf:resource="http://www.w3.org/2000/10/XMLSchema#string"/>
</daml_oil:DatatypeProperty>

<landscape:門票資訊 rdf:ID="landscape0616_00098">
  <landscape:兒童優待票>0</landscape:兒童優待票>
  <landscape:全票>60</landscape:全票>
  <landscape:半票>30</landscape:半票>
  <landscape:學生票>30</landscape:學生票>
</landscape:門票資訊>

<landscape:景點資訊 rdf:ID="landscape0616_00097">
  <landscape:交通方式>公車</landscape:交通方式>
  <landscape:停留時間>180</landscape:停留時間>
  <landscape:停靠站>木柵動物園</landscape:停靠站>
  <landscape:公車路號>611</landscape:公車路號>
  <landscape:地址>台北市文山區新光路二段30號</landscape:地址>
  <landscape:地理位置(X)>12158243</landscape:地理位置(X)>
  <landscape:地理位置(Y)>24999287</landscape:地理位置(Y)>
  <landscape:景點名稱>木柵動物園</landscape:景點名稱>
  <landscape:種類名稱>自然風景</landscape:種類名稱>
  <landscape:附註說明>台北市立動物園創設於1915年，舊址圓山。</landscape:附註說明>
  <landscape:電話>(02)2938-2300</landscape:電話>
  <landscape:門票 rdf:resource="#landscape0616_00098"/>
</landscape:景點資訊>

<daml_oil:Ontology rdf:ID="">
```

圖12　台北木柵動物園(景點資訊類)之部份資訊實體資料(DAML格式)

```
<vCard:Travel>
  <vCard:trainNumber><rxsd:string><rdf:value>101</rdf:value></rxsd:string>
  </vCard:trainNumber>
      <vCard:startingStation><rxsd:string><rdf:value>Kee Lung</rdf:value></rxsd:string>
      </vCard:startingStation>
      <vCard:endingStation><rxsd:string><rdf:value>Ping Tung</rdf:value></rxsd:string>
      </vCard:endingStation>
    </vCard:Travel>
    <vCard:Travel>
  <vCard:trainNumber><rxsd:string><rdf:value>1003</rdf:value></rxsd:string>
      </vCard:trainNumber>
      <vCard:startingStation><rxsd:string><rdf:value>Sung Shan</rdf:value></rxsd:string>
      </vCard:startingStation>
