

第二章 文獻分析

本章共分為五節，第一節簡述有關知識的意涵、來源與獲取方式；第二節進一步探討知識表徵的意涵、層次與要素；第三節探究網路知識表徵環境、難題與發展趨勢；第四節談論電腦科學界的網路知識表徵方法；最後，第五節則檢視知識表徵語言之比較研究，作為本研究設定比較面向之參考。

第一節 知識的意涵、來源與獲取方式

由於知識表徵為本研究探討的重點之一，因此本節擬先簡述知識的意涵、來源與獲取方式，而後於下兩節進一步探討知識表徵的意涵、層次與要素，及網路知識表徵環境、難題與發展趨勢。

一、資料、資訊與知識

當人們談到知識時，多傾向將其與資料和資訊加以區別，Huseman and Goodman 定義「資料」是以沒有任何判斷、觀點、以及情境脈絡的角度來描述客觀的事實；「資訊」是賦予相關性與目的之資料；「知識」則是資訊和個人經驗、真理、判斷、直覺、與價值觀的組合。（Huseman & Goodman, 1999，轉引自陳雪華，民 92）Davenport and Prusak 更進一步指出，若要由資料轉化為資訊的層次，需要有脈絡化（Contextualized）、歸類、計算、訂正、及濃縮等加值活動；而欲將資訊再轉化為知識的層次，則需有比較、知曉因果（Consequences）、關聯、與人際溝通的過程。（Davenport & Prusak, 1998，轉引自邱子恆，民 91）

二、外顯知識與內隱知識

依形式觀之，可將知識分為「外顯知識」（Explicit Knowledge）與「內隱知識」（Tacit Knowledge），最早提出此概念的學者是 Polanyi

(西元 1891 至西元 1976)，他在 1966 年時於其著作《默會面向》(The Tacit Dimension) 中特別界定「外顯知識」與「內隱知識」的差異；簡言之，「外顯知識」是可藉由語言文字或機器處理等方式轉為符號，因此在流通與分享上較為容易；而「內隱知識」則無法轉譯，難以用文字描述，需透過人們相互溝通才能進行知識分享。(Newman, 1999) 另外，Allen Newell (西元 1927 至西元 1992) 以美國人工智慧學會主席身份，於 1980 年發表的《知識階層》(The Knowledge Level) 一文中，由人工智慧學的觀點對知識作的定義則認為，由符號表徵的「外顯知識」永遠無法精確反映知識本身，Newell(1981)表示：「知識無法由符號層次的結構來加以表徵，因為知識需要同時處理結構與過程。知識永遠是抽象且無法精確掌握的，因此基本上，只能將知識視為一種「近似值」，它須透過符號表徵來使其能被實際觀看。

三、知識的來源與獲取方式

著名的人工智慧學者 Sowa(2000)在其經典著作《知識表徵：邏輯、哲學、與運算基礎》(Knowledge Representation: Logical, Philosophical, and Computational Foundations) 認為人們獲取知識的來源有三，分別為世界 (World)、模型 (Model)、與理論 (Theory)。對真實世界中的知識需透過觀察 (Observation) 來獲取；存在於模型中的知識可藉由模擬 (Simulation) 而來；而理論當中的知識則需靠推論 (Deduction) 導出，如下圖所示。運用觀察來獲取知識是最直接的方式；在工程學界通常是以較間接的方式，也就是透過建構已存在的模型或系統，並遵守真實世界情況的模擬行為來瞭解；由理論來推論或演繹則是更間接的一種方式，理論的公理 (Axiom) 是將模型中的關係形態予以抽象化而成，形成真實世界的抽象概念。欲獲取知識，需視現有資料、所需的精確度，以及可投入的時間與金錢等因素，來決定要使用何種方式。

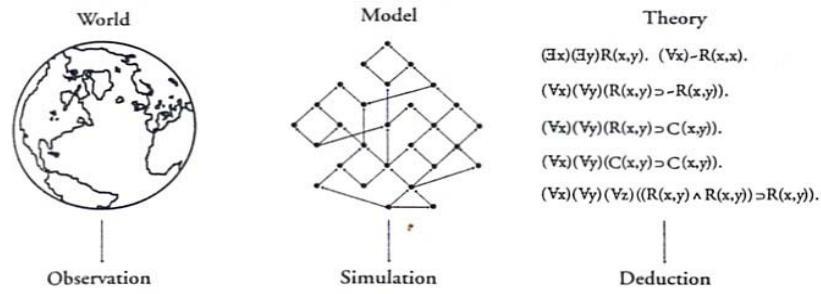


圖 2-1 知識的來源與獲取方式

資料來源：Sowa, J.F.(2000). Knowledge representation: Logical, philosophical, and computational foundations. Pacific Grove: Brooks/Cole.

上述三種知識來源彼此是互補關係，而非競爭關係；也就是說，實務上這三種知識來源通常會結合在一起使用。電腦模擬會根據最新的觀察結果持續更新；全球共通的模型則透過規則來處理本地的事務。由世界、模型、到理論是逐漸抽象的過程，愈抽象就愈會流失一些細節，但每一次的意義流失又因通則的增加而獲得補償，雖然會有意義的流失，但模型則可應用於更多不同的情境，且科學的理論亦可讓工程師在不同主題領域加以應用。總而言之，知識獲取是由資料到模型到理論的反覆週期，並以新的理論對舊有資料再詮釋。真實世界運行其中，人們透過反覆的抽象與再詮釋的過程學習和模擬知識。(Sowa, 2000)

筆者認為 Sowa 的論述某種程度呼應了社會學學者 Berger and Luckmann(1967)的陳述：「日常生活的知識基礎，即為主觀過程（與意義）的客觀化，以及透過客觀化過程而建構的互為主觀的（Intersubjective）常識世界。」換言之，透過文字或機器可處理的方式來表徵知識，是客觀化的過程；將知識轉為符號（語言也是一種符號體系），使人或機器能處理，尊重不同的觀點，同時也分享這個現實的共同意義，則是互為主觀的性質。

知識表徵雖受到語文在表達方面的限制，及知識本身所涉及的層面相當複雜等問題的影響（若考量知識的結構與關聯等），表徵後的

知識(外顯知識)實難精確,但人們可以嘗試勾勒出知識的「近似值」,分享與尊重不同的詮釋觀點。

第二節 知識表徵的定義、層次與要素

由於本研究欲分析比較的是電腦科學界的兩種知識表徵方法—RDF 與 Topic Maps,因此本節將由電腦科學的角度,來看學者提出的知識表徵的定義、層次與要素。

一、知識表徵的定義

知識表徵向來是人工智慧領域關心的重要課題,其主要在探討如何以正規的方式在資訊系統中儲存及處理知識;而圖書館界對知識表徵和資訊組織的研究同樣是不遺餘力,較具代表性的方法包括分類表、標題表、與索引典等。電腦科學界目前也已發展出許多作為知識表徵的語言(例如 OWL、CycL、與 KIF 等)和標記法(例如 RDF 和 DATR 等)。這些語法一般都會以邏輯和數學為基礎,並具有易於電腦剖析的文法。(Wikipedia, 2004a)由電腦科學的角度觀之,知識表徵是嘗試將人類的知識予以塑模(Modeling),再進一步編碼儲存於電腦中並呈現出來,進而使人們感知所欲傳遞的知識或資訊之過程,如下圖所示。

