以本體技術爲基礎的知識庫建置程序及其應用

The Ontological Techniques for Building Knowledge Bases: Its Procedures and Applications

戚玉樑
中原大學資訊管理研究所
桃園縣中壢市(32023)中北路 200 號
maxchi@mis.cycu.edu.tw

摘要

本研究探討如何藉本體技術(Ontological techniques)來建置知識庫，在本體知識庫的發展上，我們建議須分別依照規劃、設計、測試修正、佈署及整合擴展等五項階段進行，由於本體知識庫主要是由概念架構及其關係所構成，因此設計階段是其成效的重要關鍵。爲收集對真實世界的認知，我們利用正規概念分析法來獲得所須的概念及其關係，並發展常用的邏輯模式，以降低知識表達的轉換障礙。本研究利用上述發展程序並結合 Ontology 開發工具，以植物學門的維管束植物爲例，進行實際應用上的驗證。由結果顯示，本研究所提出的發展程序，特別是設計階段採用的方法，確可協助建置者加速本體知識庫的建立。

關鍵詞：本體論、知識庫、描述邏輯、正規概念分析法、推論。

Abstract

For ontology builders, creating suitable conceptual structures of specific domain is increasingly problematic. Though humans have been building ontologies for years to express subject area of interest, an efficient ontology building process is very much ad-hoc. This paper describes development procedures that comprise planning, design, testing & modification, deployment, and integration. This paper particularly emphasizes the design procedure that involves utilizing Formal Concept Analysis (FCA) to gain conceptual structures and Description Logic (DL) to represent concept relationships in expressions. The paper provides a walkthrough example that utilizes Vascular plants as an applied domain. The empirical findings indicate that development procedures have advantaged in guiding ontology builders to create ontological knowledge bases step by step. Moreover, the cognition extraction is useful and connectable for builders and other tools.

Keywords: Ontology, Knowledge base, Description logic, Formal concept analysis, and Inference.

一、 前言

本體論(Ontology)原爲哲學上探究萬事萬物並加以歸納分析的學說，由於可用在資訊科技中的知識推論功能上，因此在人工智慧或專家系統中早有應用的相關研究。近年來，隨著網路科技的發展，特別是 XML 技術的成熟，以本體論爲基礎的知識庫應用，更擴展到各式各樣的實務作業中，例如 Kim et al.(1999)在品質管理的問題上，利用 Ontology 的分類特性，追蹤及確認問題發生的根源及影響原因。Sugumaran and Storey (2002)認爲本體方法能分析實際的需求，是設計資料庫的好方法，他們並提出一套包括建置、使用、及管理 Ontology 的方法論；Jiang et al. (2003)爲了分析病歷表特徵，結合 Ontology 及自然語言處理，將大量的文字資訊予以有效的分類；Knublauch et al.(2004)爲克服資訊的查詢是以關鍵字爲主的困境，因此利用 Ontology 建立語意架構，促使資源間的連結，獲得更開放的查詢架構；Tamma et al. (2005)利用 Ontology 建立規則協定的知識庫，做爲多代理人在協商時的依據，獲得更彈性的協商環境；其他如 Web 網頁註記、電子協商、及智慧型代理人等均可發現應用 Ontology 的研究。

在資訊應用的研究中，Ontology 一詞有許多不同的定義，其中 Gruber (1995) 對 Ontology 的定義 "Ontology is a specification of conceptualization"最常被引用，因此 Ontology 即爲利用將特定事物的概念化，來統一詮釋對它們的認知。而近年來的研究更直接將知識以 Ontology 表達，並利用 XML 格式呈現，這種本體知識庫即爲目前知識應用系統常見的型式之一。Hui and Yu (2005)及 Gillam et al. (2005)都曾指出建置本體知識庫的良莠，直接關係到知識系統的績效，因此須謹慎的實施。由於建置本體知識庫是將大眾的認知予以一致化，並須呈現於資訊的格式，因此建置 Ontology 是一項工藝(或藝術)遠勝於技術的工作(Noy and McGuinness, 2001; Van der Vet and Mars, 1998)，這也意謂著本體知識庫的建置過程充滿不確定性及需要大量的時間及人力支援。

本研究之目的即爲探討如何將人爲的抽象認知，轉化爲具象的概念陳述，它包括知識工程人員如何獲取領域知識及如何將知識轉換並呈現於資訊系統中。本研究提出之本體知識庫的發展程序，建議須依五項階段進行，包括規劃、設計、測試修正、佈署及整合擴展等，由於設計階段是重要關鍵，我們特別利用正規概念分析法(FCA, Formal concept analysis)解析人爲的認知至知識的概念架構，再以描述邏輯(DL, Description Logic)進一步的呈現。爲降低使用描述邏輯的障礙，我們將常用的邏輯類型予以模式化，以提供建置時的依據，並可提供其他發展建置本體知識庫的參考。另外，本研究將上述的發展程序實際應用於自然科學博物館的知識體系建置，並以植物學門維管束植物爲先期實驗計畫，藉以評估成效並做爲後續發展的參考。在實證中，本研究採用 W3C 組織所公佈的 OWL(Ontology Web Language)做爲本體知識庫的內容格式、以 Protégé 及 OWL plugin 爲編輯工具、及利用 Racer 爲知識的推論工具。