      <vCard:endingStation><rxsd:string><rdf:value>Kaohsiung</rdf:value></rxsd:string>
      </vCard:endingStation>
    </vCard:Travel>
    <vCard:Travel>
  <vCard:trainNumber><rxsd:string><rdf:value>1009</rdf:value></rxsd:string>
      </vCard:trainNumber>
      <vCard:startingStation><rxsd:string><rdf:value>Kee Lung</rdf:value></rxsd:string>
      </vCard:startingStation>
      <vCard:endingStation><rxsd:string><rdf:value>Kaohsiung</rdf:value></rxsd:string>
      </vCard:endingStation>
```

圖 13　「交通工具資訊」之資訊實體資料(DAML 格式)

```
DAMLModel model = new DAMLModelImpl();
model.read(new FileReader("travel-daml.daml"), vcardBaseURI);
model.read(new FileReader("travel_instance.daml"), instanceBaseURI);
Iterator it = model.listDAMLClasses();
while (it.hasNext()) {
        DAMLClass c = (DAMLClass)it.next();

        Iterator itp = c.getDefinedProperties();

        while (itp.hasNext()) {//取得個別屬性的值
    DAMLProperty fnProp = (DAMLProperty)itp.next();
          // Find the travel instances
    Iterator i = model.listDAMLInstances();
    DAMLInstance travel;
    PropertyAccessor pa;
    DAMLDataInstance fullname;
    while (i.hasNext()) {
          Object o = i.next();
          if ( o instanceof DAMLDataInstance ) continue;
    travel = (DAMLInstance)o;
    pa = travel.accessProperty(fnProp);
    fullname = (DAMLDataInstance)pa.getDAMLValue();
          if ( fullname != null ){
        try{
          stmt = con.createStatement();
          if ( col == 1 ) {
            tmp_trainNumber.addElement(fullname.getValue());
            SQL = "insert into travel (trainNumber) values ('"+fullname.getValue()+"')";