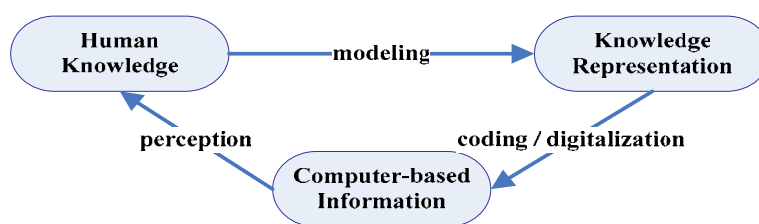


圖 2-2 知識表徵示意圖

資料來源：本研究

美國麻省理工學院 (MIT) 學者 Davis, Shrobe, and Szolovits(1993) 曾對知識表徵提出著名定義：

- **知識表徵是替代品 (Surrogate)**。無法直接儲存於電腦中的實體物件、事件、與關係，需透過符號作為替代品來表徵外部事物。電腦程式藉由處理這些內部的替代品，可以模擬外部事物，並據以從事推論。
- **知識表徵是一組對知識本體的約定 (A Set of Ontological Commitment)**。就哲學來說，Ontology 是探尋有關事物存在本質的研究。對資料庫或知識庫來說，Ontology 決定了事物存在的種類 (Category)，這些種類也就反映了程式設計人員或知識工程師對 Ontology 的約定，也就是我們以什麼詞彙 (或術語) 來認識世界。
- **知識表徵是智慧推論的片段理論 (Fragmentary Theory of Intelligent Reasoning)**。為能對某主題領域進行推論，知識表徵必須能描述其行為與互動模式，因此這種描述也就形成應用程式領域的理論，這些理論可能是透過明確的公理予以聲明，或編輯成可執行的程式。
- **知識表徵是提昇計算的媒介 (Medium for Efficient Computation)**。除了表徵知識外，人工智慧系統必須將知識以在電腦設備中能有效處理的形式予以編碼。
- **知識表徵是人類表達的媒介 (Medium of Human Expression)**。好的知識表徵語言應能促進瞭解人工智慧的知識工程師，和知悉應用程式的主題專家彼此溝通。主題專家應理解並驗證知識工程師所寫下定義與規則，是否反映該主題領域的理論。

不過 Davis et al.(1993)也表示，不論多麼努力表徵知識，所有的表徵仍舊難以做到盡善盡美、忠實呈現，因為人類的知識是不一致、定義不明、無客觀結構，讓人難以形容的；換言之，人們無法不犧牲部分意義，而完全忠實呈現所欲表達之知識，因此，任何知識表徵方法皆無可避免地具有某種程度的瑕疵。總而言之，由電腦科學的角度來看，知識表徵是指將知識由實體物件轉為電腦化資料的過程，而在轉換的過程中，涉及不同內涵的表徵層次，因此，以下擬進一步探討

表徵層次。

二、知識表徵的層次 (Level)

由電腦科學的角度觀之，「表徵」是用一組符號結構 (Symbol Structure) 來描述物件、事件、與關聯等；而代表物件、事件、與關聯等的一組符號詞彙，是依定義好的規則事先定義的。(Goel, 2004) 知識表徵是指由實體物件 (Physical Object) 轉為電腦資料的過程，在轉換的過程，可以將表徵分為幾種不同層次。電腦程式必須將現實世界的事物透過編碼的方式，轉換為電腦可處理的資料，才能加以利用。例如想利用電腦表示名叫 "Harry" 的人，那麼 "Harry" 這個字串在電腦中需利用某種編碼方式，例如 ASCII 編碼或 EBCDIC 編碼等方式記錄為 0x486172727900 或 X'C8819999A8，而這些碼最後會以 1 和 0 儲存在電腦中，如圖 2-3 所示。(Sowa, 2000)

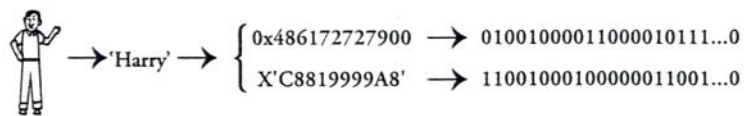


圖 2-3 實體物件與電腦表徵

資料來源：Sowa, J.F.(2000). Knowledge representation: Logical, philosophical, and computational foundations. Pacific Grove: Brooks/Cole.

知識表徵的過程基本上牽涉到符號 (Symbol)、物件 (Object)、與意義 (Meaning) 三者，因此有學者提出「意義三角形」(Meaning Triangle) 的概念來說明此三者間之關係。第一位區分符號、物件、與意義的人是亞理士多德 (Aristotle, 西元前 384 年至西元前 322 年)，Ogden and Richards(1923)則將「意義三角形」的概念普及化。以圖 2-4 為例，其最左方之三角形的左下方是名叫 Harry 的人 (物件)，該三角形的右下方是他的名字 (可印出來的符號)，上方則是 Harry 的形體的心象 (無法印成紙本的概念)，此三者即構成「意義三角形」。(Sowa, 2000)

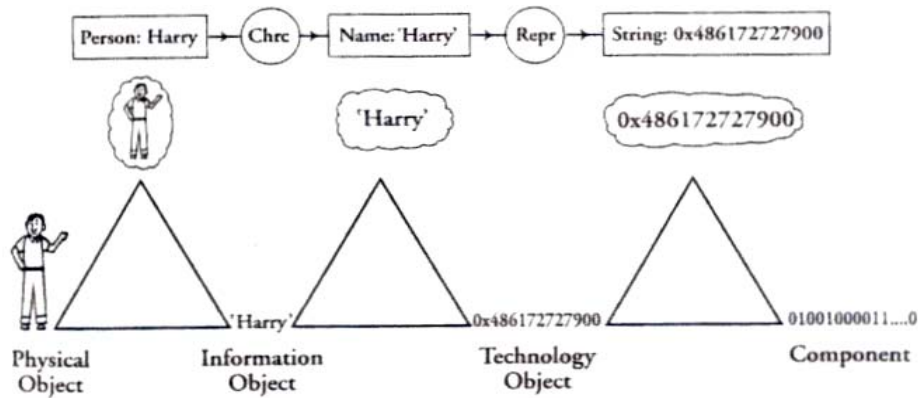


圖 2-4 實體物件轉為電腦資料的意義三角形階段

資料來源：Sowa, J.F.(2000). Knowledge representation: Logical, philosophical, and computational foundations. Pacific Grove: Brooks/Cole.

運用「意義三角形」的概念，同樣可以用來表示實體物件如何透過電腦表徵的過程。上圖的“Harry”這個人名在系統層次如同資訊物件 (Information Object)；上圖中間的三角形描繪出將”Harry”轉為符號“0x486172727900”，這是程式人員會看到的字串，視為技術物件 (Technology Object)；而上圖最右方的三角形則是顯示將字串再轉為位元的元素 (Component)。上圖三角形上方的雲狀圖則是表示所傳遞的概念，由於概念無法觀察也無法呈現於紙本之上，因此雲狀圖上方則再進一步利用概念圖加以說明，也就是 Harry 這個人的概念透過字元轉為他的名字，而名字則透過表徵形成儲存於電腦中之字串。(Sowa, 2000)

McCarthy and Hayes(1969)將這種由實體物件轉為電腦化資料的表徵方式，分為「知識論層次」(Epistemological Level)與「試探式層次」(Heuristic Level)二種。前者傾向敘述性 (Declarative) 或邏輯性的表徵；而後者則是偏向程序性 (Procedural) 的表徵。「知識論層次」主要欲瞭解在應用中，有關知識的物件與處理過程；「試探式層次」則是為了有助電腦的模擬過程，而著重物件與程式的資料結構。隨後 Brachman(1979)將 McCarthy and Hayes 的「知識論層次」進一步細分為四種不同層次，並將試「探式層次」重新命名為「實作層」，

進一步說明如下：(Sowa, 2000)

- **實作層 (Implementational)**：是指資料結構的層次，例如清單、指標 (Pointer)、或其他的程式概念 (Notion)。
- **邏輯層 (Logical)**：有關命題、述語 (Predicate)、變量、量詞、與布林邏輯之符號邏輯。
- **知識層 (Epistemological)**：以子類型 (Subtype)、結構關係、與繼承 (Inheritance) 來定義概念類型。
- **概念層 (Conceptual)**：關於語意關係、物件、與活動的層次。
- **語言層 (Linguistic)**：有關自然語言的任意概念、字詞、與表達的層次。

