

南 華 大 學

資訊管理研究所

碩士論文

基於詮釋網路服務之整合型購物決策資訊架構

An MWS-Based Integrated Shopping Decision

Information Framework



研究生：許慶堂

指導教授：邱英華 博士

中華民國 九十五年 六 月

南 華 大 學

資訊管理研究所

碩士學位論文

基於詮釋網路服務之整合型購物決策資訊架構

研究生： 許慶堂

經考試合格特此證明

口試委員： 邱英華
吳光閔
阮金聲

指導教授： 邱英華

系主任(所長)： 資訊管理學系系主任吳光閔

口試日期：中華民國 95 年 6 月 5 日

誌 謝

辛苦了兩年，終於將這份研究完成了。首先要感謝的當然是我的指導教授邱英華老師，邱老師嚴謹的態度，不管是在日常生活上，或是在研究上，大家都是有目共睹。許多研究上的小細節，過去總是沒有特別注意，但經由他的教誨和指導下，我才知道這些細節的重要性。而在生活上，那些往往會被人忽略的事，老師總是特別叮嚀，讓我在不知不覺中，也養成了嚴謹的生活態度。與老師的互動過程中，我更體會到『一日為師，終生為父』的道理。

其次，我要感謝一直陪伴我的研究夥伴，包括哲輝、家卉、宜旻、峻宇、及明得。這兩年來，我們一起同甘共苦，創造了許多美好的回憶。失意時，我們彼此鼓勵；快樂時，我們共同分享，若沒有他們在背後支持著，也許就沒有我今天的研究成果。

最後要感謝的是班上的所有同學，因為有你們，研究所的生活不僅不枯燥，還常常帶來歡樂。這兩年是一段美好的時光，我永遠忘不了和你們相處的點點滴滴，希望大家即使離別，也要繼續為了自己的理想打拼。

許慶堂 謹識

于 南華大學

民國九十五年六月

基於詮釋網路服務之整合型購物決策資訊架構

學生：許慶堂

指導教授：邱英華

南 華 大 學 資 訊 管 理 學 系 碩 士 班

摘 要

近年來因為網路興起，加快了資訊傳播的速度，顧客經常利用網路上的資訊做為購買商品的參考依據。然而，目前網路購物相關的應用並沒有做一個完整的整合，討論區、虛擬社群、購物網站等各自分散林立，造成顧客在查詢資訊上的不便。雖然也有大型的購物網站將購物、討論、評價等功能整合在一起，但這種整合的範圍只侷限在自己網站本身，而非真正將網路上所有相關的資訊統整在一起。

為了解決上述的 題，我們設計一個以詮釋網路服務(Meta Web Service, MWS)為主要核心，並結合 *DAML-S/UDDI 匹配器 (Matchmaker)* 和 *DAML-S 虛擬機器(Virtual Machine)*之整合型購物決策資訊架構(Integrated Shopping Decision Information Framework, *ISDIF*)。此架構是一個結合網路社群和商品資訊的網路服務架構。它是提供決策資訊的觀點進行開發，讓顧客在購買商品之前，能夠經

由 ISDIF 取得商品的相關資訊，做為購物的參考依據。

網路服務是以 *XML*(Extensible Markup Language,可延伸標記語言)為基礎的技術，不論我們需要的網路服務架設在何種平台，透過網路服務註冊中心，我們可以搜尋到適當的網路服務並連結和使用它，達到跨平台溝通的好處。而 MWS 本身也是一種網路服務，但其重點是集中在**語意網**(Semantic Web)的應用。語意網可以將網上有意義的內容加以結構化，並藉由本體論(Ontology)之建立，更容易取得和分享網路上的相關資源及服務。MWS 的功能類似**智慧型代理人**(Intelligent Agent)，能夠自動發現、自動整合相關網路服務，以滿足使用者的需求。

關鍵字：XML、詮釋網路服務、語意網、本體論、智慧型代理人

An MWS-Based Integrated Shopping Decision Information Framework

Student : Chiang-Tang Shiu

Advisors : Yin-Wah Chiou

Department of Information Management
The M.B.A. Program
Nan-Hua University

ABSTRACT

In recent years, the Web is developing to speed up the dissemination of information. Customers usually make shopping decision by referring to the information from the Web. However, applications like shopping websites, shopping Netnews, and virtual communities on the Web are currently not integrated together very well. It results in inconvenience while customers search the shopping information from the Web. Although some shopping websites combine the functions of shopping, discussion, and evaluation, this kind of integration is only limited to the websites themselves, and it does not integrate all relevant information on the Web.

To solve the above problem, we design a system framework to combine *Meta Web Service (MWS)*, *DAML-S/UDDI Matchmaker*, and *DAML-S Virtual Machine* to build an *Integrated Shopping Decision Information Framework (ISDIF)*, which is an MWS-based framework to include information from shopping websites and virtual communities. Customers can make a shopping decision by referring to the information from ISDIF.

Web Services technology is an XML-based architecture to achieve interoperability and portability. The major focus of MWS is the application of *Semantic Web Services*, which can organize relevant information on the Web. Also, it will be easier to acquire and share the relevant resources and services on the Web by building *Ontology*. MWS

is like an *Intelligent Agent*, it can realize automatic discovery and automatic integration to satisfy the users' demand.

Key Words : XML 、 Meta Web Service 、 Semantic Web 、 Ontology 、
Intelligent Agent

目 錄

書面頁	i
論文口試合格證明	ii
誌謝	iii
中文摘要	iv
英文摘要	vi
目錄	viii
表目錄	ix
圖目錄	x
第一章 緒論.....	1
第一節 研究動機.....	1
第二節 研究目的.....	2
第三節 研究方法.....	3
第二章 網路服務概況.....	5
第一節 統一描述、發現和整合(UDDI)	5
第二節 網路服務描述語言(WSDL).....	8
第三節 簡單物件存取協定(SOAP).....	11
第三章 實現語意網路服務相關技術.....	13
第一節 本體論(Ontology)	13
第二節 語意網與服務標記語言：DAML-S.....	15
第三節 詮釋網路服務(MWS)	17
第四節 DAML-S/UDDI 匹配器	24
第五節 DAML-S 虛擬機器	27
第六節 統一標準用語	29
第四章 網路服務自動化架構.....	31
第一節 服務發佈.....	31
第二節 自動發現.....	33
第三節 自動整合.....	34
第四節 自動調用.....	34
第五章 整合型購物決策資訊架構(ISDIF).....	36
第一節 網路服務資訊內容	36
第二節 以 MWS 為核心之整合型購物決策資訊架構	37
第三節 ISDIF 應用實例	42
第六章 結論與未來展望	47
第一節 結論	47
第二節 未來展望.....	47
參 考 文 獻.....	49

表 目 錄

表 1 案例表	21
表 2 處理後的案例表	21
表 3 合併後的案例表	22
表 4 參數轉換矩陣	41
表 5 ISDIF 整合網路服務分類表	43
表 6 網路服務本體論之 URI 儲存表	45
表 7 網路服務特徵值、對應服務、後續結果儲存表	45

圖 目 錄

圖 1 網路服務架構	6
圖 2 UDDI 資料結構	8
圖 3 WSDL 規範內容	9
圖 4 本體論的 Partition 例子	14
圖 5 DAML-S 整體架構	16
圖 6 網路服務與語意關係	17
圖 7 詮釋網路服務架構	18
圖 8 手機銷售網路服務本體論案例	19
圖 9 多個網路服務整合關係	23
圖 10 DAML-S/UDDI 匹配器	25
圖 11 DAML-S 匹配引擎	27
圖 12 DAML-S 虛擬機器	28
圖 13 網路服務自動化架構	32
圖 14 以 MWS 為核心之整合型購物決策資訊架構	38
圖 15 Motorola 手機查詢服務本體論案例	44

第一章 緒論

在本章，我們探討傳統購物網站所面臨的問題，因而激發我們提出解決問題的方法。我們使用的研究方法是結合詮釋網路服務(Meta Web Service, MWS)、DAML-S/UDDI 匹配器(Matchmaker)、與 DAML-S 虛擬機器(Virtual Machine)；利用這三者所形成的自動化網路服務架構，來建立整合型購物決策資訊架構(ISDIF)。

第一節 研究動機

網際網路尚未普及之前，顧客獲得商品相關資訊有限，主要是透過廣告傳單、電視媒體、口耳相傳、店家人員告知等方式。由於資訊的流通不便，顧客只能從商家得知商品資訊，而缺乏顧客購買後分享的使用心得和相關經驗。因此，顧客在購買商品後，常因為資訊的缺乏而造成許多購物的慘痛經驗。

隨著網際網路的發達，許多在網路上的應用也應運而生，例如虛擬社群、討論區、網路購物、及拍賣等。透過這些網路上的應用，資訊的散佈簡單到只需要幾個動作，就能將資訊傳到世界上的每一個角落。因此，不管顧客身在何地，皆能很輕易的透過網路獲得商品的相關資訊和評論，以做為在購買商品前的參考依據。然而，因為這些資訊並沒有整合，當我們想要了解某商品資訊時，必須透過搜尋方式在整個網際網路中尋找，猶如大海撈針，缺乏效率。為了解決此一問題，

本研究建立一個以詮釋網路服務為基礎之整合型購物決策資訊架構，讓消費者在購買商品前，能夠透過此架構查詢到商品的相關資訊，並利用這些資訊做為購物時的決策參考依據。

第二節 研究目的

網路服務是近年來相當流行的話題，*服務的提供者* (Service Provider) 可以透過 *網路服務註冊中心* (Universal Description Discovery and Integration, UDDI) 將欲提供的服務發佈出去，而 *服務的需求者* (Service Requester) 也能透過註冊中心的搜尋功能，找到自己需要的服務並使用它。除此之外，網路服務架構是使用開放性的標準 (如 HTTP、XML 及 SOAP 等)，做為服務和服務之間的溝通基礎，因此能夠達到跨平台整合的優勢。目前網路服務的發展重點在 **語意網路服務** (Semantic Web Service) [15]，而 Hepp [7] 指出：語意網路服務是語意網 (Semantic Web) 的必要元件。語意網的目標在於能夠讓電腦了解人類的需求，並自動作出相關的回應。XML 語言可以讓使用者自訂標籤，而語意網就是建立在 XML 此種定義客製化標籤 (Customized Tag) 的功能上。語意網能明示資訊的意義，它可讓電腦自動處理及整合網路上可利用的資訊。

本研究以語意網路服務概念為基礎，透過本體論 (Ontology) 的使用，網路服務可以達到自動發現、自動整合、自動調用等目的。本文

的目標是透過自動化的功能，使電腦了解消費者的需求，依照消費者輸入的商品特徵，透過搜尋比對和連結各個相關的網路服務，呈現給顧客產品資訊、產品購買途徑、價錢比較、產品評價、產品討論、與問題解決等資訊。這些服務的整合並非只侷限在某個網站上所提供的資訊，而是所有在網路服務上相關的資訊。顧客利用 MWS 獲得的資訊，是所有網路上提供的相關資訊比較，而非見樹不見林的單一網站資訊。

第三節 研究方法

為了實現本研究目的，我們以**詮釋網路服務**(Meta Web Service , MWS) [9]為主要核心，用於建構一套**整合型購物決策資訊架構**(ISDIF)。因 MWS 在某些方面並未完善，我們因而結合了**DAML-S/UDDI 匹配器**(Matchmaker) 和 **DAML-S 虛擬機器**(Virtual Machine)[16]使 MWS 更能發揮網路服務自動化的目的。DAML-S/UDDI 匹配器主要有下列兩個功能：