            System.out.println("已新增trainNumber至travel中："+fullname.getValue()+"");}
          if ( col == 2 ) {
            SQL = "update travel set startingStation='"+fullname.getValue()+"' where
                  trainNumber='"+tmp_trainNumber.elementAt(count)+"'";
            System.out.println("已新增startingStation至travel中：
            "+fullname.getValue()+"");}
          if ( col == 3 ) {
            SQL = "update travel set endingStation='"+fullname.getValue()+"' where
                  trainNumber='"+tmp_trainNumber.elementAt(count)+"'";
            System.out.println("已新增endingStation至travel中："+fullname.getValue()+"");
                  }
          stmt.executeUpdate(SQL);
          }catch(Exception e){
                System.out.println(e+"資料新增時發生錯誤");}
        }
            count++;
    }//while (i.hasNext())
        count=0;
        col++;
  }//while (itp.hasNext())
}//while (it.hasNext())
```

圖 14　DAML 檔與資料庫格式轉換的部份程式碼

	trainNumber	startingStation	endingStation
▶	1003	Sung Shan	Kaohsiung
	101	Kee Lung	Ping Tung
	1009	Kee Lung	Kaohsiung
*			

圖 15　資料表格實際轉換的內容

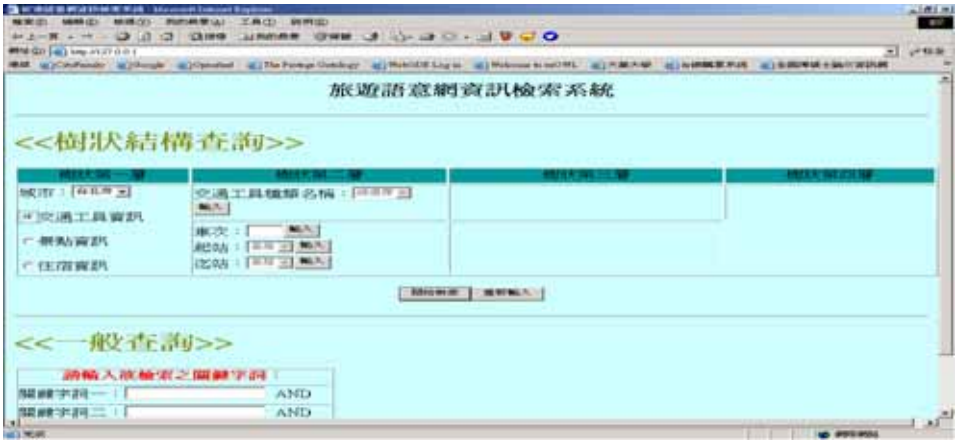


圖 16　資訊實體檢索系統之查詢介面

假設使用者在樹狀第一層的條件，選擇「城市=>台北市」、「領域別：交通工具資訊」，樹狀第二層時，則選擇「起站：基隆」、「迄站：高雄」，其餘屬性皆不指定；系統即可以得知相關關鍵字詞的階層編號：「交通工具資訊」=>「1100」，「起站」=>「1100」，及「迄站」=>「1100」。至於樹狀第三層，因樹狀第二層的「交通工具種類名稱」並未指定屬性值，所以樹狀第三層的屬性選項並未展開，所以預設值爲不指定。最後按照上述的階層編號，分別至相對應的資料表及其欄位，進行關鍵字詞比對，進而將比對結果呈現給使用者。

圖 17 呈現資訊檢索的結果，取得的資料筆數爲 4 筆。可以看到符合使用者的資訊實體之相關標籤及屬性值，以及相關最下層標籤以下未展開的資料，都會一起呈現給使用者。例如：第三層票價資訊與到站時間，使用者並未給予任何檢索的條件，但系統會依照前兩層使用者輸入的條件作爲篩選的條件，呈現所有符合條件的資訊給使用者。