Branchman 表示在這五項表徵層次中，「邏輯層」的考量較單純且不涉及內容 (Content-independent)；「概念層」則需考量概念的特定意義，但不必考慮其內部結構；「知識層」則可具體指明概念單元的正規結構，及概念單元間的相互關係；換言之，「邏輯層」處理抽象述語，「概念層」著重在特定概念的處理，而「知識層」則關注對概念的整體見解。

此外，Guarino(1995)認為上述的表徵五層次中的「知識層」是「不明的一層」(Missing Level)，可見有關概念間的關聯，以及整體結構的處理並不是件易事。因此 Guarino 便提出在「知識層」與「概念層」間，應加入「知識本體層」(Ontological Level)。如表 2-1 所示。

表 2-1 表徵層次的主要特徵

層次 (Level)	基本概念 (Primitive Concepts)	主要特徵 (Main Feature)	解釋 (Interpretation)
邏輯層 (Logical)	述語 (Predicates)	形式 (Formalisation)	任意的 (Arbitrary)
知識層 (Epistemological)	結構原詞 (Structuring Primitives)	結構 (Structure)	任意的
知識本體層 (Ontological)	滿足意義的假設 (Postulates)	意義 (Meaning)	限定的 (Constrained)
概念層 (Conceptual)	認知原詞 (Cognitive Primitives)	概念 (Conceptualisation)	主觀的 (Subjective)
語言層 (Linguistic)	語言原詞 (Linguistic Primitives)	語言 (Language)	主觀的

資料來源：Gurarino, N.(1995). The ontological level. Retrieved December 4, 2004, from <http://www.loa-cnr.it/Papers/OntLev.pdf>

由 Gurarino 的論述我們可以知道，Ontology 是將人的知識轉為電腦可處理的「知識」的重要媒介，若缺乏「知識本體層」，也就是缺乏對概念的意義、概念所涉及的角色本身的意義、以及每個角色對概念意義的限定，那麼概念的意義及概念彼此間的關係就會是任意的；因此在設計表徵架構 (Scheme) 時，應考量的要素包括：(Goel, 2004)

- 一組由符號結構組成的詞彙或 Ontology
- 指出真實世界的物件和關係，及表示物件與關係的符號之語意對應 (Semantic Mapping)
- 結合詞彙成表徵結構的一組規則
- 能表達真實世界知識的相關內容

至於如何才稱得上是好的表徵，人工智慧學者 Winston(1984)認為應具備的特性包括：明確 (Explicit)；有清楚的限制條件 (Constraint)；就所處理的問題而言必須完整 (Complete)；簡明 (Concise。沒有不必要的冗詞)；容易讓人瞭解 (Transparent)；效率高，可節省電腦的

處理時間與記憶空間；程序可被計算 (Procedures Exist to Compute)；以及隱藏細節，亦即有特別的要求時才顯示細節。

總而言之，由電腦科學的角度來看，表徵層次是指由實體物件轉為電腦化資料的過程，在轉換的過程中，涉及不同內涵的表徵層次，包括從語言提昇到概念，進一步到概念和概念間的邏輯關係。若能將「知識本體層」定義的愈精確，才愈能有助於電腦後續的自動處理，甚至是 Ontology 的分享。

三、知識表徵應具備的要素

Rich and Knight(1991)認為若要讓人工智慧技術能應用知識的話，知識表徵應滿足下列要求：

- 知識表徵應使被表徵的一組知識，具有共通且重要的屬性，而不是分別去表徵每個知識的情境 (Situation)，不具備這種屬性的都稱為「資料」，而非「知識」。
- 知識表徵應可被提供該知識的人理解。
- 知識表徵應易於修正調整，以滿足改正錯誤及反映改變的需求。
- 知識表徵應可應用於不同情境，即使該表徵尚未制定完成。
- 知識表徵應能藉由縮小可能考量的範圍，來協助其克服本身龐大的問題。

此外，Sowa(2000)表示，如欲在電腦中從事知識表徵，事實上無法單靠自然語言達成，因為電腦需要邏輯，利用公式來表示 (Formulate) 知識。或許會有某些知識無法藉由邏輯表示，但若有那樣的知識存在的話，那麼它就不可能使用任何其他標記方法在任何電腦中表徵或處理。邏輯不只是一種標記方法，邏輯的表達能力包括在電腦中，能儲存或設計程式來處理任何資訊，而知識表徵語言若要具備邏輯，必須包含四項特徵：

- **詞彙 (Vocabulary)**：邏輯必須具備一組能表示字元、字詞、圖像、或聲音等的符號。符號又可分成四類，分別是：不受主題 (Domain)

影響的邏輯符號，例如 \forall 與 \wedge ；會受主題影響的常數 (Constant)，也就是在應用領域中標示個體、屬性、或關係；變數；以及標點符號，例如與其他符號作區分或群組之用的逗點與括號等。

- **語法 (Syntax)**：邏輯必須具備能定義如何結合符號，來形成具文法或合於語法的 (Well-formed) 句子的文法規則或形成規則 (Formation Rule)。這個規則可以是文字、圖形、或抽象語法。
- **語意 (Semantic)**：為能作出有意義的敘述 (Statement)，邏輯必須有理論作為參考，來決定常數與變數和所描述的事物的關係。此外，也必須有真實性 (Truth) 的理論，來區分敘述的正確與錯誤。
- **推論規則 (Rules of Inference)**：邏輯若要不只是一種標記方法，就必須能決定如何由另一種模式 (Pattern) 推論出另一個模式。若邏輯是合理的，那麼推論規則就必須能維持語意中的真實性。

根據 Sowa(2000)的觀點，筆者認為知識表徵語言的要素包括：邏輯、Ontology、及運算。如果欠缺邏輯，知識表徵將不具規則，同時也無法區分敘述的正確或錯誤，亦難以從事推論工作；如果沒有 Ontology，詞彙就不能被妥善定義和使用，概念將無法享有一致性；而若缺乏可運算的模型，電腦程式就無法處理邏輯與 Ontology，無法實際應用。

第三節 網路知識表徵環境、難題與發展趨勢

隨著網路的發展，人工智慧領域已發展的知識表徵方法面臨了新的挑戰，因此本節擬先探討網路知識表徵的環境、難題與發展趨勢。

一、網路知識表徵環境與難題

網際網路的重要影響之一是其有助於「資訊的分享」，在沒有網路之前，人們很難如此快速地分享資訊，不過在其帶來的便利性後頭，伴隨而來的是另一項有待克服的難題——「意義的分享」(此亦為「語意網」的願景之一)。資訊分享並不完全關乎資訊的獲取和累積，

而是還要能瞭解或利用已有的資訊。具體而言，網路對傳統知識表徵的假設帶來許多挑戰，因此需要以新的觀點來看待這個問題。以下分別說明網路帶來的三項重要衝擊：(Heflin, 2001)

- **網路是分散的：**網路普及的一項促因在其去中心化的特質，不過因為任何人皆可在網路上建置網站與提供資訊，且又缺乏編審或品質控制等機制，因此也就難以利用程式對當中的資訊進行推論，當然，網路資訊的可靠性也會受到質疑；再者，還會有同義詞（不同的詞彙表達相同的概念），或是同形異義詞（相同的詞彙表示不同的概念）等問題。於是，智慧型網路代理人（Agent）無法假定它所收集的資訊都是正確且一致的；此外，既然沒有全球通行的完善控管機構，因此不同資訊來源中的資訊，彼此間當然有可能有所衝突。
- **網路是動態的：**網路上的資訊以驚人的速度隨時在改變，即使是智慧型網路代理人也難以及時掌握；再者，由於網路資訊的快速變遷，因此也不易建立標準詞彙及正規的語意；此外，若主題領域的內涵有所改變，那麼所使用的詞彙也可能不同，語意也可能被修訂。
- **網路是大量的：**以 Google 為例，截至 2005 年 4 月 5 日止，其所索引的網頁數達八十億五千八百多萬筆。網頁的數量如此龐大，以致於對表徵語言的表達性，或推論語法的訂定上將有所侷限。