本研究的內容安排如次：第二節針對知識庫呈現方式及 XML-based 的本體論等做相關研究的回顧。第三節是探討本體知識庫的發展方式，並具體建議一個發展程序。第四節是有關於描述邏輯的模型與應用，我們將常用的模式予以歸納爲通式。第五節是有關於本體知識庫的開發實證過程，我們建置了維管束植物的本體本研究。最後，第六節討論應用的結果與經驗分享。

二、 文獻探討

(一). 知識庫呈現方式

知識呈現(Knowledge representation)可回溯至 Feigenbaum (1977)所提出的專家系統觀念，他認爲系統效能是來自於專家對知識呈現的精確程度，因此推論的成效依賴不同的呈現方式。爲使推論能具有邏輯推理的能力，必須建立知識之間的關係，因此也有不同的知識呈現方式。依據 Lehmann (1992)及 Minsky (1995)的研究，常用的知識呈現有法則式、邏輯式、語意網路、框架式、及本體論等方法，說明如下：

- 法則式(Rule-based)，通常用於淺層的知識表達，並以 IF-Then 的通式表達，例如 *IF (前提/狀態) Then (結論/動作)*。
- 邏輯式(Logic-based)，經多年的發展分爲命題 (proposition)、述詞 (predicate)、及模糊 (fuzzy)等邏輯方式。
- 框架(Frame-based)是模擬人類思維的模式，將事物之間的關係以階層化的資訊結構存入，形成一種層次化的知識庫。
- 語意網絡(Semantic net)，它是模擬人類記憶結構的模型，以網絡來表達人類知識構造，常用於自然語言的研究。
- 本體論(Ontology)是對特定領域進行概念及關係定義，藉推論程序獲得隱性知識的建構，因此較能貼近對真實世界的認知。

(二). XML-based 的本體方法

網際網路上的知識庫應用，是以整合、再利用、及語意瞭解爲主要研究議題。自 XML 的問世後，Ontology 的呈現格式也走向 XML-based 方式，但由於 XML 僅支援標記間的資料交換及有限的階層，對於語意或進一步的描述則嫌不足，因此須要其他規範的填補。Decker et al. (2000)曾指出，資源描述架構(RDF, Resource Description Framework)及 RDF Schema 是 W3C 企圖解決語意問題的先導規範，它允許使用者建立階層式的概念及屬性，因此具有 Ontology 的雛型。RDF 的後繼者大多是以建立資料的階層及分類爲基礎，例如醫療資訊界以發展 XOL(XML-based Ontology exchange Language)來做爲本體知識庫的交換標準；美國馬里蘭大學所研發的 SHOE (Simple HTML Ontology Extensions)，是以改善目前網頁資訊表達的限制，而將現有 HTML 擴充爲具有表達分類及階層的功能，使得網頁的連結能以語意來認知(Horrocks et al, 2003)；美國國防部的 DARPA 研究機構，根據 RDF 推出下一代的 DAML (Darpa Agent Markup Language)，隨後結合 OIL (Ontology Interchange Language)成爲具有推論機制的 Ontology 技術(McGuinness et al., 2001; Dieter et al., 2001)。網際網路標準組織(W3C)爲統一 Ontology 標準的發展，在 2003 年公佈了 OWL 爲正式規範，並區分爲 Lite、DL 及 Full 等三種版本，其中 DL 及 Full 支援推論機制，而各種支援工具亦逐漸成形，使得 OWL 成爲目前 Ontology 中最重要的 XML-based 規範(López de Vergara et al., 2004)。以上相關的本體方法規範可參考表 1 獲得進一步的資訊。

表 1: 本體方法相關規範

規範名稱	網址
RDF/RDF Schema	http://www.w3c.org/RDF
XOL	http://www.ai.sri.com/~pkarp/xol
SHOE	http://www.cs.umd.edu/projects/plus/SHOE
DAML/DAML+OIL	http://www.daml.org
OWL	http://www.w3c.org/2004/owl

三、 本體知識庫的發展方式

建構一個特定領域的本體知識庫往往是見仁見智的問題，因此也成爲導入 Ontology 技術最大的挑戰之一。舉例而言，如果請兩組由專家及工程師所組成的團隊，分別在沒有預設的前提下，對大學的「學系」來建構本體知識庫，其 Ontology 內涵會有差異，例如某個團隊是由「人員」組成的角度切入，因此以教、職、學生等來發展其下的概念階層及關係。另一個團隊則可能是以「組織」的架構爲觀點，因此以學院、學系、班級、實驗室等來發展 Ontology。在 Van der Vet and Mars (1998)的研究中即曾建議：建置一個本體知識庫，沒有對錯及唯一性的問題，雖然本體知識庫是大眾對特定領域的共同認知，因此「特定」也可以進一步詮釋爲某類問題下的知識，但在邏輯上我們不可假設 Ontology 是唯一的。