- 服務提供者利用 DAML-S 描述服務，並透過匹配器發佈服務到 UDDI。
- 服務需求者利用匹配器找到適合的服務。

然而在搜尋服務上，常會因為不同詞彙描述同一事物，而造成搜

尋的遺漏，例如像『手機』、『行動電話』都是描述同一事物，單靠其中一個詞彙來搜尋，便會漏掉其他使用另一個詞彙的服務。此外，若沒有統一詞彙標準，因為各家服務設定的參數不同，造成服務和服務參數連接的衝突，必須仰賴人工處理，不僅花費許多人力成本，也難以達到自動化。因此我們參考 Rosetta Net 組織制定統一標準用語的方式，來解決此一衝突問題。

最後，由於 DAML-S 描述服務的基礎(Grounding)部份是利用 WSDL 規範，因此在服務的調用和執行上，OWL-S 和 WSDL 必須互相配合。DAML-S 虛擬機器(Virtual Machine)即是一個使用 DAML-S 實現網路服務自動調用的架構。DAML-S 虛擬機器能促使 DAML-S 和 WSDL 兩者互相配合，以達到網路服務自動調用和執行的功能，執行完的結果將傳給使用者。

第二章 網路服務概況

以服務為導向的計算(Service-Oriented Computing)[8]，主要分為三大部份：服務供應者(Provider)，服務需求者(Service Requestor or Consumer)，與註冊機制(Registry)。而網路服務就是一種以服務為導向的架構，它包括三個主要元件：UDDI、WSDL、與 SOAP[3]。下面各節，我們將探討這些元件的基本概念。

第一節 統一描述、發現和整合(UDDI)

在服務導向架構中(SOA)，UDDI(Universal Description, Discovery, and Integration)扮演一個服務仲介者的角色(如圖 1)。服務需求者可以透過它來查詢服務，服務提供者可以透過它來註冊和發佈服務，而其本身也要負責與其他仲介者轉送通報已登錄的資訊。為了制定一套標準機制，使商業實體能夠以標準化的方式建立註冊資訊，並描述及發佈本身提供的服務。因此，UDDI 組織(www.uddi.org)必須制定一套標準規範，目前最新版本規範為 UDDI 3.0.2，它的主要項目包括 UDDI Services and API Sets(服務及應該程式介面集)、UDDI Node(工作節點)、UDDI Registry(註冊機制)、及 UDDI Registry Data(註冊資訊)等。我們依序說明如下：

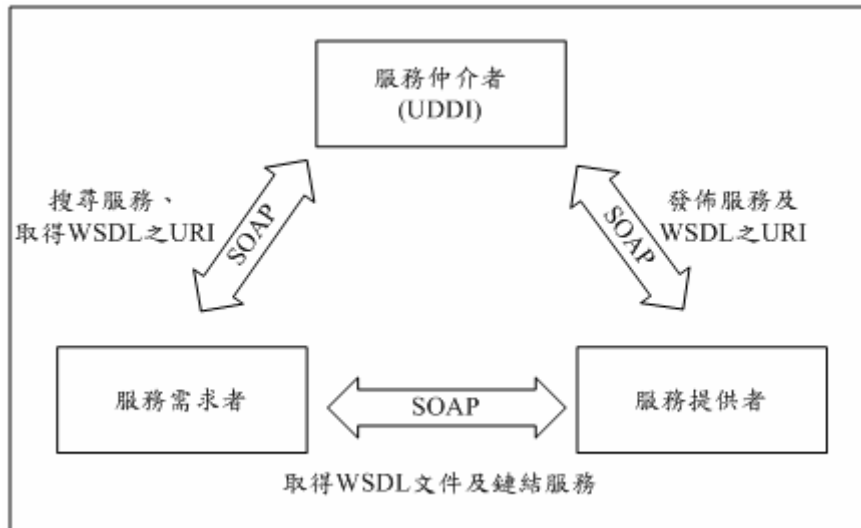


圖 1 網路服務架構

(1) **服務及應用程式介面集(UDDI Services and API Sets)**:此規範用於統一溝通和操作資料的方式,藉由 API Sets 將這些處理方式集合起來並標準化,其中又分為 Node API Sets(工作節點應用程式介面集)和 Client API Sets(客戶端應用程式介面集),提供不同情況下存取 UDDI 資料。

(2) **工作節點(UDDI Node)**:所謂的工作節點,是指至少使用一項 Node API Sets 中的 API,每一個節點都能支援 UDDI 資料的交換,且必須是 UDDI 註冊中心的一員,並能操控 UDDI 註冊中心的資料。

(3) **註冊機制(UDDI Registry)**:所謂的註冊機制,是指一至多個 UDDI 工作節點的集合體,這些工作節點因執行不同的 UDDI API,而有不同的名稱,我們可將註冊機制稱為『UDDI 註冊中心』。註冊中心的類型還包括下列兩種:

- **UDDI 註冊中心**：提供網路服務的各項作業，如註冊、發佈、查詢等。發佈者將服務發佈後，將來要更新和刪除，也都必須透過此註冊中心。
- **相容的 UDDI 註冊中心**：指依照 UDDI 規範所建立的服務仲介者，它們也許只有實作部份的功能而已。若此註冊中心是公用的，它還必須通報其他公用註冊中心新增的資訊。

(4) **註冊資料(UDDI Registry Data)**：此部份用來規範統一的註冊資訊，有些技術文件將這些註冊資訊用『電話簿』的概念來說明，共分為下列三個部份：

- **白皮書(White Page)**：指商業實體的一些基本資訊，如公司名稱、地址、電話、E-mail 等。
- **黃皮書(Yellow Page)**：指依據 UDDI 制定的類別來登錄商業實體，目的是讓需求者能依照類別找出符合本身需求的商業實體。
- **綠皮書(Green Page)**：此部份用來描述服務需求者，如何與服務提供者進行溝通交易的詳細技術資訊，如電子商務交易原則、網路服務技術規格參考資料等。

UDDI 的註冊資訊是使用 XML 格式來制定的，其資料結構共包含六種類型：**商業實體(Business Entity)**、**商業服務(Business Service)**、

服務鏈結(Binding Template)、技術模型(tModel)、商業實體關聯性(Publisher Assertion)、及訂閱(Subscriptions)。由圖 2 可知，一個商業實體可以有很多商業服務，而一個商業服務會因不同的執行方式而有不同的技術鏈結，每一個技術鏈結會對應到其唯一的 tModel(網路服務技術資訊)。

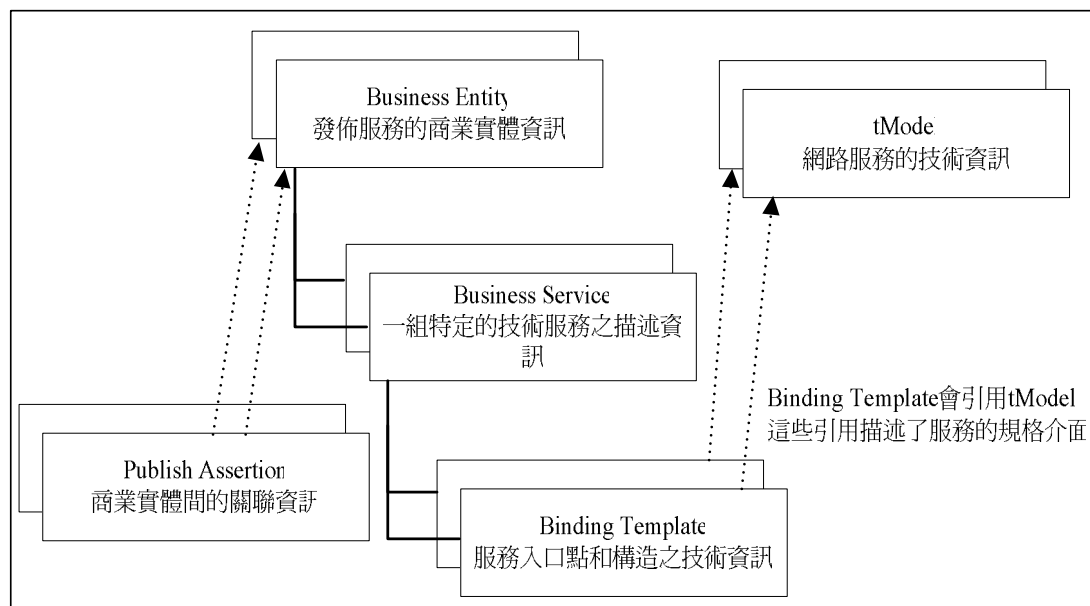


圖 2 UDDI 資料結構

(資料來源：修改自戚玉樑 民 93 年)

第二節 網路服務描述語言(WSDL)

WSDL (Web Service Description Language)主要是用來描述一個網路服務的運作方式，以及指示用戶端與它可能的互動方式。當服務需求者要使用服務時，必須先了解此服務的內涵，以及如何使用，而WSDL 就是用來幫助服務需求者找到符合的服務，並告之如何去使用

服務。目前最常使用的 WSDL 版本是由網際網路標準組織 W3C 所公佈的 WSDL 1.1 版(如圖 3 所示)，其規範可分為兩大部份，分別為**介面描述** (Interface Description) 和 **執行描述** (Implementation Description)。所謂**介面描述**，是用來定義網路服務外觀上的資訊，如資料格式、訊息、作業進行的方式等，其中包括 Types(資料類型集)、Message(訊息)、PortType(介面定義)、Binding(服務鏈結)；而**執行描述**，是用來說明如何執行服務介面所提供的服務，其中包括 Service(實作服務)、Port(服務端點)。我們依序說明如下：

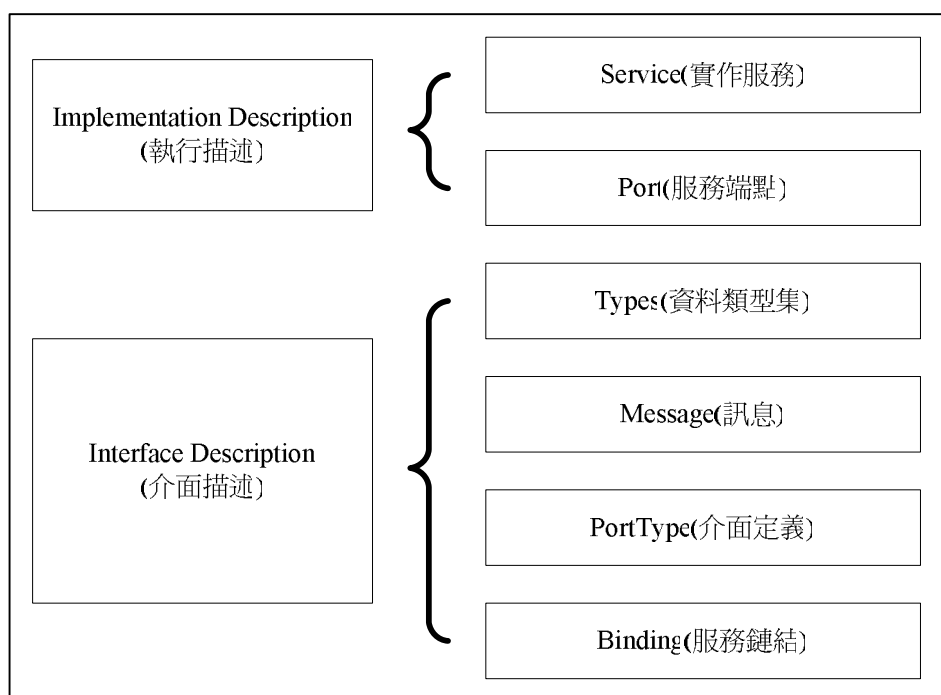


圖 3 WSDL 規範內容
(資料來源：修改自戚玉樑 民 93 年)