缺乏語意標記（Semantic Markup）是發展智慧型網路文件處理的主要障礙之一。目前的 HTML 標記主要是在處理文件的排版與呈現樣式，而非文件所承載的語意。隨著網路資源的快速成長，如何提昇發掘及獲取所需資源的效率便成為一項難題，當然，在網路資源的數量與日俱增的情況下，這項問題只會顯得更為棘手。人工智慧領域長久以來不斷發展結構知識與資訊的語言、方法、和工具，然而，若運用人工智慧領域的方法，來處理自然語言文件（半結構的）並不見得可行，因為在處理半結構的資訊時，又可能面臨四項難題：(van Harmelen & Fensel, 1999)

- **資訊搜尋**：現有的關鍵詞搜尋方式，透過文字比對會找出許多不相關的資源，例如會找出有相同字詞但卻有不同的意義，或是遺漏一些以不同字詞表示，但卻有相同涵義的資訊。
- **資訊擷取**：人們需從資訊資源擷取相關的資訊，進行瀏覽和閱讀，然而，自動化的代理人可能不具一些基本常識，來從文件擷取所需的資訊，因此無法整合不同的資訊資源。
- **一致性維護**：半結構的資源數量愈來愈多，要能維持它的一致性與正確性，實需透過機器來協助偵測一些不規則的情況。
- **文件自動產生**：機器需能處理資訊資源的語意，才有辦法從半結構的資料產生半結構的資訊。

此外，除了網路對知識表徵的衝擊外，亦需面對有關獲取知識，及知識本身的變動性的問題，茲說明如下：

- **多元獲取途徑**：數位圖書館或網站經常藉由主題階層式架構來整理文件（例如 Yahoo），使其易於瀏覽。不過，多數文件都會牽涉許多主題，因此經常導致文件重複存放，或限制使用者以單一主題的方式來獲取資源等問題；前者導致浪費資源與搜尋時間，而後者則是削弱了讓使用者找到符合其需求的資源的可能性。(Wheatley & Armstrong, 1997 as cited in Louie, Maddox, & Washington, 2003)
- **多元搜尋策略**：使用者會採用許多資訊尋求與檢索策略來找所需的資源，例如瀏覽、分析、與關聯等，但目前的資訊架構，多只支援部分的搜尋方式。(Fidel & Efthimiadis, 1999; Fidel, et al., 1999 as cited in Louie et al., 2003)
- **知識的複雜度**：真實的知識是多層次的，意義是綜合各種不同觀點而成，具有多元面向。例如知識間不同觀點的交集會具有高度相關性，但這些特徵卻很少應用到目前的資訊系統當中。(Broughton, 2001; Ellis & Vasoncelos, 2000 as cited in Louie et al., 2003)
- **知識的變動性**：真實的知識是渾沌的、缺乏組織、且經常改變的，文件本身也經常改變，它在社會中的意義同樣也會改變。若知識改變了，那麼在系統中獲取知識的途徑或結構，應具備調整的彈

性。(Pilsk, McIntyre-Colby, Andrew, & Wilson, 2002 as cited in Louie et al., 2003)

綜上所述，由於網路的資訊資源具有分散、易動、及成長快速等特性，使得知識表徵需克服多項難題，包含如何由分散的資訊來源有效搜尋和擷取資訊；如何對資訊進行一致性維護；資訊的自動處理；以及獲取資訊的途徑或結構，能隨環境和知識的改變而彈性調整等。為改善電腦處理上述問題的能力，電腦科學界一直不斷嘗試許多改進之道，W3C 在 2001 年更提出「語意網」這項願景，期望使電腦能真正有效輔助人類對資訊資源的處理與應用效率。

二、網路知識表徵的發展趨勢

網路資源於發展之初是以內容的呈現及超連結為開發重點，缺乏語意標記，因此無法讓電腦對網頁內容進行有意義的自動處理。以「超文件標記語言」(Hyper Text Markup Language, 簡稱 HTML) 為例，它是為了「呈現」資訊之用；也就是說，電腦雖可瞭解網頁「標題」(Header) 應顯示什麼，或某處的連結應連結到某個網頁等資訊，但卻無法處理文件所承載的語意，例如這是王小明的網頁，當中的連結是連到他的個人簡歷。此外，由於網路具有易於使用的特點，因此相當普及，但也正因如此，所以使用者在這種資源眾多且對資源內容缺乏語意標記的環境下，不易搜尋到相關的資源。

為使網頁能易於電腦處理，那麼勢必改善現行的網頁資訊處理方式；也就是說，網頁資訊應能為人類和電腦理解，不過電腦不需完全瞭解，而是瞭解某種程度的資訊即可；對資訊的理解，人類仍居要角，電腦是用來輔助人類理解資訊之用。使電腦嘗試理解資訊並進行有意義的自動處理，即為 2001 年起 W3C 致力推動的「語意網」所欲達成的目標，以下進一步說明其意義與設計原則。

(一) 語意網 (Semantic Web)

W3C 推動的「語意網」計畫的目標在建立一個全球性的資料交換媒介。(Miller, 2004)。目標便是發展能支援資訊發掘、資料整合、瀏覽、以及自動化處理的標準和技術，協助機器瞭解網路上的資訊資源。其特徵是以「標準萬國碼」(Unicode)與「統一資源識別碼」(Uniform Resource Identifier, 簡稱 URI)作為「語意網」的可讀性 (Readability) 和可定址 (Addressability) 之基礎；其上一層為「可擴展標記語言」(eXtensible Markup Language, 簡稱 XML)、Namespace、與「XML 綱要」(XML schema)，其中 XML 與 XML schema 可作為詮釋資料的語法互通基礎，而 Namespace 可協助詮釋資料之詞彙的語意互通與識別；第三層開始則是替全球資訊網注入語意，可利用 RDF 來標示資源與資源間的關係，並作為邏輯推論的基礎；再上一層的 Ontology 則可進一步對概念間的語意關係予以限定；最終，再透過一些推論與驗證機制或語言，例如規則標記語言 (Rule Markup Language, 簡稱 RuleML²)，讓軟體代理人能對資源進行自動且有意義的處理。(Berners-Lee, 2000)如下圖所示。

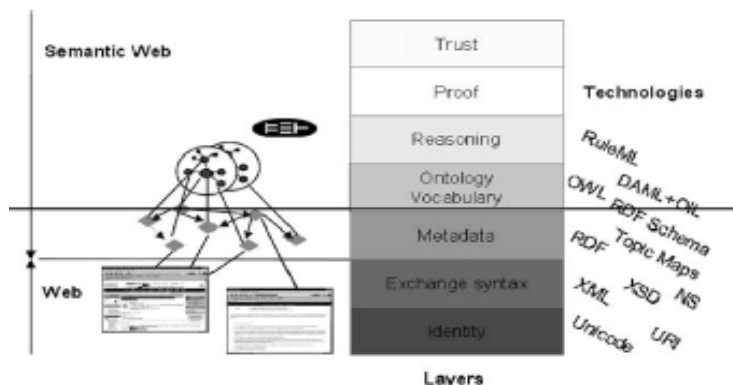


圖 2-5 語意網之語言階層

資料來源：Goble, C.(2003). The Semantic Web: An evolution for a revolution. Computer Networks, 42, 551-556.