Ontology 是由知識的概念定義、屬性、實體、及關係的集合體，建構這些元素則需要一套發展程序，在過去的研究中曾將開發的程序稱爲本體工程方法(Ontology Engineering)，例如 Noy and McGuinness(2001)曾對開發本體知識庫提出由決定範圍到建立實例等七項步驟，而 Uschold and Grueninger(1996)亦曾建議名爲 TOVE 的本體工程方法，它包含了由動機到測試等六項程序。爲有效進行本研究的開發，本研究綜合了前述研究，並建議修正爲下列五項階段：

1. 規劃：建構團隊應至少包含領域專家、工程師、及使用者，並由團隊共同策劃準備工作。團隊成員必須明白，Ontology 不易符合全部的知識表達，因此沒有正確與否的疑慮，只有是否解決問題的要求。此階段類似軟體開發的需求階段，但宜以問題導向方式界定出需求，例如：
   - 領域的含蓋範圍是什麼，是否有現存的 Ontology 可再利用？
   - 是否有標準、公約、或習慣可供參考？
   - 其他潛在的結合對象是什麼？包含匯入(import)及輸出(export)的對象。
   - Ontology 的使用對象是誰？他們將怎麼使用？

   對於 Ontology 的細部規劃，建議以訂定不同類型的問題及答案爲含蓋範圍。在 TOVE 的後續研究中，Lin et al. (1996)及 Fox and Gruninger (1998)均建議 Ontology 的工程方法可利用「問題-答案」的配對方式(稱爲 Competency Question)來協助規劃。例如我們可以列出「生長在低海拔且是單子葉的植物是什麼？」或者「適合住家的栽種的蕨類植物是什麼？」等問題，藉著思考答案的方式，逐漸形成 Ontology 的架構雛型。
2. 設計：本階段是將規劃的架構予以逐步轉換爲 Ontology 的相對元素，亦即

- 概念(或稱爲 class/set/concept)：代表 Ontology 中對某類實體的集合或概念。
- 屬性(或稱爲 property/slot/role/relation)：代表 Ontology 中實體與實體或概念與概念之間的關係。
- 實體(或稱爲 Individual/object/instance)：代表 Ontology 中的個別真實例子。

爲協助領域專家將認知轉換爲細部設計，宜借重各種資料分析方法，例如本研究利用正規概念分析法(FCA)來釐清複雜的概念及屬性關係，FCA 是衍生自格狀理論(lattice theory)，下圖是以 FCA 尋找植物概念的應用說明：圖 1a 是由左側縱軸的 Objects 組成，橫軸是初步列舉的 Attributes，若兩者具備有關係時，則在相對位置註記符號。圖 1b 是以格線圖來展開關係架構，例如其中的槭樹科及松科暫時視爲相似的概念。圖 1c 是加入喬木、灌木的屬性來區分，但由於槭樹科同時擁有兩項屬性，因此仍無法完全與松科區分。最後我們將種子再細分爲被子及裸子，圖 1d 顯示各項標的物均已完全分開，表示它們擁有其獨特的概念。在這些顯性關係定義後，再利用數學的計算方式，協助領域專家進一步確認其他未定義的關係。