- *Types*(資料類型集)：用來定義擴充的資料類型，例如：『產品名稱』、『客戶編號』等，而若是一般可以引用的資料型態，如 int、float，則不必在此重複定義。
- *Message*(訊息)：用來表示網路服務的細節抽象概念，如輸入、輸出、或錯誤等訊息。訊息裡面主要說明那些資訊需要傳輸，使用何種類型來定義訊息的資料結構等。
- *PortType*(介面定義)：用來定義一連串傳送和接收的作業 (Operation)，並對應要傳送的訊息，可以說是整個網路服務運作的縮影。WSDL 1.1 規範了四種傳輸方式，包括 One-way Operation(單向作業)、Request-Response Operation(請求/回應作業)、Solicit-Response Operation(被動回應作業)、Notification Operation(通知作業)，而在實作上通常只支援 One-way Operation、Request-Response Operation 兩種。
- *Binding*(服務鏈結)：用來定義網路服務抽象層面和具體實作層面的細節。此部份承襲了 PortType 和 Message 所提到的資料元素，描述更具體的實作資訊，例如服務的傳輸協定、資料參數型態、及作業等。服務的鏈結協定有三種方式：SOAP Binding、HTTP/Post Binding、和 MIME Binding。此外，本部份還能定義執行錯誤時該如何回應的資訊。

- *Service*(實作服務)：此部份用來具體描述一個網路服務所有作業端點的部署細節，一個網路服務並不侷限只有唯一的作業方式、網址或通訊協定，在同一個服務名稱下，可以用不同 Port 來代表不同的作業節點。
- *Port*(服務端點)：此部份用來描述服務在傳輸資料時的端點位置，每個服務端點包含了唯一的協定和網址，外界是透過此端點來和服務溝通的。

第三節 簡單物件存取協定(SOAP)

SOAP(Simple Object Access Protocol)是在網路上交換結構化和型別資訊的一種簡易通訊協定。SOAP 利用原有的開放性的標準(如 HTTP、及 XML)，做為註冊中心、服務提供者、與服務需求者彼此之間的溝通橋樑，進而達到跨平台的優點。目前 W3C 制定的規範為 SOAP 1.2 版，其內容包含了下列三個部份[3]：

- 信封(SOAP Envelope)：定義一個整體架構，包括訊息的內容、誰要去處理它、它必須強制性的接受或是選擇性的接受。SOAP 訊息本身就是一份 XML 文件，它包括了三個部份：SOAP Envelope(信封)、SOAP Header(表頭)、SOAP Body(內容本體)。
- 編碼規則(SOAP Encoding Style)：定義特定的應用程式之間，在進

行資料交換時，該使用何種資訊類型來溝通。它是有關於資料如何被編碼及序列化的一組定義集。

- 遠端程序呼叫(SOAP RPC)：定義一個用於遠端程序呼叫和回應的協定。SOAP 是利用現有的通訊協定如 HTTP、SMTP、FTP 等來發送訊息，雖然規範中並沒有硬性規定使用任何型式的協定，但卻特別強調了 SOAP-RPC 與 HTTP 的對映。

網路服務技術帶來了許多好處，包括更簡易的應用系統整合、打破不同平台難以溝通的限制、實現以服務為導向的網路時代。然而，目前網路服務技術尚有很多缺點必須克服，包括非自主性、被動性的使用、無法自己整合其他服務、必須撰寫程式執行等。新的網路服務發展重點集中在語意網路服務(Semantic Web Services)，其目的是讓電腦能夠了解網路服務的意義，並能夠自動處理資訊和執行，以達到自動發現、自動整合、自動執行的目標。

第三章 實現語意網路服務相關技術

在本章，我們將介紹用來實現語意網路服務的相關架構和概念，包括：本體論(Ontology)、語意標記語言(DAML-S)、詮釋網路服務(MWS)、DAML-S/UDDI 匹配器、DAML-S 虛擬機器、與統一標準用語。

第一節 本體論(Ontology)

一提到實現語意網路服務，本體論是一個不可或缺的概念。本體論提供了一種明確定義語意的方式，它定義了在現實世界中的『概念(Concept)』，使機器能夠了解資料的意義，並進一步做處理。本體論主要的概念包括下列三個部份[20]：

- *Partition*(分割)：指一群相關的概念，其可以利用某規則來創造和應用，我們稱為 *Partition*。我們以圖 4 來說明，由圖中可知，交通工具可分割為 2 輪及 4 輪兩種，2 輪機械和 4 輪機械為交通工具的子類別(Sub-Class)，這兩個子類別又可延伸出例子(Instance)，其中 2 輪機械的例子為『機車』，而 4 輪機械的例子為『汽車』。
- *Attribute*(屬性)：每一個本體論中的概念(Concept)，可以藉由被分配的 *Attribute* 來描述，*Attribute* 可自行定義，並可建立複雜的本體論關係。比較常見的 *Attribute* 如 Instance Attributes、Class

Attributes、Local Attributes、Global Attributes 等。

- *Relationship*(關聯)：每一個本體論描述的實體中，大部份都會與其它實體有所關聯，比較常用的關聯如 part-of，例如『棒球』是『球類運動』中的一項，即是用 part-of 來描述這兩者的關係。

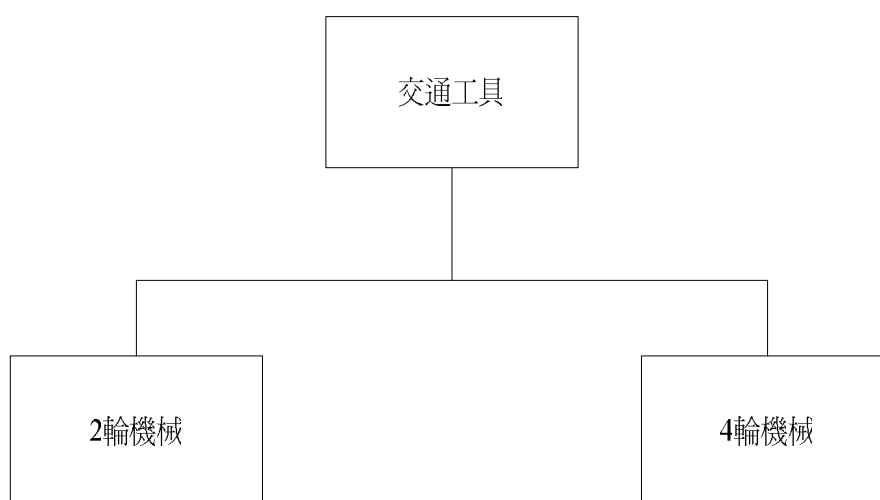


圖 4 本體論的 Partition 例子

本體論必須使用描述本體論的語言來建立，早期較著名的有資源描述架構(Resource Description Framework, RDF)[11,19]，之後演進到 DAML 和 DAML+OIL(Ontology Inference Language)，而目前比較常見的是 OWL(Ontology Web Language)[18]，它是 W3C 所提出的一種本體描述語言。OWL 的前身是 DAML+OIL，並繼承了 DAML+OIL 的框架和大部份的語法和語意特徵，這些都是用來建立本體論的描述語言。而用於建立網路服務本體論的描述語言為 DAML-S[12,13]，目

前公佈的最新版本稱為 OWL-S(Web Ontology Language for Services) [14]；近年來，另一個網路服務本體論描述語言 WSMO(Web Service Modeling Ontology, 網路服務塑模本體論)[4]也正蓬勃發展中，而本研究是使用 DAML-S 來做為網路服務本體論描述語言。在下一節我們將介紹 DAML-S 的重要特性。

第二節 語意網與服務標記語言：DAML-S

使用 DAML-S 描述一個服務，主要可分為三個部份：Service Profile(服務簡介)、Service Model(服務模型)、Service Grounding(服務基礎) [6]。這三者的關係顯示於圖 5，我們依序說明如下：

1. *Service Profile*(服務簡介)：Service Profile 描述服務提供者的資料、聯絡方式；服務的功能、IOPE(Input、Output、Precondition、Effect)；以及服務所屬的分類、服務 QoS 訊息。
2. *Service Model*(服務模型)：Service Model 描述服務的內部流程 (Process)，流程又可分為三類，分別是 Atomic Process、Composite Process、Simple Process。Atomic Process 是不可再分的流程，可以直接被調用。每個 Atomic Process 都必須描述其 Grounding 訊息，告訴大家如何去連結這個流程。Composite Process 是由多個不同的 Atomic Process 或 Composite Process 構成的流程，其中還定義了這

些流程的處理程序，包括了Sequence、Split、Split+Join、Unordered、Choice、If-Then-Else、Iterate、Repeat-Until。Simple Process則是一個抽象的概念，不能被調用，也沒有Grounding訊息。

3. *Service Grounding*(服務基礎)：Service Grounding 描述服務如何被連結使用、連結的協定、訊息格式、端口(Port)等。然而，OWL-S 並沒有定義語法來描述這些內容，而是引用 WSDL 規範。

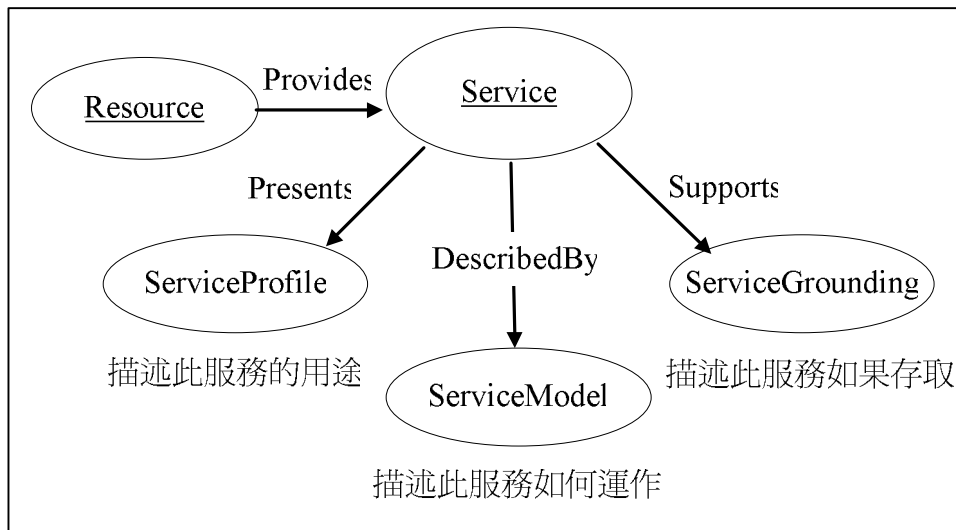


圖 5 DAML-S 整體架構

(資料來源：修改自 DAML Services Coalition 2003)

網路服務透過 DAML-S 描述，可以實現服務自動發現、自動調用、及自動組裝等能力。在下一節，我們將介紹本研究所使用來實現自動發現服務的架構：詮釋網路服務(Meta Web Service, MWS)。

第三節 詮釋網路服務(MWS)

在網路服務中加入語意(Semantic)的概念，是未來網路服務發展的方向，經由語意描述網路服務，電腦能夠直接處理服務的資訊，有助於達到自動發現、自動整合等目的。語意和WSDL並不是取代的關係，而是互補的關係。如圖6所示，語意能夠幫助使用者找到適合的網路服務。然而，在訊息、鏈結、操作方面，依然要仰賴WSDL。