² <http://www.ruleml.org/>

「語意網」是為網頁加入有意義的結構，建立讓軟體代理人能悠遊於網頁間，並替使用者執行複雜工作的網路環境。例如當軟體代理人到達某校的網頁時，將不只知道該網頁的關鍵詞（例如「教學」與「研究」），同時也會知道有那些教師在該校服務，以及開設那些課程等資訊；換言之，「語意網」不只可以幫助人們找到和主題有關的資料，還可從中推論出無法透過關鍵詞檢索，而直接找到的資料。(Powers, 2003)也就是說，「語意網」的運作仰賴讓電腦能獲取結構化的資訊，以及設定讓程式能執行自動推理的推論規則。(Takeda, 2004)

傳統的知識表徵系統是採集中式的 (Centralized) 運作模式，需要大家對共通的概念（例如雙親、交通工具）採用和分享完全相同的定義，不過這種集中式的作法，隨著資訊資源的快速增長，則顯得不足以有效處理當前資訊環境中之資訊。此外，傳統的系統會對問題加諸許多限制，好讓電腦能夠正確的提供答案，且傳統的知識表徵系統，一般都設有其自行採用的推論規則，即使資料能在不同的系統間傳遞，但推論規通常卻無法在其他系統中運作。「語意網」為能保有變通性，因此可以接受有問題無法回答的情況發生。(Berners-Lee, Hendler, & Lassila, 2002)

總而言之，「語意網」的挑戰在提供既可表達資料，又可對資料進行推論的語言，同時還能將現已發展的知識表徵方法的原則，適切地應用至網路環境。(Berners-Lee, Hendler, & Lassila, 2002)根據 Tim Berners-Lee(2000)的說法，RDF 是達成「語意網」的基礎之一，所以 RDF 的設計和「語意網」的發展有著緊密的關係，因此，以下進一步說明「語意網」的主要設計原則：(Koivunen & Miller, 2001)

1. 任何事物皆可藉由 URI 來指定 (Identified)

在真實世界中的人、地、物，都可藉由指定不同的 URI，來在「語意網」中加以表示，例如一位名為 “Tim Berners-Lee” 的這個人，我們可以用其電子郵件信箱的 URI 來指定。

2. 可對資源與連結定義類型 (Type)

目前的網路是由資源與連結所組成，如圖 2-6 的圖 a 所示，這些

資源通常不含詮釋資料，來說明其用途及其與其他資源的關係，雖然人類或許可以輕易瞭解資源與資源間的關係，但對機器而言卻不是件易事。對「語意網」來說，它雖同樣是由資源與連結所構成，但卻可對資源與連結加以定型，讓機器對資源可以多些瞭解，如圖 2-6 的圖 b 所示，例如指出某些資源是另一資源的另一個版本。類型通常由類型所指向的 URI 節點來定義。

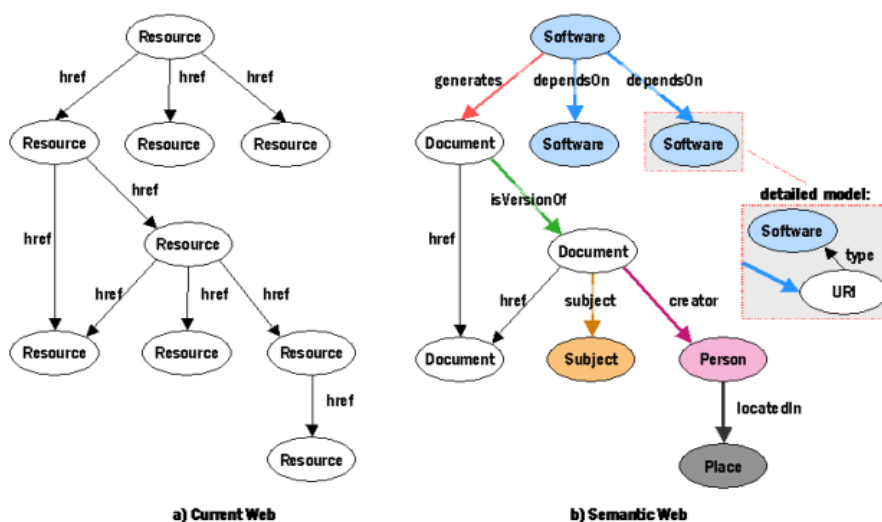


圖 2-6 資源與連結在語意網中可以定義類型

資料來源：Koivunen, M-R. & Miller, E(2001). W3C Semantic Web activity. Retrieved March 10, 2005, from <http://www.w3.org/2001/12/semweb-fin/w3csw>

3. 容許連結的網頁不存在

目前的網頁作者可以藉由超連結輕易地連結其他資源，不過目前缺乏當被連結的資源被移走時，告知連結者這項資訊的機制，因此當連結到不存在資源時，網頁便會顯示 404 的訊息。「語意網」也會面臨同樣的問題，但由於任何人都可對任何資源加以敘述，並對資源間的連結定義類型，所以若該情況發生，人們仍有可能從中獲得相關的資源。

4. 資訊不需絕對真實可信

真實或可信 (Truth or Trustworthiness) 應由應用程式在處理資訊

時自行評估，應用程式由敘述的情境 (Context) 來決定相信那些資訊，例如何人在何時具備何種憑據 (Credential) 說了什麼。以下圖為例，Tiina 為 Elisa 的員工，她想要造訪 W3C 的會員網頁，為能順利進入該網頁，她必須提供有權進入該網頁的證明 (Proof)，所以她会指向由 Kari 與 Alan 所定的四項敘述 (Kari 與 Alan 被定義有權制定那些敘述)。W3C 的應用程式會接收 Tiina 提供的證明，並由 Kari 與 Alan 所定的敘述，來驗證她是否有權進入 W3C 的會員網頁。

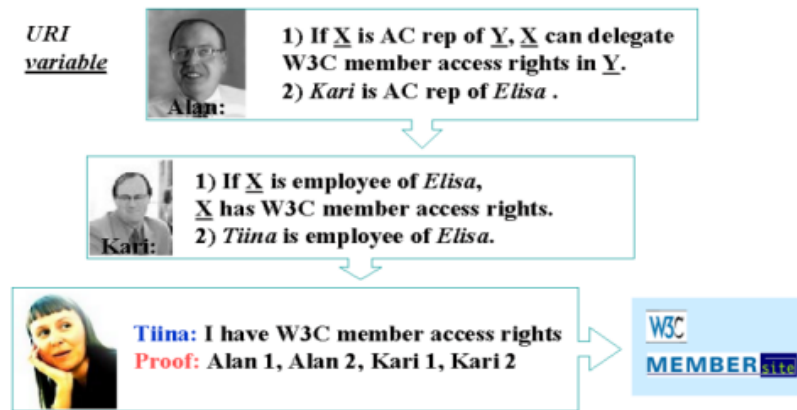


圖 2-7 定義造訪權利的驗證鍊 (the chain of trust)

資料來源：Koivunen, M-R. & Miller, E(2001). W3C Semantic Web activity. Retrieved March 10, 2005, from <http://www.w3.org/2001/12/semweb-fin/w3csw>

5. 便於結合相關資源

不同的社群在不同的地方，或同一社群在不同的時間，都可能對相似的概念加以定義，因此若能結合網路上和這些概念有關的資料，應是有幫助的。例如網路上存在姓為 Miller 的人的新舊資源，為能結合新舊資源，人們可以在新資源中加入屬性 "previousIdentity" 指向舊資源；人們也可以定義新舊資源所使用的詞彙之轉換機制，例如定義 "worksAt" 等於 "employedBy"。只要相關的應用程式找得到，新增的資源可以儲存在任何地方，然後藉由指向舊資源的 URI 來結合這些資源。

6. 極簡的設計 (Minimalist Design)