	種子	孢子	草本	木本
松科	X			X
槭樹科	X			X
金星蕨科		X	X	
天南星科	X		X	

圖 1a：FCA 初始定義

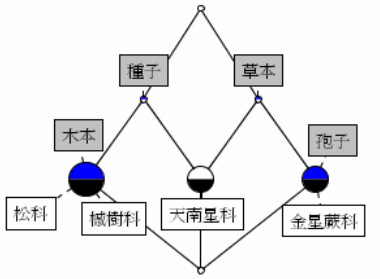


圖 1b：FCA 初始格線圖

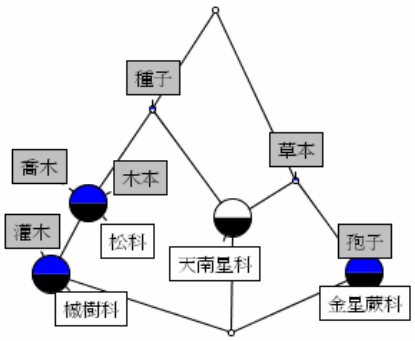


圖 1c：加入喬木及灌木屬性

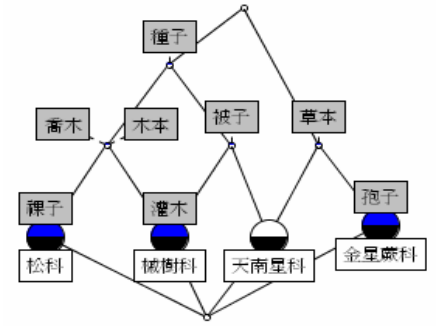


圖 1d：加入被子及裸子屬性

3. 測試修正：前述的設計階段應已逐步完成 Ontology 的顯性知識(explicit knowledge)，我們藉推論來獲得進一步的隱性知識(inferred knowledge)，由於推論過程應用了邏輯中的集合 (例如交集、聯集、餘集等)、公理(例如遞移性、對稱性、封閉性等)、及其他邏輯的組合，因此由推論而來的知識必須再檢驗正確性。本階段主要的驗證工作爲利用推論器，對知識進行一致性(consistency)的測試，當不一致性的狀況產生時，應回到設計階段修正。通常不一致性的問題較易藉偵

測而修正解決，另一類問題則是沒有發生不一致性，但在知識的認知中卻不正確，舉例而言，假如有一種百合科的植物名爲"台灣油點草"，它具有耐乾燥、少陽光、可盆栽等居家栽培的特性，但經推論後並未歸類於這個概念中，解決的方式是在屬性中加上封閉公理(Closure Axiom)，例如 ∀hasFeatures(耐乾燥⊔少陽光⊔可盆栽)，因此這類問題須仰賴領域專家的檢驗。

4. 開發佈署：Ontology 藉由邏輯推論所衍生的本體知識庫，此時已可回應該領域的特定問題，爲降低一般使用者在操作上的障礙，通常藉開發 Web- based 的前端介面，達成知識資源的分享。前端介面除使用 Ontology 爲知識的主要來源外，亦可藉資源的連結，使知識架構更爲完整。舉例而言，植物學門下已分別建立許多珍貴的圖檔，若結合 RDF 的使用，可由下達 URI 的方式取得資源，應用系統將更能完整表達知識體系。目前介面開發主要是以 Java-based 的應用程式爲主，可參考 Jena 及 SOFA 等獲得進一步的 API 資訊。
5. 整合及擴展：Ontology 的再利用包含整合及擴展兩個方向，整合是指再利用其他的本體知識庫，例如植物學門爲強化關於生長環境的知識表達，可能需要地質學門的本體知識庫來說明。擴展則是指 Ontology 是否能提供外界的再利用，例如動物學門爲說明草食動物的食物狀況，必須借重植物學門的本體知識庫。在 Gruninger et al. (2000)的研究中曾建議：整合及擴展的工作應在規劃階段即考量橫向的潛在需求，而預留可延伸的空間。本階段則是 import 或 export 的概念，作法上須以邏輯關係將其連結在一起。

四、 應用描述邏輯補足知識呈現模型

建置本體知識庫之目的是提供可作業的知識來源，而描述邏輯是補足 Ontology 在知識呈現所缺少的推理依據。Diego et al. (2002)的研究指出，知識庫包含 TBox ($\mathcal{T}$)及 ABox($\mathcal{A}$)兩項基礎元件，TBox 是以「詞彙」來組成，它代表對事物認知的「概念」，Ontology 將領域建立爲概念及子概念的階層架構，於是該 Ontology 成爲一個知識的概念分類，而在概念中的實體也因此有了包含關係(subSumption)，由於知識是認知之間進行交互作用的結果，概念層的包含關係並不能完整呈現其中的複雜性。Donini et al. (1996)指出：ABox 是以宣告「關係」的方式(稱爲 assertions)來定義對知識的其他說明，因此 ABox 補足複雜性的描述，通常藉邏輯式來完成。圖 2 是參考自 Baader et al. (2003)關於知識呈現使用描述邏輯的架構圖，其中描述邏輯是用來協助描述兩項元件的內容，圖右的推論機制須藉由兩項元件已知的知識，產生間接或可推導的隱藏知識，知識庫則做爲推理或應用程式的資訊來源。推論機制是我們利用知識庫的重要目的之一，成功的推論除了結果應正確外，須避免因邏輯設計不當，而產生模稜兩可的結論及效能低落的問題。爲加速在設計階段對描述邏輯的應用，我們歸納常用的方式及情境，製作成概念邏輯模型、屬性限制模型及模型補充，並製作成通用模式，如以下各節。