圖 16 下半部畫面所呈現的是「一般查詢」的介面。透過此介面使用者可以自行輸入欲查詢的關鍵字詞(最多三組)，以進行相關的查詢。在此一模式下，當系統接收到使用者輸入的關鍵字詞後，會嘗試至『旅遊詞彙庫』搜尋該關鍵字詞位於『階層編號表』內階層編號的起始與結束編號。依序比對「標籤類別」與「資訊實體類別」的詞彙，若符合，則可以取得相對應的階層編號範圍，根據此起始與結束編號到「階層編號表」檢索詞彙的所有階層編號及欄位屬性，最後按照可能之階層編號到「都會旅遊資訊實體庫」內進行資訊檢索。若比對未發現符合的詞彙，則系統退化成全文檢索的方式，轉而檢索資料庫的各個資料表內的「備註欄位」，以取得使用者需要的資訊。

在一般查詢的部份，將使用一個例子來說明應用服務式資訊實體樹狀結構，和 Google 搜尋引擎的資訊檢索方式的殊異處，並明確指出本系統所應用之樹狀結構，對於資訊檢索的益處。

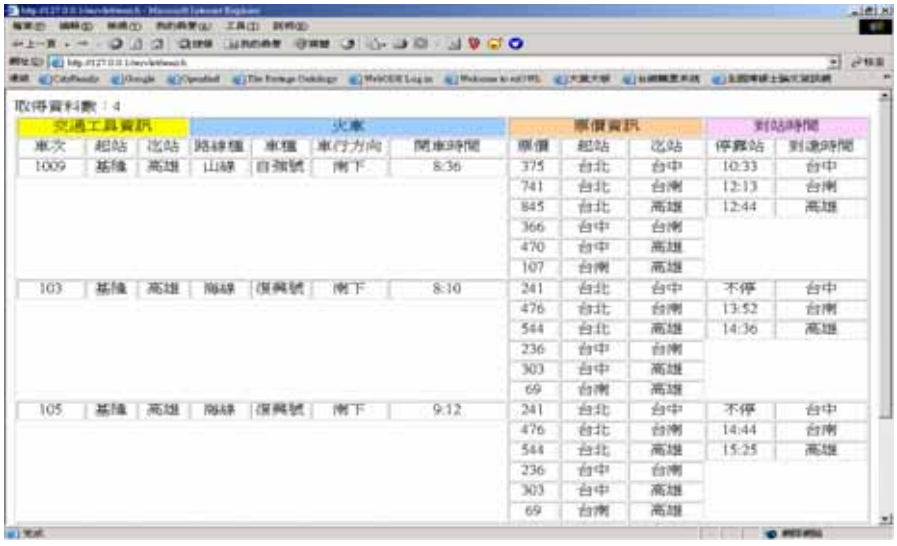


圖 17　資訊實體檢索系統之樹狀結構查詢檢索結果

查詢條件=>關鍵字詞一：北投春天酒店

關鍵字詞二：假日價格

經在兩種檢索系統進行此一資訊檢索動作後，不難發現在 Google 全文檢索搜尋引擎中，使用者雖只輸入兩組關鍵字詞，但是搜尋引擎爲了不遺漏任何可能的資訊，會自動將關鍵字詞，再做一次斷詞的動作。因此，只要符合「北投」、「春天酒店」、「酒店」、「假日價格」、「假日」、「價格」的網頁內容，都包含在檢索結果；而這些檢索結果對使用者而言，卻可能包含有意義與無意義的資訊。例如：有意義的資訊，包含「北投春天酒店，假日 9 折」、「假日入住每房須補 1600 元」…等；無意義的資訊，包含「北投捷運站-酒店定時接送」、「北投春天酒店-露天風呂泡湯卷 2 張」、「春天酒店爲五星級溫泉旅館」…等。使用者在面對混雜的資訊檢索結果，仍然需要以人工主觀的判斷，真正符合本身需求的資訊內容。

而旅遊語意網內的資訊檢索方式，透過資訊實體樹狀結構的輔助，將特定領域的關鍵字詞作了區隔，確保關鍵字詞之間不會互相干擾，也促使系統在進行資訊檢索時，可縮小檢索的範圍，以提高檢索結果的精確性。在本例中，針對「北投春天酒店」、「假日價格」二組關鍵字詞，進行檢索。在資訊實體的樹狀結構中，「北投春天酒店」擁有兩組階層編號：「1200」代表「北投春天酒店」屬於「景點資訊」內的資訊，「1300」代表「北投春天酒店」屬於「住宿資訊」內的資訊；所以利用階層編號的識別，系統得知「北投春天酒店」是兼屬於標籤「景點名稱」與「住宿名稱」的屬性值，而「住宿名稱」又是「假日價格」的上層節點。因此，可以得知使用者真正關心的是「住宿資訊」內「北投春天酒店的假日價格」下的各類房間的假日價格，而不是「景點資訊」或單純的關鍵字詞「北投春天酒店」、「假日價格」。本資訊實體檢索系統檢索結果如圖 18 所示，而

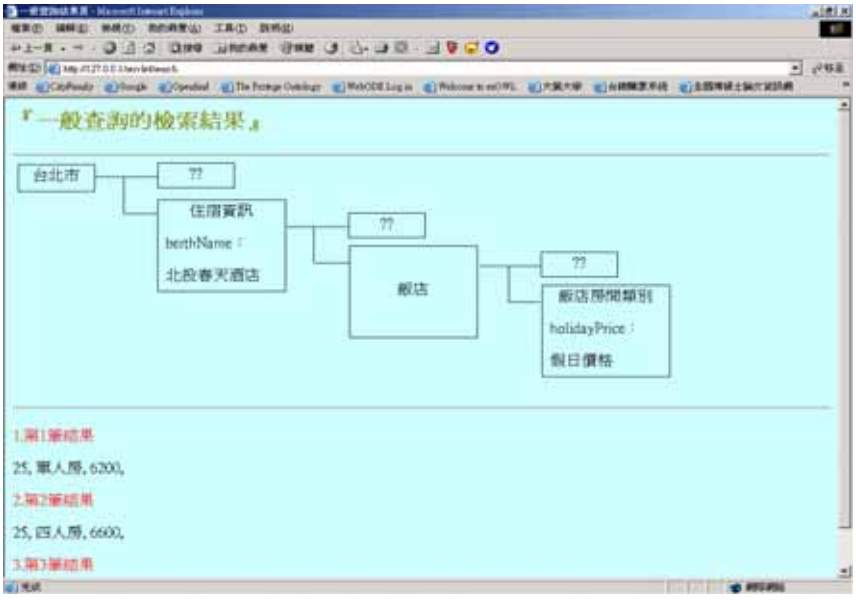


圖 18　資訊實體檢索系統之一般查詢檢索結果

Google 搜尋結果 (民國 93 年 10 月 20 日) 特性則簡述如表 2。

4.3 旅遊排程暨服務預約案例

案例：「一日遊」行程

假設有2位已成年的旅遊者，希望安排在2004年7月5日由台中市出發，目的地爲台北市的一日遊行程，且出發時間是早上8:00，搭乘的火車爲莒光號，而遊玩的景點類型及偏好分別爲古蹟

表 2　Google 查詢之檢索結果簡述

Google進階檢索方式	符合項目個數	第一個網頁	第一個網頁特性	檢索結果
包含**全部**的字詞	2570	花旗卡友特惠案	已過期	前100項未包含春天酒店本身網頁
包含**完整的字句**： "北投春天酒店假日價格"	0			
包含**完整的字句**： "北投春天酒店" "假日價格"	5	台灣溫泉祭–神遊網	網址已失效	未包含春天酒店本身網頁