W3C 的「語意網活動」只對必要的事物標準化，同時也有為未來的複雜性作準備的研究在進行中。目前已有一些應用利用已存在的標準技術，例如 Dublin Core 與簡易聯合供稿系統 (Really Simple Syndication, 簡稱 RSS) 等來開發。總之，W3C 認為「語意網」技術的發展有著無限可能，也希望能結合已存在的標準技術來達成綜效 (Synergy)。

語意網語言 (Semantic Web Language, 例如 RDF 與 OWL) 能用來描述「無法」在網路上可獲取的資源。「語意網」是 W3C 的口號與願景，推動目標是為了改善網路資源的搜尋問題，現階段的發展主要在協助或促進網路相關應用系統，提昇對文件意義的感知與處理，例如能讓使用者知道文件與其他事物 (線上或非線上) 的關係。

(二) 資訊的可信度

朝「語意網」發展的最後階段，需要電腦進一步從事資源間的推論與驗證。知識表徵研究傳統上在正規邏輯 (Formal Logic) 的影響下，認為知識應是一致且完整的，在此前提下，資料和知識實質上差別不大。不過由於網際網路的開放和分散等特性，因此人們難以掌握所有的資料或知識，也無法要求資料或知識總是具有一致性，於是，資訊的可信度 (Trust) 便成為「語意網」重要的議題之一，可信度並不是單純地指可相信的程度，還會牽涉到人類事物 (就好像人際網路) 的複雜性。所以，資料與知識的可信度乃知識表徵的一項新的挑戰。(Takeda, 2004)

若考量知識創造的過程，Takeda(2004)認為可將「語意網」視為全球性的知識創造實驗，因為「語意網」提供大眾知識表徵語言，期盼不同領域的使用者能建立與散佈該領域的知識。早期知識庫的開發都是在機構內部，由一群專家來建構，而這種知識庫只提供部份人士使用並需持續維護；相反地，「語意網」則提供了讓每個人都可參與知識庫建構的過程，這會是人們共同合作建構知識的過程。

第四節 電腦科學界的網路知識表徵方法

過去的人工智慧知識表徵方法，主要是處理在相同儲存環境中已結構化的資料；然而，網路資源則通常具有缺乏結構、多樣、與動態等特性，表徵的方法主要透過標記語言 (Markup Language)，讓人們對內容資訊予以標記。以下，首先簡要說明目前主要用以標記網頁資訊之「超文件標記語言」與 XML 的優點和限制；接著簡單敘述 RDF 與 Topic Maps 之基本概念。

(一) 超文件標記語言 (Hyper Text Markup Language, 簡稱 HTML)

要使資訊可被機器處理 (Machine-processable) 那麼資料就須具有結構 (Structure)。有許多方法可將資料結構化，以資料庫為例，是將資料儲存在有屬性的表格，而在網路上將資料結構化的主要技術則是「文件標記」，也就是在文件中的特定序列 (Sequences)，包含了指出如何作用文件內容之資訊。標記不只描述文件資料的輸出 (Layout) 與邏輯結構，同時也讓資訊在某種意義上自我描述 (Self-describing)，這種標記通常是以標籤 (Tags) 的形式表示，例如 或 <title>。(Ding, Fensel, Klein, & Omelayenko, 2002)

HTML 是由 W3C 制定，最早版本 HTML 1.0 是在 1993 年時公布，目前最新版是在 1999 年 12 月時所公布的 HTML 4.01。HTML 的優勢包括：(明寰資訊，民 91)

- **易於編寫**：主要原因在其標籤數量不多，易於學習；標籤也都是固定的，因此使用者無法自訂標籤。
- **跨平台**：在不同的電腦，只要安裝相同的瀏覽器，應該就可以看到相同的網頁顯示結果。
- **資訊呈現能力**：HTML 的功能主要是為了顯示網頁內容資訊，方便使用者表現圖案和文字。

雖然 HTML 對網際網路的蓬勃發展有一定的貢獻，但使用者卻無法從 HTML 標籤得知標籤間夾帶的資訊內容。因此，就描述 Ontology 來說，HTML 主要具有以下幾點限制：(梁中平、徐千惠，民 86，轉引自高顯霖，民 92)

- **缺乏延伸性 (Extensibility)：**HTML 標籤不具可延伸性。HTML 是一種特定用途的標記語言，無法根據不同領域的應用需求，讓使用者自訂需要的標籤，此種不能延伸的限制使其無法被應用於描述 Ontology 概念，因為 Ontology 概念需要豐富且富有彈性的字彙 (標籤) 來描述。
- **缺乏結構性 (Structure)：**HTML 缺乏嚴謹的結構。欲描述 Ontology，除了需有豐富的字彙外，更重要的是能定義字彙間的關係，而 HTML 在此點更顯薄弱，因此 HTML 無法完善的描述 Ontology 概念。
- **缺乏語意性 (Semantics)：**HTML 的標籤無法反映出夾在標籤間的資料內容，此限制對邁向「語意網」是一項嚴重的障礙。雖然使用者能夠輕易地透過瀏覽器來閱讀 HTML 所編輯的文件內容，但對軟體代理人而言，卻無法從 HTML 標籤得知文件的相關資訊，所以很難對文件進行自動或有系統的分類整理，以及處理使用者的相關要求。

「標記語言」會指定可以使用的標記標籤、標籤如何使用、以及標籤的意義。HTML 即為網路上最廣為人知及廣為應用的「標記語言」，例如與表示的是清單項目，瀏覽器會利用這些資訊來顯示網頁。不過 HTML 只提供如何呈現 (Render) 內容的資訊，而這樣的資訊不足以協助「語意網」所預達到的有效自動處理服務，因此應用程式需要另一種能指定文件內容的特定部分之作用的「標記語言」，例如對比較價格的應用程式來說，若所有價格都以<price>的標籤來註記 (Annotate) 的話，將能有效協助這方面的自動處理；換言之，若能指定標籤的意義就顯得相當重要。(Ding, Fensel, Klein, & Omelayenko, 2002)

簡言之，HTML 無法有效支援 Ontology 的描述，僅稱得上是一種資料的「呈現」語言；同時，HTML 也面臨資訊複用、資料交換、自動文件處理、以及無法支援較精確的查詢等限制 (ArborText, n.d., 轉引自陳嵩榮, 民 88)。由於上述限制，因此便有 XML 的發展，希望透過 XML 來補 HTML 不足之處，以下進一步說明之。

(二) 可擴展標記語言 (eXtensible Markup Language, 簡稱 XML)

XML 是一種語法上看起來很類似 HTML 的文件標記語言，都是使用成對的標籤來標示一份文件的內容。但是，XML 的目的並非用於標示一份文件的排版與呈現風格；而是用來描述一份文件的內容 (Content) 與結構 (Structure)。(林光龍, 葉建華, 歐陽彥正, 民 91)

XML 是「標準通用標記語言」(Standard Generalized Markup Language, 簡稱 SGML) 的一個子集合，在功能上它去除了 SGML 的複雜部分，同時彌補了 HTML 應用範圍的侷限。1998 年二月中旬 W3C 公布了 XML 1.0 版，當中明白地指出 XML 本身並不是一個程式語言；而是一種詮釋語言 (Meta-language)，一個可用來定義其他語言的語法。

使用者可以依據 XML 1.0 版中所規定的「文件型別定義」(Document Type Definition, 簡稱 DTD) 規則，自行定義構成該份 XML 文件的元素 (Element) 架構、元素標籤、和屬性。DTD 是一套建構 XML 文件標籤的語法規範，使用者可以透過建立 DTD，定義一套符合自己需求的 XML 文件。定義一份 DTD 文件，也就相對應的規定了一份 XML 文件中，可以出現那些標籤、標籤的使用順序、那些標籤可以包含子標籤、每一個標籤可以有那些屬性等，所有這類的資訊與規則都定義在 DTD 文件中，所以 DTD 可以是 XML 文件本身的一部分；但是也可以單獨成為另一份文件，再經由引用 (Include) 的程序與 XML 文件結合。(林光龍、葉建華、歐陽彥正, 民 91) 一般而言，XML 具備以下優點：(高顯霖, 民 92)