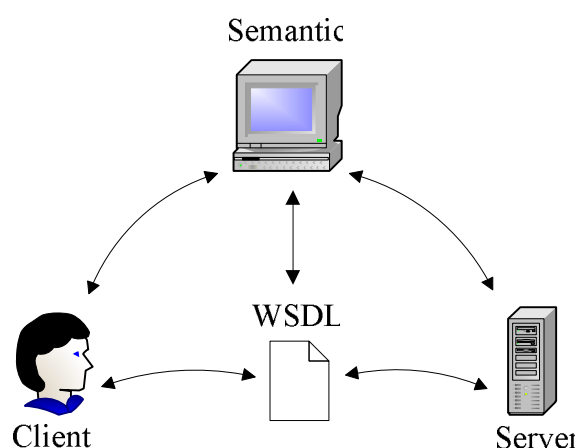


圖 6 網路服務與語意關係

(資料來源：修改自戚玉樑 民 93 年)

Burstein et al. [5] 認為語意網路服務發現和互動的過程，可分為三個階段：服務發現 (Candidate Service Discovery)、服務接合 (Service Engagement)、與服務啟用 (Service Enactment)。其中服務發現階段，可以透過語意配置者、註冊代理人等架構來實現發現服務。本文用來發現和整合服務的機制即是**詮釋網路服務**(Meta Web Service, MWS)，圖 7 顯示出 MWS 的整體架構，它正是一個實現語意

的架構。MWS 結合**決策支援系統**(Decision Support System , DSS)概念，能在接收到使用者發出的問題後，透過決策和分析的過程，找出相關網路服務並自動將之整合，最後產出最佳的結果，傳回給使用者。

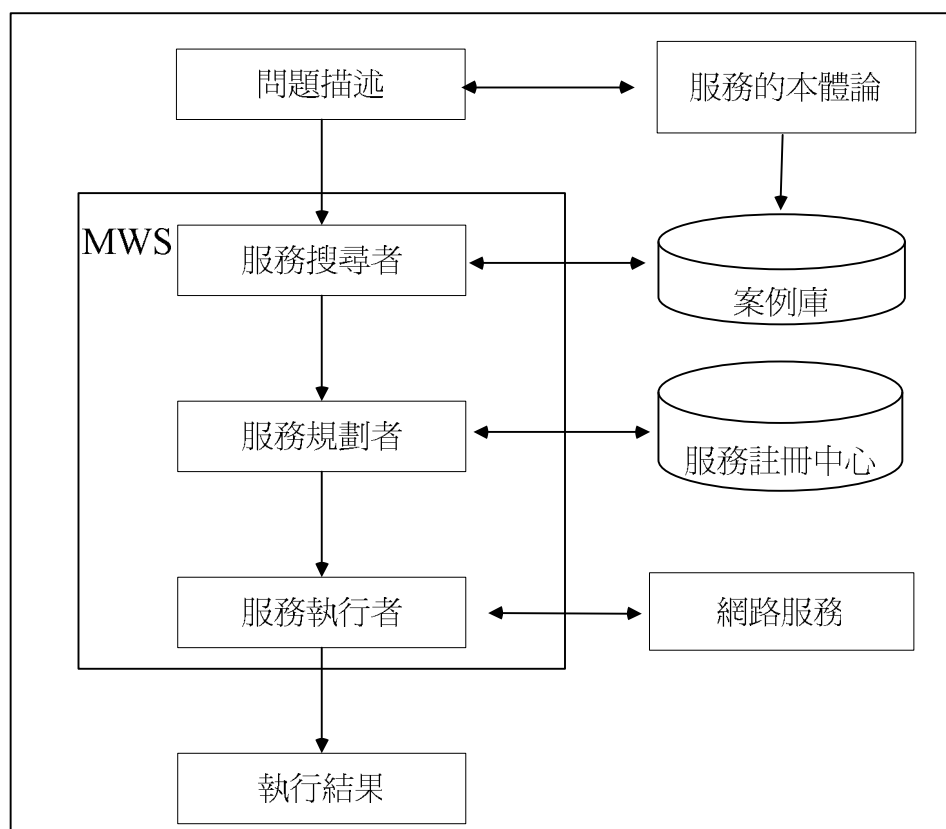


圖 7 詮釋網路服務架構
(資料來源：修改自kwon 2003)

MWS架構的前提是：網路服務在發佈時，必須使用DAML-S描述服務的本體論，再與之一起發佈。因DAML-S和網路服務架構尚未整合，目前的做法是將本體論儲存在對應的URIs上。DAML-S是整個架構的主要關鍵，它是一種語意的網路服務描述語言，為網路服務供應

商提供了一套核心的標記語言集，使之可以一種明確的、電腦能夠解釋執行的方式，來描述網路服務的屬性和功能。因此，透過DAML-S描述，電腦能夠實現自動發現、自動整合等能力。服務的本體論包括了服務提供者的基本資料、參數描述、屬性和功能等。我們以手機銷售網路服務提供者的基本資料本體論為例，其型式如圖8所示。

```
<service : ServiceProfile rdf: ID = "Profile_Mobilephone_sale">
  <profile : serviceName>Mobilephone_sale</profile : serviceName>
  <profile : textDescription>This service provides people who want to
buy the mobile phone we produce</profile : textDescription>
  <profile : provideBy>
    <profile : ServiceProvider rdf: ID = "Michel P.">
      <profile : name>Mobilephone_sale</profile : name>
      <profile : phone>05-2955451</profile : phone>
      <profile : email>u8000115@yahoo.com.tw</profile : email>
      <profile : webURI>http://www.motorola.com.tw/~sale</profile : webURI>
    </profile : ServiceProvider>
  </profile : provideBy>
</service : ServiceProfile>
```

圖 8 手機銷售網路服務本體論案例

MWS的核心是：決策支援系統 (DSS)和案例庫 (Case Base)。

MWS中的*Service Finder*(服務搜尋者)在接收到使用者發出的問題描述後，會先到案例庫中尋找過去的案例，若有相符的案例則採用此案例，而不必到UDDI中搜尋，目的是為了減少到UDDI中搜尋所花費的

時間成本。然而，將案例儲存在案例庫中，在搜尋上會遭遇到功能性相依問題：假設在案例庫中有4個案例(如表1)，每個案例對應了6個特徵值(Feature)，其中Yes代表此特徵值是某案例執行時必需的，而No代表不必需。經過執行後，每個案例都有其輸出的結果。

1. 若今天使用者需求的服務特徵是(Yes, Yes, Yes, Yes, No, No)，我們使用最簡單的配對方式來搜尋案例，最符合的案例是 Case#1，因為 Case#1 的特徵值為(Yes, Yes, No, No, No, No)，總共有 4 個特徵值符合需求($4/6=67\%$)。
2. 然而，若我們更仔細的觀察，會發現 Case#1 和 Case#3 可以互相合併，因為 Case#1 的輸出結果，恰好等於 Case#3 的輸入特徵值 Feature #3。合併後的特徵值為 Case#1&3=(Yes, Yes, Yes, No, No, No)，總共有 5 個特徵值符合使用者需求($5/6=83\%$)。
3. 最後，我們又可發現，合併後的 Case#1&3 可以和 Case#4 合併，因為 Case#1&3 的輸出結果和 Case#4 的輸入特徵值 Feature #4 相同。合併後的值將變成 Case#1&3&4=(Yes, Yes, Yes, Yes, No, No)，剛好完全符合我們的需求($6/6=100\%$)。

表 1 案例表

Cases	Feature#1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6	Result
Case#1	Yes	Yes	No	No	No	No	Feature #3
Case#2	Yes	No	Yes	No	Yes	Yes	Solution#1
Case#3	No	No	Yes	No	No	No	Feature #4
Case#4	No	No	No	Yes	No	No	Solution #2

(資料來源：修改自 kwon 2003)

為了解決上述的功能性相依問題，Kwon 將表 1 改成表 2 的樣式，案例中的特徵值若為 Yes，則改成-1，輸出的結果若為某特徵值，則將此特徵值改為 1。若有兩個案例的同一個特徵值分別為-1 及 1，則可將這兩個案例合併，合併後的結果如表 3 所示。經由這種方式，便能避免漏掉最佳相似度的案例。

表 2 處理後的案例表

Cases	Feature#1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6
Case#1	-1	-1	1	0	0	0
Case#2	-1	0	-1	0	-1	-1
Case#3	0	0	-1	1	0	0
Case#4	0	0	0	-1	0	0

(資料來源：修改自 kwon 2003)

服務搜尋者找到合適的方案後，會通知 *Service Planner* (服務規劃者) 進行下一個步驟，但若在案例庫中並沒有符合使用者需求的方案時，服務規劃者便會到註冊中心去搜尋合適的方案。因為從註冊中心搜尋是一件非常花費時間和成本的事，因此 MWS 架構使用案例庫儲

存過去成功的案例，利用以**案例**為基礎的**案例推理引擎(Case Base Reasoning, CBR)**來找出合適方案，以縮短不必要的時間和成本。

表 3 合併後的案例表

Cases	Feature#1	Feature #2	Feature #3	Feature #4	Feature #5	Feature #6	Result
Case#1	-1	-1	1	0	0	0	Feature #3
Case#2	-1	0	-1	0	-1	-1	Solution#1
Case#3	0	0	-1	1	0	0	Feature #4
Case#4	0	0	0	-1	0	0	Solution #2
Case#1&2	-1	-1	-1	0	-1	-1	Solution#1
Case#1&3	-1	-1	-1	1	0	0	Feature #4
Case#3&4	0	0	-1	-1	0	0	Solution #2
Case#1&3&4	-1	-1	-1	-1	0	0	Solution #2

(資料來源：修改自 kwon 2003)

服務規劃者主要為接收服務搜尋者推薦的方案，並找出每一個網路服務的URI，讀取其基本資訊，最後整合相關的網路服務，產生輸入和輸出參數的轉換矩陣，傳給*Service Executor*(服務執行者)去執行。網路服務整合的關係如圖9所示，我們說明如下：

- 圖中虛線的部份，是每個網路服務的範圍，每個網路服務都有自己的輸入和輸出參數，不同的Var編號代表不同的參數。符號⊙代表某服務需要的輸入參數，符號○代表某服務的輸出參數。
- 某網路服務的輸出參數，若和其他網路服務的輸入參數相同，我們便可利用此參數將兩個網路服務連結在一起，亦即某網路服務

的輸出參數，可為其他網路服務的輸入參數，這類的參數我們用符號●來表示。

- 經過服務與服務間的連結，使用者可以得到他想要的結果。從使用者的角度來看，使用者只是操作一個網路服務來得到結果而已，但從設計者的角度來看，這是經由許多不同網路服務整合而來所得到的結果。