包含**任何一個**字詞： 北投春天酒店OR假日價格	724	春天酒店Spring City Resort	春天酒店首頁	進入線上訂房系統後，再自行操作找出特定房間價格

寺廟佔所有行程景點數目比例的65%、文化藝術佔所有行程景點數目比例的35%。此外，預算控制在4,000元內，希望能安排「緊湊」10小時的行程，整個旅遊需求如圖19所示。本研究所謂的「緊湊」行程，意指安排景點遊玩皆僅停留「最短遊玩時間」；相對地，若使用者選「適中」或「鬆弛」的行程，在排程考慮時，將會分別安排景點遊玩的時間爲「最短遊玩時間」的1.3及1.6倍。

依據上述使用者的旅遊需求，交通需求是搭乘火車，且景點類型的需求爲古蹟寺廟與文化藝術類型的景點，其偏好程度分別爲65%和35%。當「旅遊服務排程模組」接收到使用者所下達的旅遊需求，並參考了台灣地區城市旅遊知識本體之樹狀結構後，可將其轉換成下列三組查詢：

查詢一：<城市>“台北市”，<交通工具種類>“火車”

查詢二：<城市>“台北市”，<種類名稱 >“古蹟寺廟”

查詢三：<城市>“台北市”，<種類名稱>“文化藝術”

其中‘，’表示“且（AND）”運算。

經向「旅遊服務註冊中心」的旅遊資訊實體資料庫提出三組資訊檢索的請求，以確定是否具有符合條件的結果。查詢結果包括了排程所需的相關資訊，如查詢一所得結果爲「車站名稱」爲「台北火車站」，「X座標」及「Y座標」分別爲「12151757」及「25047188」。至於查詢二及三結果分別如表3與表4所示。

圖19　一日遊的旅遊需求

表 3 一日遊之資訊檢索對應查詢二之旅遊資訊實體庫資料

景點名稱	X座標	Y座標	城市代碼	最短停留時間	景點型態	門票價格	URL
士林官邸	12153012	25094593	1	30	1	EntranceFee0	No
孔廟	12151620	25072445	1	30	1	EntranceFee0	No
四十四坎	12151611	25066857	1	30	1	EntranceFee0	No
陳維英老師府	12151105	25071923	1	30	1	EntranceFee0	No
艋舺教會	12150353	25039515	1	30	1	EntranceFee0	No
龍山寺	12150015	25036840	1	30	1	EntranceFee0	No
溫泉博物館	12150601	25136492	1	60	1	EntranceFee0	No
仙跡巖	12154484	24996524	1	90	1	EntranceFee0	No

表 4 一日遊之資訊檢索對應查詢三之旅遊資訊實體庫資料

景點名稱	X座標	Y座標	城市代碼	最短停留時間	景點型態	門票價格	URL
紅樓電影博物館	12150680	25042186	1	30	2	EntranceFee0	No
布政使司文物館	12150766	25027433	1	30	2	EntranceFee0	No
國立歷史博物館	12150766	25027433	1	60	2	EntranceFee0	No
中正紀念堂	12151759	25037057	1	50	2	EntranceFee0	No
台北市美術館	12152491	25072461	1	60	2	EntranceFee1	No
國父紀念館	12156014	25040035	1	60	2	EntranceFee0	No
國父史蹟館	12152001	25047629	1	30	2	EntranceFee0	No
天文科學教育館	12151729	25096959	1	90	2	EntranceFee0	No
台北海洋生活館	12152329	25088497	1	120	2	EntranceFee2	No
故宮博物館	12154858	25101666	1	60	2	EntranceFee3	No