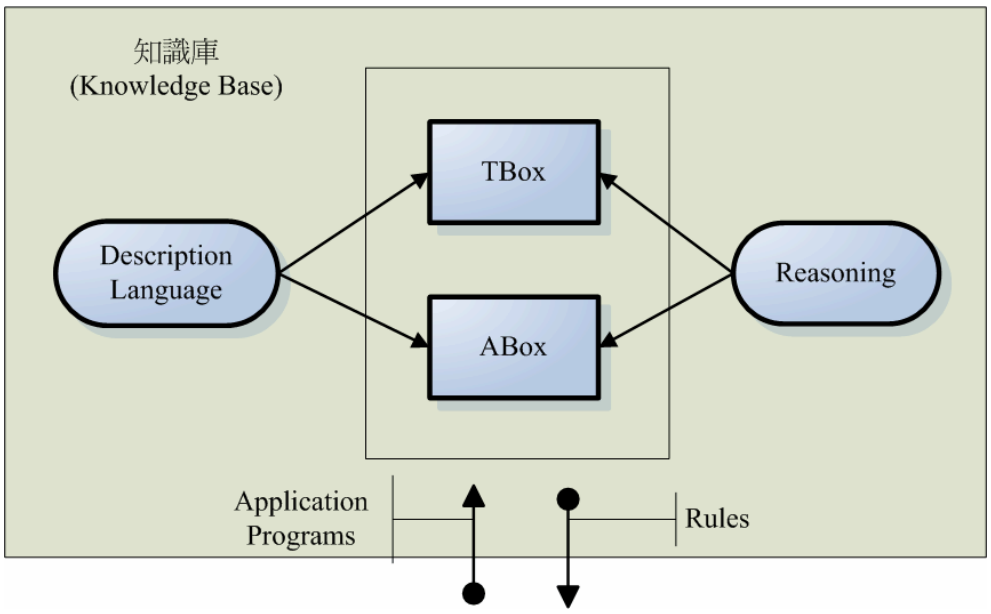


圖 2：知識呈現使用描述邏輯架構圖(Baader et al., 2003)

(一). 概念邏輯模型

概念是描述一個知識的抽象集合，在 OWL 中稱爲類別(Class)，概念與概念之間可藉邏輯關係形成二元關係，並表達另一個新的概念，例如「低海拔植物⊓單子葉植物」形成「低海拔單子葉植物」的新概念。當呈現概念的「補集」時，須利用一元邏輯關係，例如「¬低海拔植物」說明「非」低海拔植物的其他植物。一個概念的組成可能是複合的概念有限集合，我們進一步定義如以通用模式，邏輯符號及使用法參考表 2。

1. 假設 $C(x)$表達一個特定的概念，若它是有限集合的概念，則 $C(x)'$, $C(x)''$表示其次集合，$_{R(x)}$表示它們的邏輯關係。因此若 $\boldsymbol{C_{Complement}}$ 爲一個概念的補集，可表達爲

$$C(x) = (C(x)' \ {}_{R(x)} \ C(x)'')$$
$$C_{Complement} \equiv \neg C(x) \quad \text{.....(1)}$$

2. 假設 $C(x)$及 $C(y)$分別表達不同的特定概念，若它們是有限集合的概念，則 $C(x)'$, $C(x)''$及 $C(y)'$, $C(y)''$分別表示其次集合，$_{R(x)}$及 $_{R(y)}$分別表示邏輯關係。因此若 $\boldsymbol{C_{new}}$ 爲一個新的概念，$_{R(x,y)}$爲它們的邏輯關係，此時 $\boldsymbol{C_{new}}$ 可表達爲

$$C(x) = C(x)' \ {}_{R(x)} \ C(x)'' \quad or \quad C_{Complement(x)}$$
$$C(y) = C(y)' \ {}_{R(y)} \ C(y)'' \quad or \quad C_{Complement(y)}$$
$$C_{new} \sqsubseteq C(x) \ {}_{R(x,y)} \ C(y) \quad \text{.....(2)}$$

描述邏輯中常用於概念的邏輯有四種，表 2 是邏輯符號搭配其使用範例及使用情境，說明如下：

表 2：描述邏輯符號及概念使用範例情境

符號	使用範例	使用情境
⊑	Dicotyledon⊑VascularPlant	前者(subsumee)是較窄的概念，而後者(subsumer)是較通俗的概念，本例爲雙子葉植物包含於維管束植物。
⊓	Dicotyledon⊓Rhododendron	兩者皆是概念，每個概念可利用下節之屬性限制再予以限定，本例爲雙子葉植物與杜鵑科植物之交集。
⊔	Monocotyledon⊔Dicotyledon	使用方式同上，本例爲單子葉植物與雙子葉植物之聯集。
¬	¬Monocotyledon	本例爲一元邏輯，本例表示「非」單子葉植物的概念。

(二). 屬性限制模型

屬性是對概念更進一步的描述，例如對「汽車駕駛」的概念描述中，利用邏輯式 $CarDriver \sqsubseteq \exists hasVehicle(Car)$，其中 *hasCar* 是屬性，說明概念 *CarDriver* 與概念 *Car* 的擁有關係，而邏輯符號(∃)表示「至少」有一項成立的限制；承前例，若概念 Car 還包含如 Sedan、Coupe、Van、truck 等子概念，則描述「轎車駕駛」的概念，可能寫爲 $SedanDriver \sqsubseteq \exists hasVehicle(\exists hasCarType(Sedan))$。前述邏輯式經判斷，一位「轎車駕駛」應該也屬於「汽車駕駛」的分類，但事實上，推論結果並不會產生這個分類，深究其原因爲該邏輯式僅表達至少存在一個這樣的特性，但 *SedanDriver* 也可能會駕駛其他類型的車子，故推論並不會驟然決定，因此該邏輯式須加上具封閉性的「唯一」邏輯(∀)。「轎車駕駛」的概念改寫成 $SedanDriver \sqsubseteq \forall hasVehicle(\forall hasCarType(Sedan))$。用於屬性限制的邏輯可分爲對量、基數、及內容值等三種類型，通用模式如下：