- **可延伸性**：XML 具備了可擴展的特性。其標籤是由使用者自行訂定，因此使用者便可根據所欲描述的 Ontology 來訂定相關的字

彙（標籤），透過自訂的 XML 標籤，使用者可以瞭解標籤間所涵蓋的資訊內容。

- **資料內容與呈現方式分開：**XML 文件的內容和呈現方式是獨立的，因此增加了資料交換時的便利性。應用程式只要瞭解該 XML 的 DTD 或綱要（Schema），便可輕易地存取和交換資料。若要將 XML 以不同的形式呈現在瀏覽器上時，則可使用「延伸式樣規語言」（eXtensible Stylesheet Language，簡稱 XSL）或「串接樣式表」（Cascading Style Sheets，簡稱 CSS）來作不同的呈現設計。
- **可使用 DTD 或 Schema 來定義標籤屬性和文件結構：**XML 文件可藉由 DTD 和 Schema 來定義 XML 文件的規則和架構，例如元素的個數、出現的順序、相關屬性、階層架構、和水平關係等相關資訊。如此一來，便有助於描述基本的 Ontology 概念本身的屬性和概念間的關係。

XML 作為一種資訊標記的方法，雖具有上述優點，及有助電腦的自動化處理，並可描述文件的結構和幫助資料交換³，但 XML 卻也面臨了無法指出結構的意涵，也就是無法讓電腦透過標籤的名稱進而瞭解當中的語意。XML 提供的只是「表面模型」（Surface Model），一種樹狀結構而已。因此，對「語意網」來說，無法透過 XML 來承載語意，它只是一種「傳遞資料的或資料交換的格式（Format）」；換言之，XML 只是一種易於為機器處理的資料格式。（Stojanovic, Staab, & Studer, 2001）

因此，若要於網路上架構出能表徵知識，同時亦有助電腦自動推論的機制，就必須在其他方法上彌補 XML 無法提供語意交換的限制，而 RDF 及 Topic Maps 即是可為 Ontology 提供語意階層互通的資料模型，並可用於表達主題或概念間關係的方法。以下即分別對 RDF 與 Topic Maps 予以簡單說明。

³ 上述事件之所以能達成，就在於遵守已定義好的標記（markup）方式。

(三) RDF

XML 是將資料結構化的標準語法，而 RDF 則是對資源進行敘述的資料模型。RDF 如其名稱，它不是語言，而是在網路上表徵資源的模型。(Ding, Fensel, Klein, & Omelayenko, 2002)RDF 是由 W3C 制定和公布，它是一種在網路環境中，可用來標示資源與資源間的關係，並可作為邏輯推論之基礎之方法。RDF 規範最早的版本是在 1999 年 2 月 22 日公布⁴，目前最新的版本是在 2004 年 2 月 10 日所公布⁵。

RDF 所表示的任何敘述結構皆由「三元素」(Triples) 組成，其關係如圖 2-8 所示。RDF 可用來表達網路上資源和資源間之關係，此「三元素」分別為：(Klyne & Carroll, 2004)

- **主詞 (Subject)：**資源
- **述語 (Predicate)：**屬性，也就是資源的特性或特徵
- **受詞 (Object)：**屬性值，就是屬性的內容



圖 2-8 RDF 基本概念示意圖

所有以 RDF 來描述的事物皆可稱為「資源」(Resource)，例如網頁、文獻、與人等，因此也就表示，「三元素」中的「主詞」和「受詞」皆是可透過 URI 來表示的資源，不過「受詞」亦可為純文字字串。為能區別不同的「述語」，因此「述語」同樣也可透過 URI 來表示，例如「<http://www.w3c.org/2001/sw/>是由 W3C 建立 (Created By)」，可表示為"<http://www.w3c.org/2001/sw/>",Created By,"W3C"，而本例的「建立」(Created By) 一詞剛好可以用 Dublin Core 的詞彙 "<http://purl.org/dc/elements/creator>" 來標示，所以上方的敘述可以改成"<http://www.w3c.org/2001/sw/>",<http://purl.org/dc/elements/creator>,"W3C"。

⁴ <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

⁵ <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

雖然「三元素」的表示方式相當有彈性，但 RDF 要求「三元素」必須由「主詞」指向「受詞」，如此一來，才能降低機器解讀的困擾；(Nilsson, 2001)換言之，透過三元素的資料模型，可以允許人們由知識表徵的觀點來繪製一個有向圖 (Directed Graph)，並蘊含某種程度的語意在其中。(Klyne & Carroll, 2004)

此外，RDF 以 URI 來標示 Ontology 內的詞彙，可為該詞彙提供唯一性，這麼一來，Ontology 開發人員便可針對某詞彙提供專屬定義（即使這個詞彙已有別的開發者定義過，或是引用別的開發者的詞彙延伸而來）。RDF 基於這個基礎，將能以機器可讀 (Machine-readable) 的方式「敘述任何事物」(Say Anything About Anything)。(Nilsson, 2001)

(四) Topic Maps

Topic Maps 原本是為了提昇索引 (Index) 方法而發展，於 2000 年初期時成為 ISO 國際標準 (ISO/IEC 13250:2000⁶)，並在 2001 年時由 TopicMaps.Org 組織⁷將 ISO/IEC 13250:2000 應用於全球資訊網，制定公布了 XTM v1.0 規範。Topic Maps/XTM 規範提供一呈現資訊資源之結構的模型及文法 (Grammar)。

Topic Maps 有三個基本核心概念，分別為「主題」(Topic)、「關聯」(Association)、與「資源指引」(Occurrence)。「主題」是指在主題地圖中所欲表達的事物，例如哈姆雷特這個劇本或莎士比亞這位劇作家；「關聯」的作用是能將相關的主題關聯在一起；而「資源指引」，則負責連結和主題有關之資源，(Pepper & Moore, 2001)以圖 2-9 為例，具體說明 Topic Maps 的三個基本核心概念：

⁶ ISO/IEC 13250:2000 Topic Maps: Information Technology -- Document Description and Markup Languages, Michel Biezunski, Martin Bryan, Steven R. Newcomb, ed., 3 Dec 1999.

⁷ TopicMaps.Org 是於 2000 年成立之獨立團體，透過應用 XML 相關規範，發展使 ISO13250 能適用於全球資訊網之主題地圖規範。TopicMaps.Org 的官方網站：<http://www.topicmaps.org/>；XML Topic Maps (XTM) v1.0 Available at: <http://www.topicmaps.org/xtm/1.0/>

- **主題**：每個主題皆扮演具體化某抽象主題（Subject）的數位替代品，例如本例將 Puccini 這個人轉為符號具體表現在電腦中，且每個主題可能會有一或多個名稱（Names）。本例的主題為 Puccini、Lucca、Tosca、與 Madame Butterfly。
- **關聯**：相關的主題能透過關聯而將彼此關聯在一起。本例的 Puccini 這個主題，透過 “born in” 而和另一主題 Lucca 關聯在一起；此外，Puccini 這個主題，還透過 “composed by” 與另二個主題 Tosca 及 Madame Butterfly 關聯在一起。
- **資源指引**：由主題連向某資訊資源的連結。本例的資源指引以虛線表示。