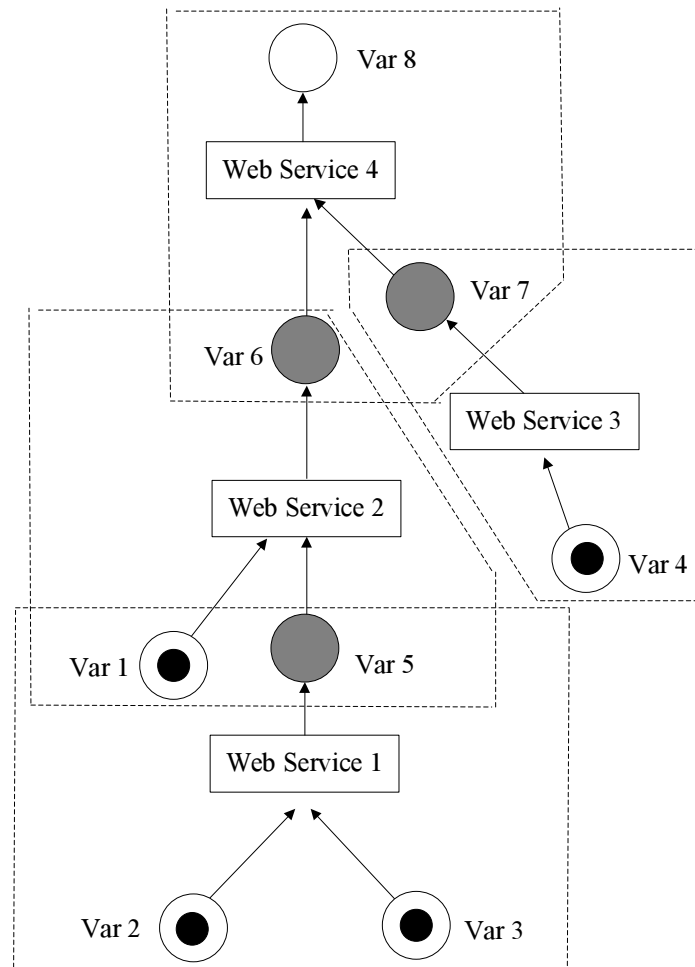


圖 9 多個網路服務整合關係
(資料來源：修改自 kwon 2003)

MWS的自動發現與自動整合等優點，使得**語意網**(Semantic Web)的概念不再只是夢想。然而，MWS在實際運作中，依然有許多未詳細描述的部份，為了補足這些地方，我們結合DAML-S/UDDI Matchmaker 和DAML-S Virtual Machine架構，使整個MWS在實行上能夠更實際。在下列兩節，我們將分別介紹這兩個架構。

第四節 DAML-S/UDDI 匹配器

DAML-S/UDDI 匹配器(Matchmaker)將 DAML-S 與 UDDI 中的 *tModel*(技術模型)結合起來，從而在 UDDI 中增加語意的訊息。所謂的 *tModel*，它描述了一個網路服務技術詳細資訊，是關於網路服務具體鏈結的細節。經由 DAML-S 描述成本體論後，可讓電腦了解服務本身的意義，方便做資訊的處理和應用。然而，DAML-S 卻無法完整描述服務的執行描述、介面描述等。相反的，UDDI 中的 WSDL，定義了詳細的介面描述，稱為 *tModel*，但我們無法利用 WSDL 做到語意的處理。因此，為了結合 DAML-S 與 WSDL 兩種優點，DAML-S/UDDI 匹配器便應運而生。DAML-S/UDDI 匹配器[16]主要可分為三大模組，其架構如圖 10 所示，我們依序說明如下：

- **通訊模組**：所有的通訊請求都必須通過此模組進入匹配器中，包括服務提供者向註冊中心發佈服務的廣告訊息、服務需求者向註冊中

心查詢服務的請求。

- **DAML-S/UDDI 轉換器**：此模組用來處理 DAML-S 文件和 UDDI 格式文件的轉換，服務提供者的 DAML-S 廣告資訊可以被轉換成 WSDL 格式，並向 UDDI 發佈服務。
- **DAML-S 匹配引擎**：此模組為整個匹配器核心，其又可分為四個部份，分別為廣告資料庫、本體論資料庫、匹配引擎、與 DAML+OIL 推理機，其架構如圖 11 所示。廣告資料庫用來儲存服務提供者發佈的廣告資訊；本體論資料庫用來儲存服務的本體論；DAML+OIL 推理機用來做為本體論之間的推理；而匹配引擎用來做為服務和需求之間的配對。

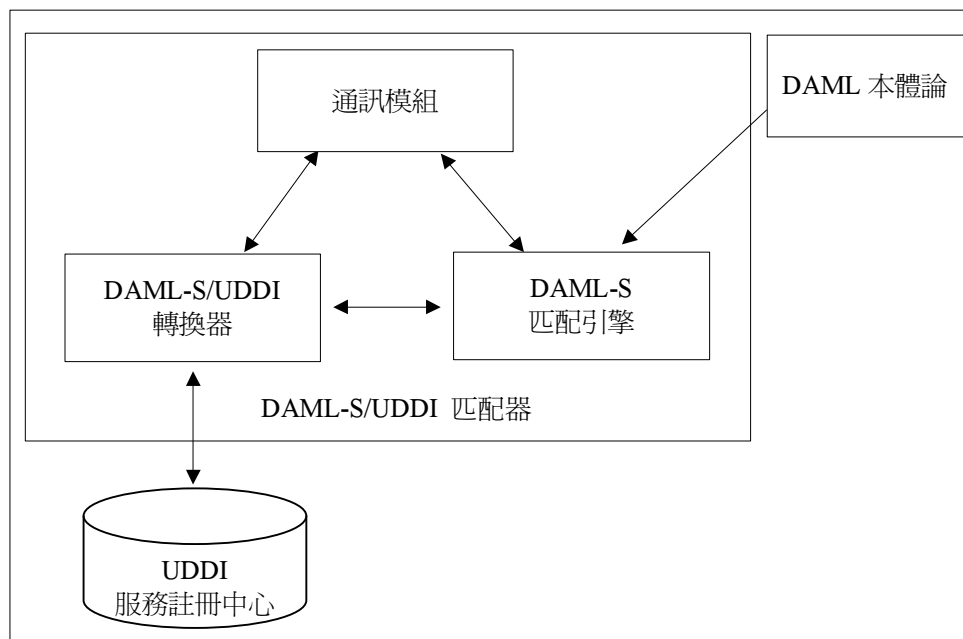


圖 10 DAML-S/UDDI 匹配器

(資料來源：修改自 Sycara et al. 2003)

當一個服務提供者欲發佈服務到註冊中心時，首先必須以 DAML-S 描述整個服務的功能和相關資訊；這些資訊會經由通信模組傳送到 DAML-S/UDDI 轉換器，轉換器利用這些資訊轉換成 UDDI 發佈的格式；最後送到 UDDI 中進行發佈，而原本的服務廣告資訊將會被送到廣告資料庫中儲存。

當一個服務需求者欲查詢服務時，必須先以 DAML-S 描述一個服務的功能需求，此需求會經由通信模組傳送到 DAML-S 匹配引擎中；匹配引擎將需求傳送到 DAML+OIL 推理機，推理機利用本體論資料庫裡的本體論，與使用者發出的需求進行推理配對。我們舉一個例子來說明推理的作用，譬如有個賣 Pet Food(寵物食物)的服務，現在有一個想買 Cat Food(貓的食物)的請求，如果是透過關鍵字的方式做匹配，這個請求並無法和適合的服務做匹配。若在 Pet Food 中的本體論有一條 Cat is a Pet(貓是寵物)的知識，透過推理機的推理，Cat Food 的需求便能夠和 Pet Food 做配對。最後，推理機將推理的結果傳送到匹配引擎做匹配，匹配引擎會從廣告資料庫中找到最適合的服務，並透過此服務名稱到 UDDI 中找到服務並取得 WSDL 文件。

因 MWS 在服務搜尋上並沒有做更詳細的描述，於是我們將 DAML-S/UDDI 匹配器加入 MWS 中的服務搜尋者上。DAML-S/UDDI 匹配器能夠實現語意搜尋的目的，且能跟 MWS 中的

服務搜尋者進行搭配。

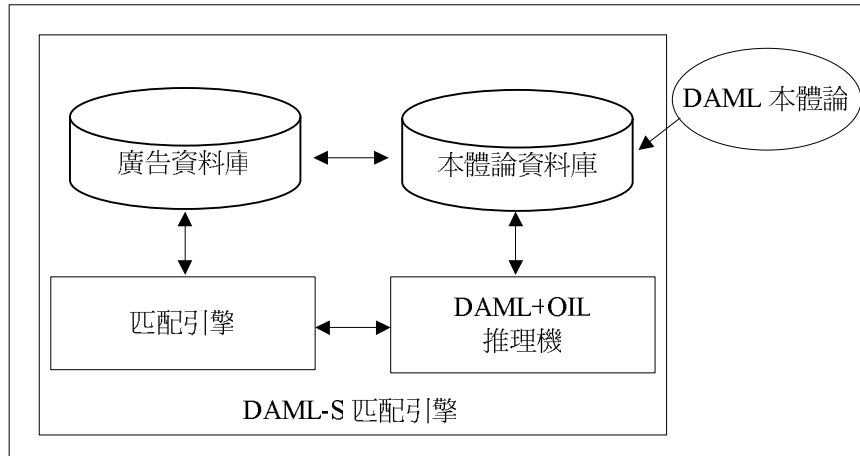


圖 11 DAML-S 匹配引擎

(資料來源：修改自 Sycara et al. 2003)

第五節 DAML-S 虛擬機器

由於 DAML-S 描述服務的 Grounding(基礎)部份是利用 WSDL 規範，因此在服務的調用和執行上，DAML-S 和 WSDL 必須互相配合。DAML-S 虛擬機器(Virtual Machine)[16]即是一個使用 DAML-S 實現網路服務自動調用的架構(如圖 12)，它包括了三大部份：網路服務呼叫(Web Service Invocation)、DAML-S 處理器(Processor)、DAML 推論引擎(Inference Engine)。其中網路服務呼叫從服務取得連結相關資訊後，轉換成 DAML-S 傳給 DAML 推論引擎。DAML 推論引擎負責處理從其他服務取得的本體論和 DAML-S 相關規定，如 DAML-S Process Model、DAML-S Grounding，然後再將這些訊息送給 DAML-S

處理器。而 DAML-S 處理器能透過網路服務呼叫與服務提供者溝通和互動，並使用 DAML-S 和本體論來決定服務如何被執行。

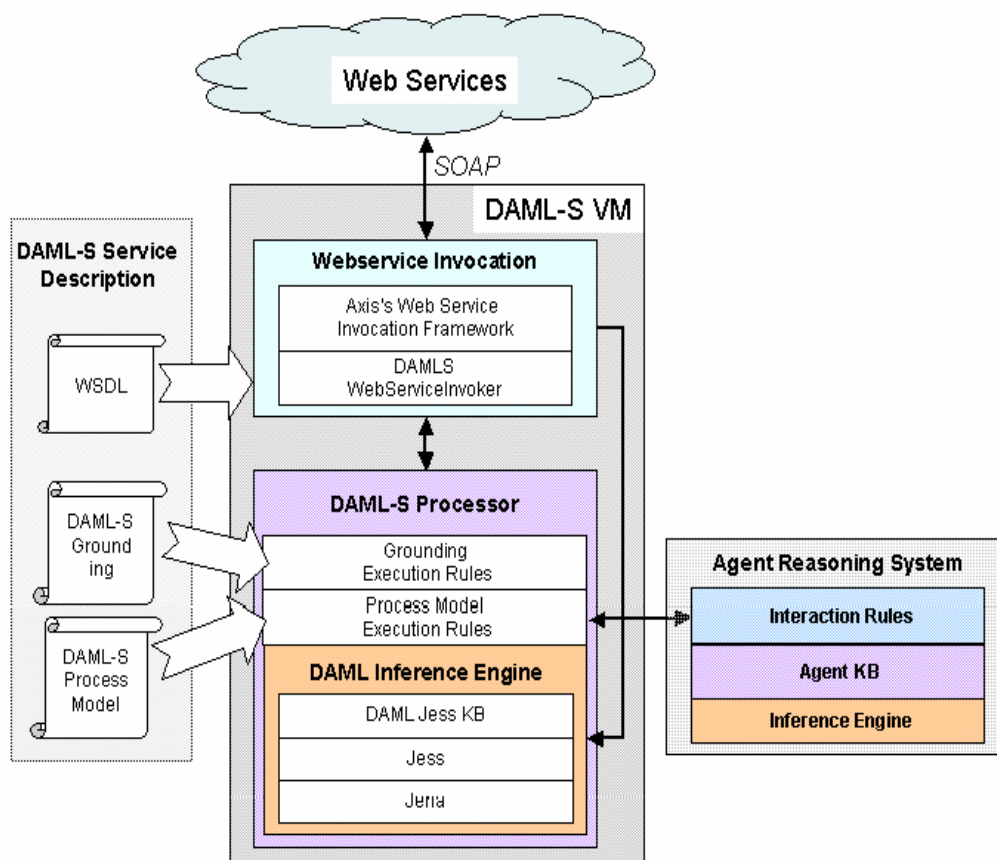


圖 12 DAML-S 虛擬機器

(資料來源： Sycara et al. 2003)