1. 量的限制關係(quantifier)：用於表達「一些」或「唯一」的限制關係，它的組成依序是一個量的限制邏輯符號、一個屬性、及一個概念所組成，概念若爲有限集合，則如通用模式(1~2)所示。假設 $\boldsymbol{C_{new}}$ 爲一個概念，模型以量的限制邏輯 $_{R(x)}$起始(意義參考表 3)，*P(x)*代表屬性名稱，*C(x)*爲某項特定概念，$\boldsymbol{C_{new}}$ 可表達爲

$$C_{new} \sqsubseteq {}_{R(x)}\, P(x)\, C(x) \quad \text{.....(3)}$$

2. 基數的限制關係(cardinality)：基數是指作用的個數，因此基數限制是表達對個數「至少」、「至多」、或「等於」的關係，它的組成依序是一個屬性、一個基數的限制邏輯、及一個數字。假設 $\boldsymbol{C_{new}}$ 爲一個新概念，*P(x)*代表屬性名稱、$_{R(x)}$爲基數限制邏輯，*Numeral* 爲某一數字，$\boldsymbol{C_{new}}$ 可表達爲

$$C_{new} \sqsubseteq P(x)\ {}_{R(x)}\ Numeral \quad \text{.....(4)}$$

3. 內容值的限制關係(hasvalue)：用於表達概念與實體間的限制關係，它的組

成依序是一個屬性、一個內容的限制邏輯符號、及一個實體。假設 C_{new} 爲一個新概念，$P(x)$代表屬性名稱、$_{R(x)}$爲內容值的限制邏輯，$I(x)$代表某一實體名稱，C_{new} 可表達爲

$$C_{new} \sqsubseteq P(x)\ {}_{R(x)}\ I(x) \quad \text{.....(5)}$$

推論是很嚴謹的運算過程，對於沒有明確定義爲「真」的概念，不能斷然推論爲「真」，反之亦然。因此使用屬性的限制邏輯須非常謹慎，它也是決定推論正確與否的關鍵，表 3 是用於邏輯符號及屬性限制的使用範例情境。

表 3：描述邏輯符號及屬性限制的使用範例情境

符號	使用範例	使用情境
∃	∃*hasStem(ErectStem*⊓ *Dicotyledon)*	邏輯符號表達其後的屬性條件至少存在一項。
∀	∀ *hasVascular Dicotyledon*	邏輯符號表達其後的屬性條件存在唯一項。
≥	*hasShape*≥*3*	本例限制擁有植物形體的單元數目須至少爲 3 項。
≤	*hasShape*≤*3*	本例限制擁有的植物形體單元數目須至多爲 3 項。
=	*hasShape= 3*	本例限制擁有的植物形體單元數目須爲 3 項。
∋	*hasStem*∋ *ErectStem*	邏輯符號表達屬性且至少存在一項的實體。

(三). 模型補充

前述所推導的通用模式，可協助描述邏輯應用在 Ontology 的建置，但部份特性或限制方法須結合使用的情境而有額外的因應，本節補充應特別留意的三種類型：

1. 屬性及其相反屬性：在定義概念間的屬性時，通常以正向思考來訂定所具有的特性，因此以是 *has* 的性質，例如 *hasSteam*、*hasFlower* 等。所謂相反屬性(inverse property)，即是可以反向說明概念間的意義，例如假設 A 及 B 均爲概念，若 A *hasChild B*，則它的相反屬性成立時，可以寫成 *B hasParent A* 。屬性並一定都有其相反屬性，但若 *hasParent* 與 *hasChild* 是一對屬性及相反屬性時，習慣上另以 *IS-A* 的格式強調，因此 *hasParent* 改爲寫爲 *isParent*，相反屬性是具有雙向關係的意義。
2. 互斥概念：概念之間是否相關，須依 Ontology 欲解決的問題爲導向，如果問題的發展在概念的下層，此時須考慮將平行的概念設爲互斥，反之則不宜。舉例而言，若植物有陸生及水生兩種概念，如果衍生的問題不超過這個前提(亦即$\{陸生\} \cap \{水生\} = \phi$)，則應將兩個概念設爲互斥，以利推論的正確性。
3. 基本或定義的概念：邏輯式應用於概念定義時，一般定位在「必須」(necessary)的層次，前述所有的通用模式均可置於其中，我們統稱它們爲基本條件(Primitive)，其意義爲：當一個特定概念成立時，這些描述它的邏輯式亦須成立；但若這些邏輯式成立時，卻不能反推爲那個特定概念，亦即基本條件僅支