其中，X座標與Y座標的部份，係採用Papago電子地圖（研勤科技等，民93）所使用之景點地理座標位置，本研究並將以其作爲評估景點與景點間距離的依據。城市代碼是城市於系統中的代號，如台北的代碼爲1、台中的代碼爲2、台南的代碼爲3，及高雄的代碼爲4等。最短停留時間係指景點遊玩時需要的最短遊玩時間，時間單位以十分鐘爲基本單位，而本系統中所定義之最短遊玩時間係依據（大台中觀光網，民93）來定義。

景點型態係依據景點特色區分爲七大類：1代表的是古蹟寺廟、2代表的是文化藝術、3代表的是溫泉泡湯、4代表的是自然風景、5代表的是遊樂園區、6代表的是百貨公司，及7代表的是主題街道類的景點。URL則是定義某一景點若有提供網路訂票時，所需的相關資訊，包括了超連結和網路訂票的參數等，若景點沒有提供網路訂票的服務時，則以No來表示該景點未提供網路訂

票服務。此外，由於景點的門票費率並沒有統一，有些景點可能是免費，而有些景點則是依據遊客的種類而有相對應的門票費率，爲了使本系統能準確的計算出成本花費，因此將景點門票進行分類，其分類如表5所示。其中，EntranceFee0代表參觀該景點是免費的，EntranceFee1至EntranceFee5則依不同遊客種類而有其相對應的門票價格；此外，在本系統裡也使用到EntranceFee6、EntranceFee7與EntranceFee8的門票價格。而本系統所使用的門票價格是依據各景點所公告的門票價格來定義。

當系統接收到使用者輸入的旅遊需求後，接下來會交由「旅遊服務排程模組」進行旅遊行程安排的工作。「旅遊服務排程模組」主要是依據使用者需求條件、各景點的門票費用與最短停留時間來計算權重後，再利用檢索功能取得台北市的部份旅遊景點資訊與鐵路交通相關資訊（如表3與表4），來搜尋出最多可遊玩的景點數目並計算其近似最短路徑，過程訊息如圖20所示。在進行刪點的動作之後，最後搜尋出符合本次條件之最大遊玩的景點數目爲9個。

如圖21所示，當「旅遊排程規劃器」找出一條可行的景點近似最短路徑後，接下來第1行，系統將欲排列的景點放入陣列中；透過其餘各行顯示得知，系統經由排列組合，考量到路徑成本與時間的因素，最終判斷找出一條最短路徑，同時也在允許的計算時間內產生了100個不同的方案。

最後系統取出前三個成本最低的方案，如圖22所示；參酌加入排程模組提供的細部景點時間資訊後，透過圖23的方案選擇畫面，交由使用者進行選擇。

當使用者選擇第一個行程方案之後，接下來，系統會依據該方案的需求，啓動「行程服務預約」機制進行行程服務預約之事宜，如圖24顯示了行程服務預約的結果。

至於二日遊在處理時與一日遊行程安排最大的不同點，在於行程安排模組是要一次安排 10 小時*2 的景點行程後，再予以切割成二天的行程。而在處理「三日遊」行程時，與前兩種排程最大的不同處，是在於加入中間都市，並固定安排一日遊（如案例一）加過夜，而目的都市則是安排二日遊的行程。詳細案例說明及排程結果請參照（王治立，民 93）。

表 5　本系統採用之門票價格一覽表

門票代碼	成人	孩童	軍警	學生	老年人
EntranceFee0	0	0	0	0	0
EntranceFee1	30	15	15	15	0
EntranceFee2	480	430	430	430	350
EntranceFee3	0	0	50	50	0
EntranceFee4	40	15	15	15	0
EntranceFee5	60	30	30	30	0
EntranceFee6	20	10	0	0	0
EntranceFee7	450	400	0	0	300
EntranceFee8	500	400	0	0	0

```
01 開始搜尋...
02 搜尋最大可走點數!!
03 依照門票與最短停留時間重新計算權重...