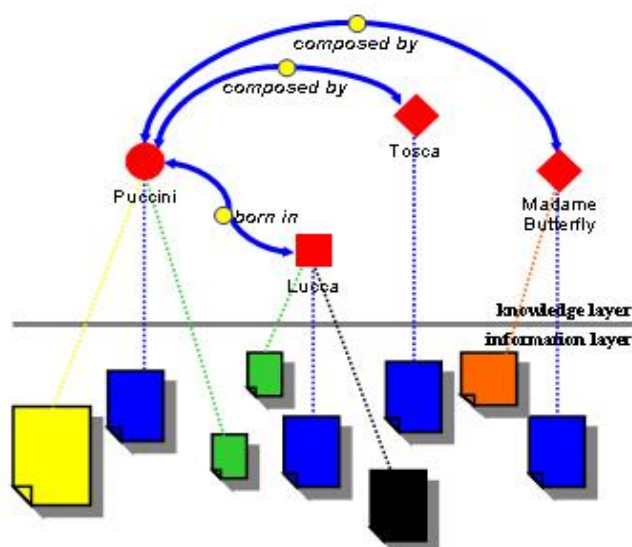


圖 2-9 主題地圖基本概念示意圖

資料來源：Pepper, S.(2003). ISO 13250:2002—Topic Maps: An international standard knowledge representation for humans and agents. Retrieved Aug. 11, 2004, from <http://www.csc.liv.ac.uk/~valli/WG-Ontology/Helsinki/Id3-Ontopia.ppt>

主題的「特徵」(Characteristics) 有三，分別是：主題的「名稱」、主題的「資源指引」、以及主題在關聯中所扮演的「角色」(Role)；此外，主題、關聯、以及資源指引，皆可透過其他主題加以定訂其類型 (Type)。主題在「範圍」(Scope) 內有其特徵；也就是說，在範圍

內之特定情境下的名稱與資源，才會被視為是該主題的名稱、資源、與關聯。不同的 Topic Maps 亦可加以「合併」(Merge)。「合併」是由 Topic Maps 作者在建立 Topic Maps 時即給予指示，或在執行中經過使用者或應用程式的判斷而發生。一或多個使用並遵守此文法且相互關聯的文件，即稱為 Topic Maps。(Pepper & Moore, 2001)總而言之，Topic Maps 可說是將主題、關聯、及資源三者，運用主題索引的概念及網路的特性加以結合，提供新的知識組織模式，讓使用者可以定義複雜的知識架構，創造虛擬的知識地圖；同時，也是能幫助使用者能有效的瀏覽資訊資源的一套標準方法。

(五) 小結

綜上所述，雖然透過 XML 可以用來彌補 HTML 之不足，用來描述一份文件的內容與結構，但 XML 仍舊面臨了無法指出結構的意涵，也就是電腦無法透過標籤的名稱，進而瞭解當中的語意等問題。若要在網路上架構出能表徵知識，同時亦有助於電腦自動推論的機制，就必須在其他方法上彌補 XML 無法提供的語意交換機制，而 RDF 及 Topic Maps 即是可為 Ontology 提供語意階層互通之資料模型，並可作為承載語意之知識表徵的一套語法規範（即 RDF/XML 與 XTM），不過這兩種方法起初在發展的目的並不相同，它們是為了解決不同問題而發展出來的不同方法，它們是兩個不同的語法家族。(Garshol, 2003)因此，本研究希望對它們在表達概念或主題關聯之語法、所能表達的語意、及應用情境等問題進行分析比較，嘗試瞭解其異同。

第五節 網路知識表徵語言之比較研究

本節將檢視目前有關網路知識表徵語言之比較研究，嘗試瞭解相關研究所採用的分析比較標準，作為本研究在設定比較項目之參考。

Gil and Ratnakar(2000)嘗試比較 XML Schema(2001)、RDF Schema(2000)、與 DAML+OIL(2001)⁸三種標記語言之表達性，該研究的價值除了在獲得比較所得的結果外，值得一提的是作者提出了八項分析面向，其所提出的分析要項與比較結果如表 2-2 所示。

表 2-2 XML Schema、RDF Schema 與 DAML+OIL 之比較

分析面向	細項	XML Schema(2001)	RDF Schema(2000)	DAML+OIL(2001)
概念能否隨情境之不同而有不同解釋	情境 (Context)	可以	可以	可以
類別(Class)	表達物件類別與屬性	無(類別可能是任何元素，且其屬性亦可能為其子元素)	可以(rdfs:Class & rdfs:Property)	可以(adml:Class, daml:ObjectProperty, daml:DatatypeProperty)
	具繼承性	無	類別與屬性皆可 (rdfs:subClassOf, rdfs:subPropertyOf)	類別與屬性皆可(利用 RDF 語法)
屬性/元素限制	屬性/元素的值域	可以(區域與全域皆可)	可以(僅適用於全域)	可以 (全域 rdfs:range)

⁸ DAML 全稱 DARPA Agent Markup Language；OIL 全稱 Ontology Interchange Language。DAML+OIL 是基於 XML 與 RDF 而發展的語言，同時具有正規描述邏輯 (description logic, 簡稱 DL) 及比 RDF Schema 更豐富的 Ontology 建模原詞。DAML+OIL 希望藉由擴展 RDF Schema 的描述能力來提昇自動推論的能力。

				daml:Restriction, onProperty, toClass)
	屬性/元素的主題領域	可以	可以(僅適用於全域 rdfs:domain)	可以(僅適用於全域 rdfs:domain)
	屬性/元素的基數 (cardinality)	可以(僅適用於區域 minOccurs, maxOccurs)	無	可以(區域 minCardinality, maxCardinality, cardinality) (全域 UniqueProperty 或限制為"#Resource"的子類)
資料類型與實例 (Instance)	基本資料類型	可以(數值、日期、與字串資料類型)	可以	可以(可以使用 XML Schema Datatypes)
	屬性/元素值的列舉	可以 <enumeration>	無	可以<daml:one of>也可以指向 XML Schema 的列舉資料類型
	實例	可以	可以<rdf:ID>	利用 RDF 語法
資料集 (Data Sets)	有界清單 (Bounded Lists)	無	無	可以<daml:collection>
	有序 (Ordered) 資料集	可以(由預設值來維護資料集順序)	可以<rdf:Seq>	可以<daml:list>
否定、互斥、與連結	否定 (Negation)	無	無	可以<daml:ComplementOf>
	互斥類別	無	無	可以 <daml:disjointUnionOf> <daml:unionOf>
	連結類別	無	可以 <rdf:subClassOf>	可以 <daml:intersectionOf>

定義	必要、充分 及有條件的 資格	無	無	可以 <daml:sameClassAs> <daml:UnambiguousProperty>
屬性的類別	反屬性	無	無	可以<daml:inverseOf>
	遞移屬性	無	無	可以 <daml:TransitiveProperty>

資料來源：Gil, Y. & Ratnakar, V.(2000). A comparison of (semantic) markup languages. Retrieved December 22, 2004, from <http://www.isi.edu/expect/web/semanticweb/paper.pdf>

透過比較可以發現上述三種語言中，RDF Schema 和 DAML+OIL 能夠表達某種程度的語意，但 XML Schema 則不能。若缺乏語意那麼自然也就無法透過標準的方法，去解譯語言中所使用的構詞，及完成一致的推論工作。DAML+OIL 植基於描述邏輯 (Description Logics) 而發展，它具有許多 XML 和 RDF Schema 所沒有的特性，但在應用上也相對複雜許多。因此作者認為雖然 XML Schema 無法承載語意，但它較容易應用，因此同樣能有助於「語意網」的內容和工具的發展。

Gil and Ratnakar 的研究偏重語法能力的比較，且所比較的三者 (XML Schema、RDF Schema、與 DAML+OIL)，事實上本來就是為了解決不同層次的語意問題，因此所得的分析比較結果，自然以 DAML+OIL 能滿足的項目較多。雖然其提出的分析比較項目對本研究具有一定的參考價值，但其缺乏對應用情境、資訊檢索、資訊過濾、與資源整合等問題之比較。因此本研究除嘗試比較 RDF 與 Topic Maps 之語法與所能表達之語意外，亦將應用情境、資訊檢索、資訊過濾、與資源整合等面向納入比較。