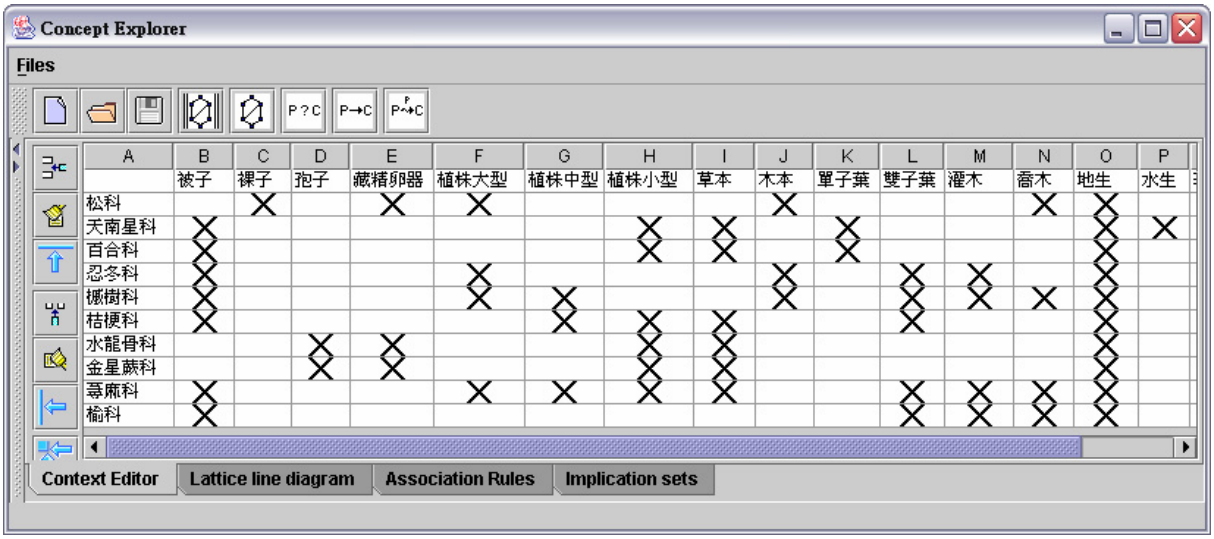
援單向推論。領域專家如果認定某些邏輯式具有雙向推論的特性(亦即反推的性質可成立)，則須將邏輯式定位在「必須且滿足」(necessary & sufficient)的層次，並使用全等邏輯符號(≡)表示，例如 $Parent \equiv \exists hasChild(Person)$ ，可使任一符合 *hasChild(Person)*的實例反推成立，這類可反推的邏輯式稱爲已定義條件(Defined)。

五、 實證維管束植物之本體知識庫

自然科學博物館爲提供大眾更簡易的知識體系查詢，在其數位博物館架構下，以植物學門的維管束植物做爲本體知識庫的先導應用範圍，期望藉 Ontology 的引進，改善使用者的對專屬知識在查詢上的障礙，並藉此實驗做爲未來推廣至各學門領域的經驗及修正。

(一). 本體知識庫的規劃及設計

本先導計劃開始之初，即曾對使用者進行使用狀況調查，發現使用者普遍受限於對專業術語的背景知識不足，因此希望以開放性的語意查詢輔助改善。本研究經評估現有的系統資料及可行的資訊技術後，決定採用 Ontology 來建立下一代的知識體系，並召集學門專家、技術專家、終端使用者、及維護人員共同組成。在規劃出欲解決問題的大略方向後，利用正規概念分析法(FCA)來篩選本體知識庫所需的概念及屬性關係。圖 3 是領域專家利用 FCA 的 Concept Explorer 工具的部份情況，最左欄代表 Objects、最上列代表 Attributes，其中註記爲「**X**」的格位表示 Object 具有該項 Attribute。



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
		被子	裸子	孢子	藏精卵器	植株大型	植株中型	植株小型	草本	木本	單子葉	雙子葉	灌木	喬木	地生	水生
1	松科		X		X	X				X				X	X	
2	天南星科	X						X	X		X				X	X
3	百合科	X						X	X		X				X	
4	忍冬科	X				X				X		X	X		X	
5	椰樹科	X				X	X			X		X	X	X	X	
6	桔梗科	X					X	X	X			X			X	
7	水龍骨科			X	X			X	X						X	
8	金星蕨科			X	X			X	X						X	
9	蕁麻科	X				X	X	X	X			X	X	X	X	
10	榆科	X										X	X	X	X	

圖 3：建立 Objects 及 Attributes 間的關係

正規概念分析法協助領域專家將顯性的關係予以確認，並經由 FCA 的分析特性找出領域專家忽略或待確認的其他關係，例如屬性檢查工具發現我們陳述的 Objects 都具有地生性質(如圖 4a)，由於沒有疑慮，可接受此項關係；又例如圖 4b 認爲具有地生及著生性質的 object，應該也具有孢子、藏精卵器、植株小型、及草本的性質，如果專家