04 依照權重可走景點爲：
05 7，6，5，4，3，2，1，0，16，15，12，11，17，13，10，
06 14，9，8
07 計算最短路徑...：
08 [8,9,14,10,13,17,11,12,15,16,0,1,2,3,4,5,6,7]
09 最短走法爲：
10 7->10->9->5->11->4->8->14->13->2->3->1->12->16->0->15->17->6
11 時間超過可用時間..重新計算...
12 ・・・
13 近似最短路徑爲：7->10->5->13->8->14->12->15->17
14 最大可走點數：9
```

圖 20　一日遊之景點篩選程式執行過程之輸出資訊

```
01 將要排序之地點放入陣列中
02 地點：17->15->12->14->8->13->5->10->7
03 排列組合中...
04 p：1
05 判斷路徑成本與時間…
06 路徑爲：7->10->5->13->8->14->12->15->17
07 路徑超出預算或時間!!
08 ・・・
09 p：1441
10 判斷路徑成本與時間…
10 路徑爲：5->7->10->13->8->14->17->12->15
12 路徑計算成功!!
13 路徑爲：5->7->10->13->8->14->17->12->15
14 p：1442
15 判斷路徑成本與時間…
16 路徑爲：5->7->10->13->14->8->12->15->17
17 路徑計算成功!!
18 路徑爲：5->7->10->13->14->8->12->15->17
19 ・・・
20 共排序1540次　非法路徑有0個　排列成功有100個
```

圖 21　一日遊之旅遊排程規劃器執行過程之輸出資訊

方案一：台中火車站－台北火車站－國父史蹟館－台北市立美術館－士林官邸－孔廟－陳維英老師府－四十四坎－紅樓電影博物館－艋舺教會－中正紀念堂－台北火車站

方案二：台中火車站－台北火車站－國父史蹟館－台北市立美術館－孔廟－士林官邸－四十四坎－紅樓電影博物館－艋舺教會－陳維英老師府－中正紀念堂－台北火車站

方案三：台中火車站－台北火車站－國父史蹟館－紅樓電影博物館－四十四坎－陳維英老師府－艋舺教會－孔廟－士林官邸－台北市立美術館－中正紀念堂－台北火車站

圖 22　一日遊之旅遊排程規劃器輸出之行程方案

圖 23　經由旅遊排程規劃器自動產生之一日遊方案

圖 24　一日遊之火車票服務預約結果

5. 結論及討論

本研究嘗試運用知識本體的資訊共通性，透過旅遊服務整合雛型系統TSWISS及資訊實體檢索系統的建置，來顯示網際網路上服務資源分享、重覆使用的可行性。首先定義暨建構了一個可行之台灣地區城市旅遊知識本體雛型，並提出一套DAML檔與資料庫格式轉換之方式，可自動轉換網路資料成爲有效檢索的旅遊資訊實體庫。使用者透過人性化的介面表達個人需求後，系統可透過階層式的查詢介面，在大幅節省查詢字串解析的情況下，精確地檢索出旅遊資訊實體庫中的相關資訊，在依據旅遊者之時間、預算、個別景點類型偏好、及設定行程遊玩鬆緊的程度等，進行客製化的旅遊排程規劃，並成功與服務預約機制相結合。故本研究主要貢獻爲：

(1) 提出整合式旅遊排程演算法

本研究整合了貪婪法（greedy strategy）理論、排列組合與綜合評估法的觀念，實作出一套新的旅遊排程方法。相較於一般的方法，本研究的旅遊行程安排能夠更準確地符合旅遊者的個別需求，進而達到客製化的效果。

(2) 旅遊詞彙庫階層編號的設計，確切提高了資訊實體的檢索效率

本系統在旅遊詞彙庫的設計方式，是根據詞彙在知識本體樹狀結構中可能出現的位置，來設定詞彙的階層編號，故此編號可能會超過一個以上。此方式雖然會造成詞彙庫的資料量變得龐大，但透過階層編號，系統就只需針對在階層關係中最底層節點，及其所包含的子節點範圍內進行資訊檢索即可。故在增加檢索精確性下，同時也減少資訊檢索過程中所要檢索的資訊數量，自然也就提昇了檢索的效率，縮短了資訊檢索的時間。例如系統實作中的「北投春天酒店」、「假日價格」範例。