我們將 DAML-S 虛擬機器與 MWS 中的服務執行者結合，因 MWS 在執行服務上，並沒有做詳細的描述，而 DAML-S 虛擬機器能夠實現網路服務自動調用，且使 MWS 在執行服務上更具體。

第六節 統一標準用語

網路服務架構是以 XML 為基礎，雖然 XML 能夠達到跨平台溝通的好處，且能自行定義標籤，但這樣高度的延展性也帶來了新的問題：企業間定義標籤的不一致性。假設有兩個同樣是販賣手機的服務，其中一個定義的類別 (Class) 名稱為 Cell Phone，而另一個定義的類別名稱為 Mobile Phone。以人類的角度來看，我們知道這兩個名詞都是指同一種東西，但電腦卻無法辨別。而另一個問題在於服務宣告的參數名稱上，假設有兩個販賣手機的服務，兩個服務都定義了手機數量這個參數，但其中一個宣告參數的名稱為 num，而另一個宣告的名稱為 number，若我們想將這兩個服務整合，因為變數名稱不同，就必須依賴人工來判斷和連接，而無法達到自動化整合的目的。

為了解決上述的問題，相關的研究很多，其中最常見的是以代理人(Agent)的方式處理這些不一致的定義，例如戚玉樑與關銘[2]提出語意解譯代理人 (Semantic-Aware Agent ,SAA)，其能計算服務間組合的相似程度，讓服務流程設計者能搜尋符合其需求的網路服務並整合；邱宏昇等人[1]提出 XML 轉換技術，利用程式將兩個不同企業的 XML 文件轉換成相同定義的格式，以方便彼此間資料的溝通。而近年來正發展中的另一個網路服務描述語言 WSMO (Web Service Modeling Ontology, 網路服務塑模本體論)[4]則是用調解 (Mediator)

來解決因詞彙或協定不同，造成服務間整合會發生的問題，而我們則是提出統一標準用語的構想來解決此一問題。

關於本文所提出的統一標準用語，我們參考統一標準用語制定最著名的組織 *Rosetta Net*。該組織是由美國最大之供應鏈軟體與服務供應商 Ingram Micro 號召 IBM、HP、Intel、Microsoft 等大廠，在 1998 年籌組的標準組織。此組織為一非營利、獨立運作之協會 (Consortium)，成立宗旨為發展、制定與推動應用全球性之產業供應鏈標準，以解決各家廠商沒有統一標準流程和用語，所造成的效率和成本問題。目前 Rosetta Net 制定標準的範圍為半導體製造、電子元件、及電腦製造等，台灣大多數的半導體業也已開始遵循 Rosetta Net 標準，並與企業內部資訊系統整合 [17]。

我們建議使用本系統的服務提供者，以及未來相關產業的廠商能夠組織共同聯盟，並以 Rosetta Net 為例，制定產業間彼此溝通的共同字彙、流程、及通訊協定等，以解決服務整合會產生的衝突。成立組織制定統一標準用語，除了能夠解決服務間的整合問題，組織內的企业亦能解決彼此溝通的衝突，並加深企業間的合作關係。

第四章 網路服務自動化架構

本研究所提出之系統架構，必須在網路服務自動化的前提下運作。為了實現網路服務自動化，服務提供者在發佈服務時，必須在符合語意服務的架構上發佈。而服務需求者在搜尋和執行上，主要可分自動發現、自動整合、與自動調用。圖 13 呈現出實現網路服務自動化的架構，在下面各節，我們將介紹該架構的主要功能：服務發佈、自動發現、自動整合、與自動調用。

第一節 服務發佈

由於目前 UDDI 並沒有支援 DAML-S 格式的網路服務描述文件，然而，為了實現語意網路及網路服務自動化，DAML-S 與 UDDI 的結合是一項重要的因素。MWS 提出在發佈服務時，將 DAML-S 文件的 URI 一併發佈於 UDDI 中，服務需求者再從 UDDI 中找到服務，並從其 DAML-S 的 URI 取得文件和處理。而 DAML-S/UDDI 匹配器則是透過 DAML-S/UDDI 轉換器自動將發佈服務的訊息轉換成 WSDL 格式，發佈於 UDDI 中，並將本體論儲存於資料庫裡，以便做為配對之用。本文採用後者的方法，因其架構在服務配對上的效率較高，在實際運作上的可能性也較高。

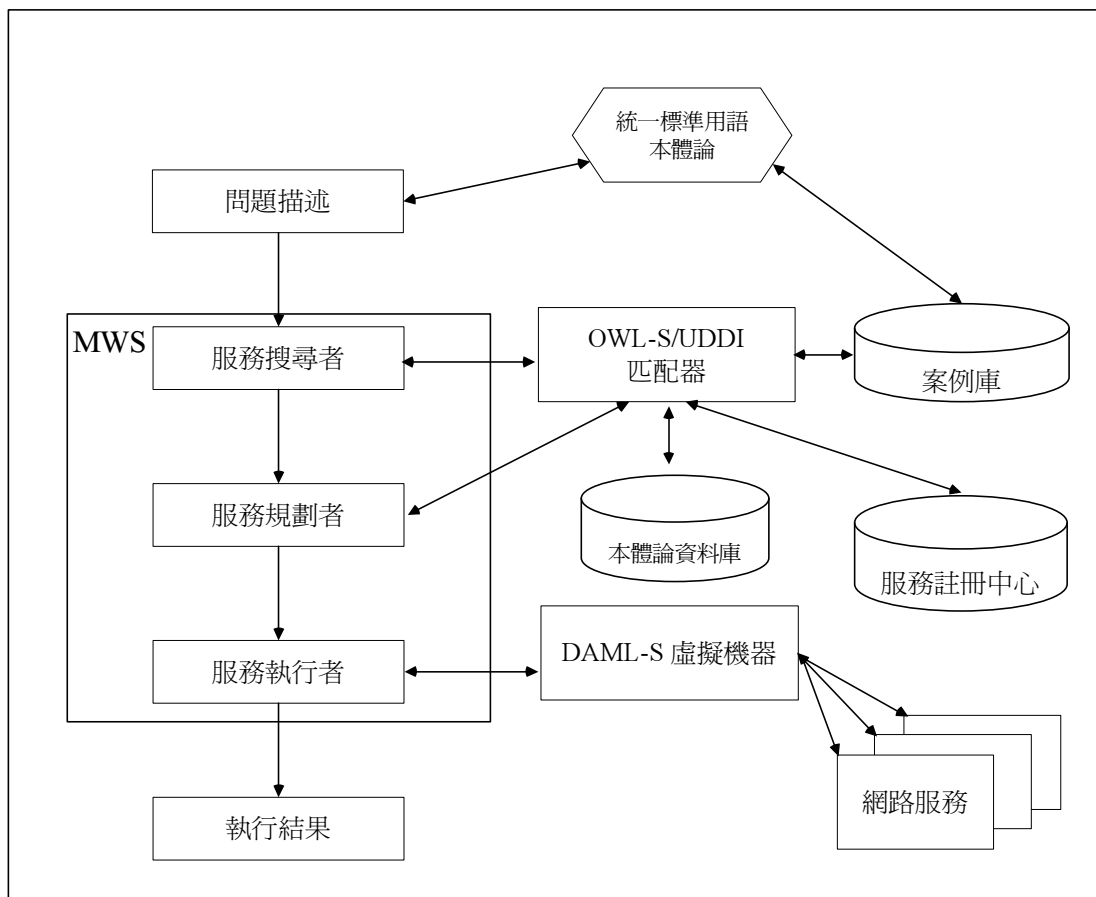


圖 13 網路服務自動化架構

為了實現網路服務自動化，服務提供者必須利用 DAML-S 格式，建立符合 DAML-S/UDDI 匹配器的服務廣告訊息，而此廣告訊息必須採用統一標準用語規範。統一用語標準需要未來相關產業共同制定和協商，型式可參考 Rosetta Net 標準。當服務提供者將廣告訊息建立完後，再經由 DAML-S/UDDI 匹配器向 UDDI 發佈服務，此時，這個服務便符合了實現網路服務自動化的基本要求。

第二節 自動發現

過去網路服務的使用，必須透過程式設計師或對網路服務有相當程度了解的人，到 UDDI 中透過搜尋的方式找尋適合的服務，最後從 UDDI 中得到服務 WSDL 文件的 URI，再利用此 WSDL 文件的規範撰寫程式連結服務並使用它。有時候常會因為詞彙的不同，造成找到的服務並非使用者真正所需求的，對於使用者而言，這是非常被動性且非自主性的。因此，我們希望網路服務能夠達到自動發現的目的，當使用者輸入自己的需求，系統會自動找到最適合的服務且自動連結它。

本架構用來自動發現服務的元件是服務搜尋者，我們保留 MWS 中埋置的案例庫，用來儲存過去成功的案例。然而，因為考慮到案例的時效性，因此我們建議每隔一段時間後，可將這些過時的案例刪除，這也可以節省資料庫的空間。當我們要搜尋一個服務時，會先到案例庫中搜尋，目的是減少在註冊中心搜尋的時間成本。因為 MWS 只有描述搜尋上的演算法，而忽略了使用本體論比對的部份，因此，我們在這個部份加入了 DAML-S/UDDI 匹配器，它能夠儲存服務提供者發佈服務的本體論，藉由這些本體論和使用者的需求比對，若在案例庫中能夠找到符合的案例，本架構便可直接套用案例之前的規劃，但若在案例庫中並沒有符合的案例，服務需求者便會通知服務規

劃者到 DAML-S/UDDI 匹配器去尋求相關的服務，並整合規劃這些服務。

第三節 自動整合

本架構用來自動整合的元件為服務規劃者，它能夠接收服務搜尋者從案例庫中找到的服務規劃案例，或自行由 DAML-S/UDDI 匹配器中找到相關的服務，並進行規劃。在本架構剛開始運作時，由於案例少，因此大多必須依靠服務規劃者到 DAML-S/UDDI 匹配器中搜尋和整合這些服務，當案例庫中的案例越多，本架構在搜尋上的效率才會越高。

在整合服務上，我們延用 MWS 中的參數轉換矩陣，透過 MWS 提出的這個方法，服務和服務之間能夠進行銜接，整合成一個無縫的應用程式，使用者只需要知道如何輸入自己的需求，本架構便會整合多個服務，並將執行後的結果送回給使用者，對於使用者來說，並不知道這個結果是結合多個服務而產生的。