從研究過程中，不難發現此類研究若能加入代理人的運作機制，將使得旅遊語意網整體服務系統的運作，尤其是在旅遊行程安排暨服務規劃上，更能順利達成自動化、客製化服務的目的。此外，和旅遊業者合作，建立更完整的旅遊領域本體庫，以利進行更細部的旅遊行程規劃，如考量到食的問題、和景點開放時間限制等。另外，由於本研究目前僅考慮以鐵路爲主的旅遊行程規劃，雖然在本系統中也使用了公車資訊，但僅以計算出概略成本爲主要考量；若是能夠提供更充份的公車、計程車與汽機車出租等各種交通工具的資訊以供選擇，而不僅限於鐵路交通而已，將使得旅遊者在旅遊安排及實際旅遊過程更爲便利。最後，爲了讓資訊檢索的方式更趨向語意式，在知識本體中同義詞的定義，以及推論規則的建立與處理，將可以更加提昇資訊實體的檢索效能。

參考文獻

大台中觀光網，http://travel.tccg.gov.tw/index1.asp，民國 93 年。

王治立，「旅遊語意網整體服務系統之建置」，大葉大學資訊管理學系研究所碩士論文，民國93年。

中華民國交通部觀光局，「民國91年國人旅遊狀況調查」，http://202.39.225.136/statistics/File/200212/91domestic.htm，民國91年。

吳俸昌，「在時間限制下的個人化旅遊行程規劃」，國立清華大學資訊工程學系研究所碩士論文，民國 91 年。

科威資訊，「中華民國旅行業經理人網路資料庫及系統建置計畫」，http://www.cowell.com.tw/index.htm，民國 91 年。

研勤科技、崧旭資訊、勤崴科技，http://www.papago.com.tw/index.htm，民國 93 年。

高虹，「語意網：電腦也能看懂」，科學人雜誌，民國 91 年 8 月，47-56 頁。

梁書豪，「旅遊代理人以協商之方式推薦旅遊行程」，國立清華大學資訊工程學系研究所碩士論文，民國 90 年。

連惠英，「智慧型旅遊路線排程系統」，靜宜大學資訊管理學系碩士班碩士論文，民國 91 年。

陳肇男，「旅遊行安排及探勘分析之實作」，雲林科技大學資訊管理系研究所碩士論文，民國 88 年。

葉禮宗，「對話介面代理人－以推薦旅遊行程爲例」，國立清華大學資訊工程學系研究所碩士論文，民國 91 年。

資策會網站，http://www.find.org.tw/0105/news/0105_news_disp.asp? news_id =2888，民國 92 年。

廖鴻圖、廖平，資料結構突破，台北：儒林圖書有限公司，民國 89 年。

Berners-Lee, T., Hendler, J. and Lassila, O., “The Semantic Web,” *Scientific American*, May 2001, pp. 24-30.

Chandrasekaran, B., Josephson, J. R. and Benjanmins V. R., “What Are Ontologies, and Why Do We Need Them?” *IEEE Intelligent Systems*, Vol. 14, Issue 1, 1999 Jan.-Feb., pp. 20-26.

Guarino, N., “Formal Ontology and Information Systems,” In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems*, FOIS’98, Trento, Italy, Amsterdam: ISO Press, 1998, pp. 3-15.

Horowitz, E. and Sahni, S., *Fundamentals of Computer Algorithms*, 3rd ed., Singapore: Computer Science Press, 1989.

Noy, N. F. and McGuinness, D. L., “Ontology Development 101: A Guide to Creating Your First Ontology,” http://protégé.stanford.edu/useit.html, 2004.