第四節 自動調用

本架構用來自動執行服務的元件為服務執行者，它能夠接收服務規劃者的服務規劃，並透過這些規劃執行服務，然後將結果送回給使用者。由於 MWS 在執行上並沒有詳細的描述，在自動執行的實際運作上，服務的本體論和服務的 WSDL 文件必須互相配合，本體論能

夠實現語意，而 WSDL 則是實際執行的細部描述。因此，我們在這個部份加入了 DAML-S 虛擬機器，它能夠利用 DAML-S 進行語意的處理，然後透過 WSDL 進行實際的執行。

過去網路服務尚未引入語意網概念時，網路服務的連結必須透過程式設計師撰寫程式去執行服務，這不僅非常被動性，也限制了網路服務擴展的方向，自從語意網概念的引入，網路服務邁向一個新的紀元。自動調用類似於代理人的角色，透過上一節所描述的自動整合，再自動將這些整合的服務連結和執行，讓網路服務的未來有無限的可能。

第五章 整合型購物決策資訊架構(ISDIF)

ISDIF 的建置，必須包含資訊內容及系統架構兩部份。在本章，我們將 ISDIF 分為網路服務資訊內容及整體架構兩部份來做說明。另外，我們也以一個例子來說明 ISDIF 的實際應用。

第一節 網路服務資訊內容

消費者在購買商品前，大都會參考一些資訊，我們將這些資訊分為三類：商品資訊、商品評論，及商品購買率與購買途徑。這三類分別描述如下：

- **商品資訊**：消費者在購買一件商品前，基本上最需要了解的資訊，包括商品的介紹、功能、價錢、常見問題解決等。過去因為網路尚未發達，即使各地商品的價錢不一，消費者也會因無法得知資訊，或考慮到路程遙遠，而只能在較近的商家購買商品。自網路發達後，資訊散佈快速，加上宅配的服務又趨於成熟，不僅讓網路購物成為目前的主流，消費者更能在網路上找到一個最滿意的價錢來購買商品。因此，本系統提供最基本的資訊即商品的相關介紹，包括商品的特色、功能、價錢等。
- **商品評論**：消費者在得知商品資訊後，另一個考慮購買的因素，即曾經購買此商品之消費者的評價。這種口耳相傳的購買現象已

經盛行已久，然而因過去沒有網際網路，人與人在評論一件商品時，只能在某個小地區流傳而已。自網際網路發達後，各種商品討論區、虛擬社群等網路應用紛紛出現，評論商品不再只限於某個小地區，這些評論將散佈到世界各個角落。因此，本服務另一個必需的資訊為消費者對商品的使用經驗和評論。

- **商品購買率與購買途徑：**本系統會在使用者查詢某樣商品後，紀錄相關資訊，並以圖表統計方式呈現給使用者。統計的資料可分為三類，分別為該服務經使用者搜尋後，在系統中列出的數量（曝光量）、使用者進一步點擊該服務查看的數量（點擊量）、以及使用者在該服務下單的數量（交易量）。經由這些資料，我們可算出該服務的點擊率以及商品購買率，這些資訊將成為使用者另一項購買的參考依據。因為在購買方式上，可分為兩大類：*網路購買*和*實體商店購買*。在網路購買上，本系統能夠直接追蹤使用者在網路下單的相關資訊；但在實體商店購買上，因無法透過電腦自動紀錄，只能以問卷方式取得使用者是否購買以及購買地點等資訊。

第二節 以 MWS 為核心之整合型購物決策資訊架構

在圖 14 我們呈現出本文所提出的整合型購物決策架構 (ISDIF) 的完整架構。我們主要結合產品網站、虛擬社群、產品討論區、購物網站等網路服務。經過 MWS 的發現、整合和執行後，可以呈現商品

資訊、商品評價、商品購買率，以及商品購買途徑等相關的購物決策資訊。本架構主要可分為三個部份，分別為服務搜尋者、服務規劃者、及服務執行者，我們依序說明如下。

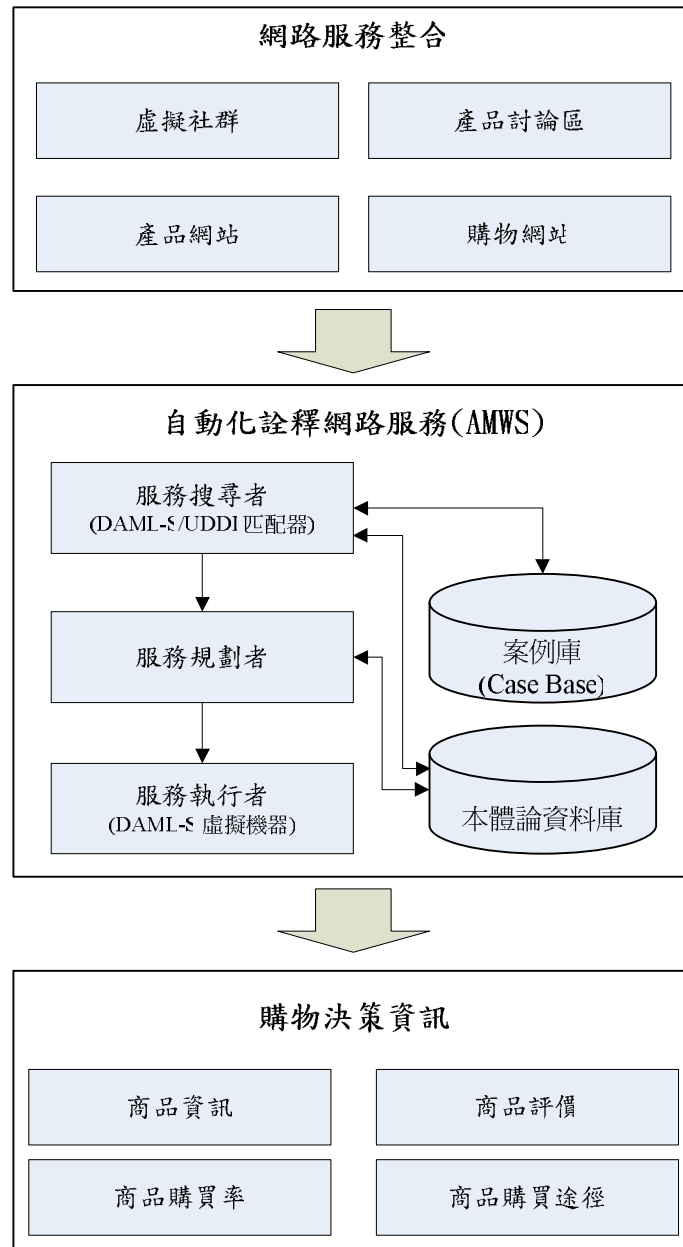


圖 14 以 MWS 為核心之整合型購物決策資訊架構

(1) **服務搜尋者(Service Finder)**

雖然在傳統的 MWS 中，**案例庫(Case Base)**能夠減少搜尋和服務整合的時間和成本，但在使用本體論發現服務方面，MWS 並沒有更詳細的描述。為了解決此一問題，在我們提出的網路服務自動化架構中，保留了案例庫，並加入 DAML-S/UDDI 匹配器，因而構成一個較完整的服務搜尋部份。

在實際的運作上，首先，使用者必須輸入產品特徵來查詢產品。我們將產品特徵分為四大類，分別為所在地區、產品品牌、產品類型、與產品型號。使用者輸入的產品特徵會轉換成本體論，並到案例庫進行比對。儲存在案例庫中的資料可分為下列兩部份：

(1) 網路服務及其本體論之 URI，此 URI 會對應到 DAML-S/UDDI 匹配器中的本體論資料庫，我們將這些服務細分為三大類，分別是產品資訊、產品討論、與購買途徑。

(2) 過去查詢的案例特徵值、對應的服務、與後續結果。

案例推理引擎(Case Base Reasoning, CBR)是本架構推理的核心，它會依據使用者輸入的商品特徵，和案例庫中的案例做比對。我們提出一套比對流程，其步驟如下：

- 利用使用者輸入商品之特徵值，與案例庫中儲存的案例特徵值比

對，若相似度 100%，亦即使用者輸入的特徵值與案例庫中某個案例的特徵值相同，則套用此案例的對應服務。

- 若比對的相似度未符合要求，這時我們將此訊息傳給服務規劃者，它會利用輸入的特徵值本體論，到 DAML-S/UDDI 匹配器中的本體論資料庫進行比對。此處要特別注意的是，所找到的服務，必須要包含產品資訊、產品討論、以及購買途徑這三大類。

初期因為案例庫中儲存的案例不多，必須從 DAML-S/UDDI 匹配器去搜尋服務，因此在執行上會花費較多的時間。當案例庫中的案例越來越多，儲存的網路服務也越多，我們在搜尋的時間上將會慢慢縮短。為了確保得到的資訊能夠完整，因此我們將案例庫中儲存的網路服務分為三類，並規定所找到的網路服務，必須包含這三種，缺一不可。當找到符合的案例後，服務搜尋者會將對應的網路服務傳給服務規劃者。

(2) **服務規劃者(Service Planner)**

服務規劃者用來接收服務搜尋者找到的案例，若服務搜尋者在案例庫中並未找到符合的案例，服務規劃者便會自行到 DAML-S/UDDI 匹配器尋找相關的服務，然後透過註冊中心取得服務的 WSDL 文件和基本資料，並將這些服務分類為產品資訊、產品討論、以及購買途徑，這也是呈現給使用者的三個資訊分類。

當服務規劃者有了這些服務的資訊後，便會開始整合服務，整合方式是以參數轉換矩陣做為服務規劃，表 4 是以圖 9 為例的一個參數轉換矩陣，由表中可知，每個服務的輸入參數會被設為-1，輸出參數設為 1，當兩個服務的同一個參數，一個為-1，一個為 1，則可將此兩個服務銜接起來。當這些相關的服務整合規劃後，便會將這些整合的規劃交由服務執行者處理。

另外，服務規劃者還必須統計過去使用相同特徵值搜尋的案例數、將這些案例的後續結果做整理，以及過去相似案例的統計等，這部份也就是前面所說的商品購買率，是另一項供使用者參考的資訊。

表 4 參數轉換矩陣

	Web Service1	Web Service2	Web Service3	Web Service4
Var1	0	-1	0	0
Var2	-1	0	0	0
Var3	-1	0	0	0
Var4	0	0	-1	0
Var5	1	-1	0	0
Var6	0	1	0	-1
Var7	0	0	1	-1
Var8	0	0	0	1

(3) 服務執行者(Service Executor)

服務執行者收到服務規劃者傳來的資訊後，會到 DAML-S 虛擬機器進行服務調用，利用取得的網路服務調用資訊，連結到網路服

務，並透過服務的規劃執行這些服務及取得回傳結果，最後將結果依據規劃分類，並加入統計資料呈現給使用者。服務執行者呈現給使用者的是購買商品的決策資訊，當使用者決定是否購買，以及利用何種方式購買，系統都會將這些資訊紀錄到資料庫中，以讓未來的使用者做為另一項參考資訊。

第三節 ISDIF 應用實例

在本節，我們以購買手機的決策資訊為例來說明**整合型購物決策資訊系統** (ISDIF)的應用。首先，我們必須了解本系統會整合哪些網路服務、其服務內容為何，以及這些服務提供者在提供服務時必須注意的事項。我們將 ISDIF 會整合的網路服務分為三類(例如表 5 所示)，分別為產品資訊、產品討論、購買途徑。我們依序說明如下：

- **產品資訊**：主要提供產品資訊查詢，大部份產品公司都會在自己的網站上介紹公司產品，因此，此類網路服務，可由產品公司或專門介紹產品之網站提供，有些公司會將顧客對產品常見問題解決放於網站上供查詢，我們將這些資訊歸於此類。在此類服務的操作介面上，我們建議輸入參數為產品名稱或型號之關鍵字，而輸出是符合關鍵字的產品資訊及常見問題解答。
- **產品討論**：主要提供使用者一個討論空間，用來分享使用者購買

後的心得、經驗分享等。目前網路上有許多專門討論某些特定產品的討論區或論壇，另外，也有大型網站提供家族或社群功能，許多產品有自己專屬的家族或社群，供顧客一起討論或分享。此類服務的操作介面上，我們建議輸入參數為產品名稱或型號的關鍵字，輸出的資訊是符合關鍵字的產品討論文章，使用者也可回應文章參與討論。

- **購買途徑**：主要提供產品購買資訊，以及網路訂購服務，最著名的如 Yahoo 奇摩拍賣、e-Bay 拍賣，也有網站提供各地實體商店經營門市的資訊等。在此類操作介面上，我們建議輸入參數為產品名稱或型號之關鍵字，輸出為相關產品購買方式、價錢、賣家資訊、購買注意事項、訂單連結等。

表 5 ISDIF 整合網路服務分類表

類型	服務提供者
產品資訊	產品公司網站：MOTOROLA、SAGEM、BENQ、NOKIA
產品討論	討論網站：PALMisLIFE、手機王、Yahoo 奇摩家族
購買途徑	線上購物網站：Yahoo 奇摩拍賣、e-Bay 拍賣、EZ-Shop

為了符合本系統的設計，服務提供者必須使用統一標準用語，利用 DAML-S 建立描述服務的本體論。經由本體論描述網路服務，電腦能夠了解服務內容為何，並能自動存取相關資訊做處理。例如，Motorola 公司提供一個可查詢手機資訊的網路服務，Motorola 公司必

須使用 DAML-S 描述此服務可查詢哪幾款型號的手機資訊，其本體論型式如圖 15 所示，利用此本體論文件，本系統便能判斷此服務是否和使用者所需相符。

```
<daml:Class rdf:ID = "Type">
  <rdfs:label>Type</rdfs:label>
  <rdfs:comment>The type of a mobile phone</rdfs:comment>
  <daml:one of rdf:parseType = "daml:collection">
    <daml:Thing rdf:ID = "A630">
      <rdfs:label>A630</rdfs:label>
    </daml:Thing>
    <daml:Thing rdf:ID = "A840">
      <rdfs:label>A840</rdfs:label>
    </daml:Thing>
    <daml:Thing rdf:ID = "C155">
      <rdfs:label>C155</rdfs:label>
    </daml:Thing>
    <daml:Thing rdf:ID = "E310">
      <rdfs:label>E310</rdfs:label>
    </daml:Thing>
  </daml:oneOf>
</daml:Class>
```

圖 15 Motorola 手機查詢服務本體論案例

由於目前 UDDI 並沒有支援 DAML-S 格式的網路服務描述文件，為了解決此一問題，本系統將 DAML-S 與 UDDI 結合在一起，以實現語意網路及網路服務自動化。MWS 提出在發佈服務時，將 DAML-S 文件的 URI 一併發佈於 UDDI 中，服務需求者再從 UDDI 中找到服務，並從其 DAML-S 的 URI 取得文件和處理。而 DAML-S/UDDI 匹配器則是透過 DAML-S/UDDI 轉換器自動將服務的廣告訊息轉換成 WSDL 格式，發佈於 UDDI 中，並將本體論儲存

於資料庫裡，以便做為配對之用。本系統是使用 DAML-S/UDDI 匹配器發佈機制，因此，服務提供者在建立好服務描述文件後，必須透過 DAML-S/UDDI 匹配器發佈到 UDDI 中，如此才能滿足本架構運作的條件。

在案例庫的資料庫中，主要可分為兩個部份，其設計的方式分別如表 6 及表 7 所示。表 6 儲存了網路服務本體論之 URI，並將他們分為三類（亦即產品資訊、產品討論、與購買途徑）；表 7 儲存了每個案例和其特徵值、對應的服務、以及結果。經由以上這些步驟，我們便可建構出整個 ISDIF 系統。

表 6 網路服務本體論之 URI 儲存表

類別	網路服務	本體論之 URI
產品資訊	I1	http://www.motorola.com.tw/ontologies/I1.daml
	I2	http://www.motorola.com.tw /ontologies/I2.daml
產品討論	T1	http://www.motorola.com.tw /ontologies/T1.daml
	T2	http://www.motorola.com.tw /ontologies/T2.daml
購買途徑	C1	http://www.motorola.com.tw /ontologies/C1.daml
	C2	http://www.motorola.com.tw /ontologies/C1.daml

表 7 網路服務特徵值、對應服務、後續結果儲存表

案例	特徵值	對應的服務	結果
Case1	嘉義,印表機,EPSON,EPL-5900	I1,T1,C2	未做購買行為
Case2	台北,電腦,Acer,Veriton 5100	I2,T2,C1	Yahoo 拍賣購買

現今手機廠牌眾多，各家推出的手機型號五花八門，消費者在選擇上常覺得無所適從。假設張三利用 ISDIF 系統欲查詢 Motorola 型號 V300 這款手機，於是將產品手機、公司 Motorola、型號 V300、與地區嘉義等輸入 ISDIF。經由 ISDIF 的搜尋和比對，在註冊中心 UDDI 中找到了中華電信手機優惠方案、比價王手機館、奇摩拍賣、eBay 拍賣、Motorola 官方網站、與 Motorola 虛擬社群等網路服務。

從 Motorola 官方網路服務中，我們得到此款手機的產品資訊、相關問題解決及維修資訊；從比價王手機館與 Motorola 虛擬社群網路服務中，我們統整出此款手機的評價和討論；從中華電信手機優惠網路服務、奇摩拍賣、eBay 拍賣與比價王手機館網路服務中，我們找到嘉義地區各通訊行的價格與網路上此款手機拍賣的資訊，並依照價格高低排列出來。

第六章 結論與未來展望

第一節 結論

我們以詮釋網路服務(MWS)為核心，並結合 DAML-S/UDDI 匹配器和 DAML-S 虛擬機器建置一個**整合型購物決策資訊架構**(ISDIF)，讓使用者能夠透過本架構找到相關的資訊，以做為購物的決策依據。網路服務自動化已是現在和未來的趨勢，本研究提出的網路服務自動化架構，改進了 MWS 原來的缺點，讓網路服務自動化更邁進一步。從我們的研究顯示出：ISDIF 在解決網路購物相關應用的整合問題上有很大的貢獻，其能整合建置於不同平台上的系統，幫助不同系統間資料的交換。網路購物是未來的主流，本架構能夠整合網路上的購物資訊，對未來網路購物將有很大的幫助。

第二節 未來展望

雖然本文所提出的 ISDIF 在語意網路服務和購物決策資訊的研究領域方面，已向前推進了一大步。然而，在未來展望方面，依然會面臨到一些問題與挑戰。例如：目前網路服務的發佈尚未普及，鮮少公司將網路服務發佈於註冊中心以供別人連結使用，因此本系統將面臨的第一個問題是註冊中心裡網路服務的缺乏，當註冊中心裡相關網路服務發佈的越多，本系統才能整合更多相關服務以供使用者參考。

此外，ISDIF 是以 DAML-S 描述的本體論進行比對。服務提供者

在發佈服務時，除了建立 WSDL 檔案外，還必須以 DAML-S 建立服務的本體論，如此才適用於本系統。而 OWL 服務聯盟 (OWL Services Coalition)[14]與全球資訊網協會 (W3C/OWL)[18]已呈現出最新的本體論網路服務描述語言版本，名為 *OWL-S* (Ontology Web Language for Services)。雖然 OWL-S 繼承了 DAML+OIL 的框架和大部份的語法和語意特徵，但 McCool[10]指出目前有關 OWL 資料格式 (Data Format) 可利用的資訊很少。因此，未來是否採用最新版本本體論網路服務描述語言，以及是否會面臨到哪些衝突或困境，亦是我們未來研究發展的方向。

參 考 文 獻

1. 邱宏昇、陳文誌、蕭淑玲，「企業間電子商務 XML 文件轉換技術之研究」，TANET 2000 台灣區網際網路研討會，2000 年。
2. 戚玉樑、關銘，「使用語意解釋代理人(SAA)改善網路服務組合準確性」，2004 國際學術研討會，2004 年。
3. 戚玉樑，「網路服務技術導論」，台北：全華科技圖書有限公司，民 93 年。
4. Bruijn, J., Fensel, D., Keller, U., and Lara, R., “Using the Web Service Modeling Ontology to Enable Semantic E-Business,” *Communications of the ACM*, December 2005, pp.43-46.
5. Burstein, M., Bussler C., Zaremba M., Finin T., Huhns M. N., Paolucci M., Sheth A. P., Williams S., ”A Semantic Web Service Architecture,” *IEEE Internet Computing*, September/October, 2005, pp.72-81.
6. DAML Services Coalition, “DAML-S: Semantic Markup for Web Services,” <http://www.daml.org/services/daml-s/0.9/daml-s.html>, 2003.
7. Hepp, M., “Semantic Web and Semantic Web Services : Father and Son or Indivisible Twins ?” *IEEE Internet Computing*, March/April 2006, pp. 85-88.
8. Huhns, M. N. and Singh, M. P., “Service-Oriented Computing: Key Concepts and Principles,” *IEEE Internet Computing*, January/February 2005, pp.75-81.
9. 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.
10. McCool, R., “Rethinking the Semantic Web, Part2,” *IEEE Internet Computing*, January/February 2006, pp.93-96.
11. Ogbuji, U., “An introduction to RDF : Exploring the standard for Web-based metadata,” <http://www-128.ibm.com/developerworks/xml/library/w-rdf/?dwzone=xml&dwzone=xml>, December 2000.

12. Ouellet, R. and Ogbuji, U., “*Introduction to DAML : Part I*,”
<http://www.xml.com/lpt/a/2002/01/30/daml1.html>, 2002.
13. Ouellet, R. and Ogbuji, U., “*Introduction to DAML : Part II*,”
<http://www.xml.com/lpt/a/2002/03/13/daml.html>, 2002.
14. OWL Services Coalition, “*OWL-S: Semantic Markup for Web Services*,” <http://www.daml.org/services/owl-s/1.0/owl-s.html>, 2003.
15. Paolucci, M. and Sycara, K., ” Autonomous Semantic Web Services,” *IEEE Internet Computing*, September/October 2003, pp.34-41.
16. Sycara, K., Paolucci, M., Ankolekar, A., and Srinivasan, N., “Automated Discovery, Interaction and Composition of Semantic Web services,” *Journal of Web Semantics*, Volume 1, Issue 1, December 2003.
17. Tao, Y.H., Hon, T.P., and Sun, S.I.,”An XML Implementation Process Model for Enterprise Applications,” *Computers in Industry* (55), 2004 , pp.181-196.
18. W3C/OWL, “*OWL Web Ontology Language Overview*,” W3C Recommendation, <http://www.w3.org/TR/owl-features/>, February 2004.
19. W3C/RDF, “*Resource Description Framework (RDF) Primer*,” W3C Recommendation, <http://www.w3.org/TR/rdf-primer/>, February 2004.
20. Wikipedia, The Free Encyclopedia, “*Ontology (computer science)*,” [http://en.wikipedia.org/wiki/Ontology_\(computer_science\)](http://en.wikipedia.org/wiki/Ontology_(computer_science)).