不認同時，可藉由圖 4c 重新定義關係。本例再加上植株中型的性質(如圖 4d)，並確認儲存。

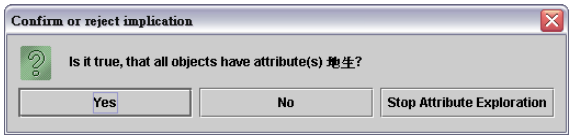


圖 4a：屬性檢查例-1

圖 4b：屬性檢查例-2

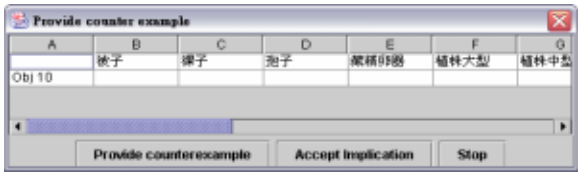


圖 4c：重新定義屬性

圖 4d：定義新屬性

設計階段的主要工作即爲找出 Ontology 所需的本體元件，而元件可衍生出概念或屬性的來源，圖 5 即爲經由 FCA Concept Explorer 協助所找出的關係，它是利用 Duquenne-Guigues 演算法(應用於 FCA 的數學模式之一)的計算方式獲得，其顯示格式爲

編號 <符合概念的數目> 前提項目 = => 結論

例如編號 24 可解釋爲：具有被子、植株小型、雙子葉、及地生性質的植物是草本植物，並已且有一項 object 符合。編號 24 以後的關係，因爲沒有任一項 object 符合，故以紅色字區隔，它們有可能是正確的關係，只是目前所提供的 object 數不足，因此暫時沒有符合的 Object，領域專家應再行確認其必要性。如果一切無誤，即可將這些概念及關係定義於 Ontology 中。

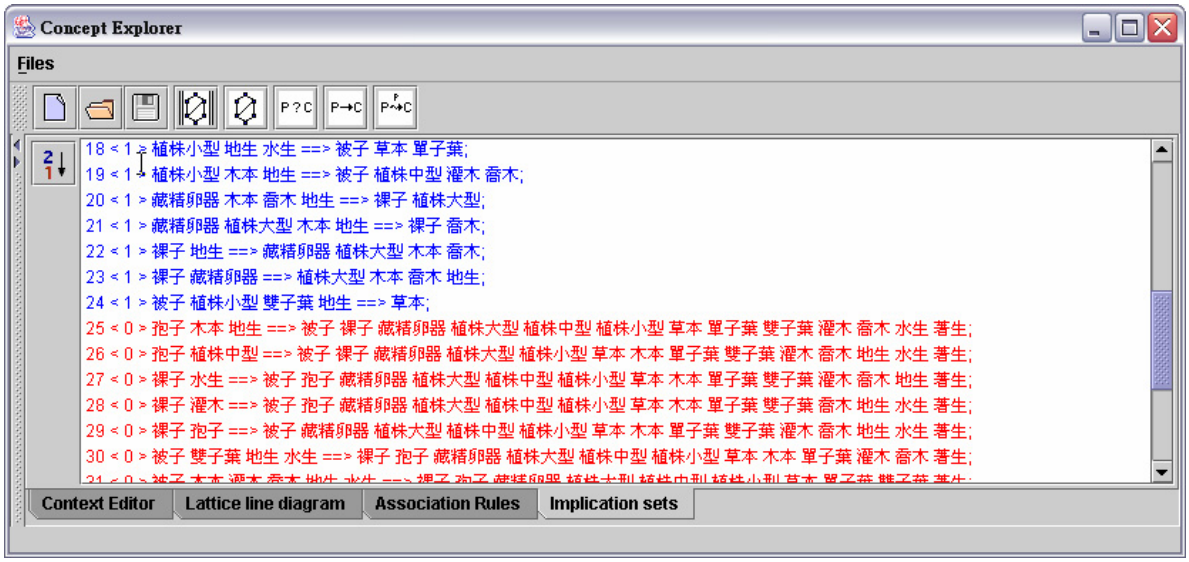


圖 5： 以 Duquenne-Guigues 計算獲得的關係

(二). 本體知識庫的呈現

本研究建置維管束植物的本體知識庫，是使用史丹佛大學醫學資訊中心所研發的

Protégé(參考 http://protege.stanford.edu)及 OWL Plugin 編輯工具完成，它是目前較成熟的 Ontology 開發工具之一 (Noy et al., 2001)。由於前面的設計階段已將本體所需定義的概念、屬性、及關係篩選出來，因此本階段是將它們分別對應至 classes、properties、及 Asserted conditions 中。爲方便說明本例，我們將範圍予以縮小，圖 6 是 Protégé-OWL plugin 的編輯畫面，主要視窗的用途如下：

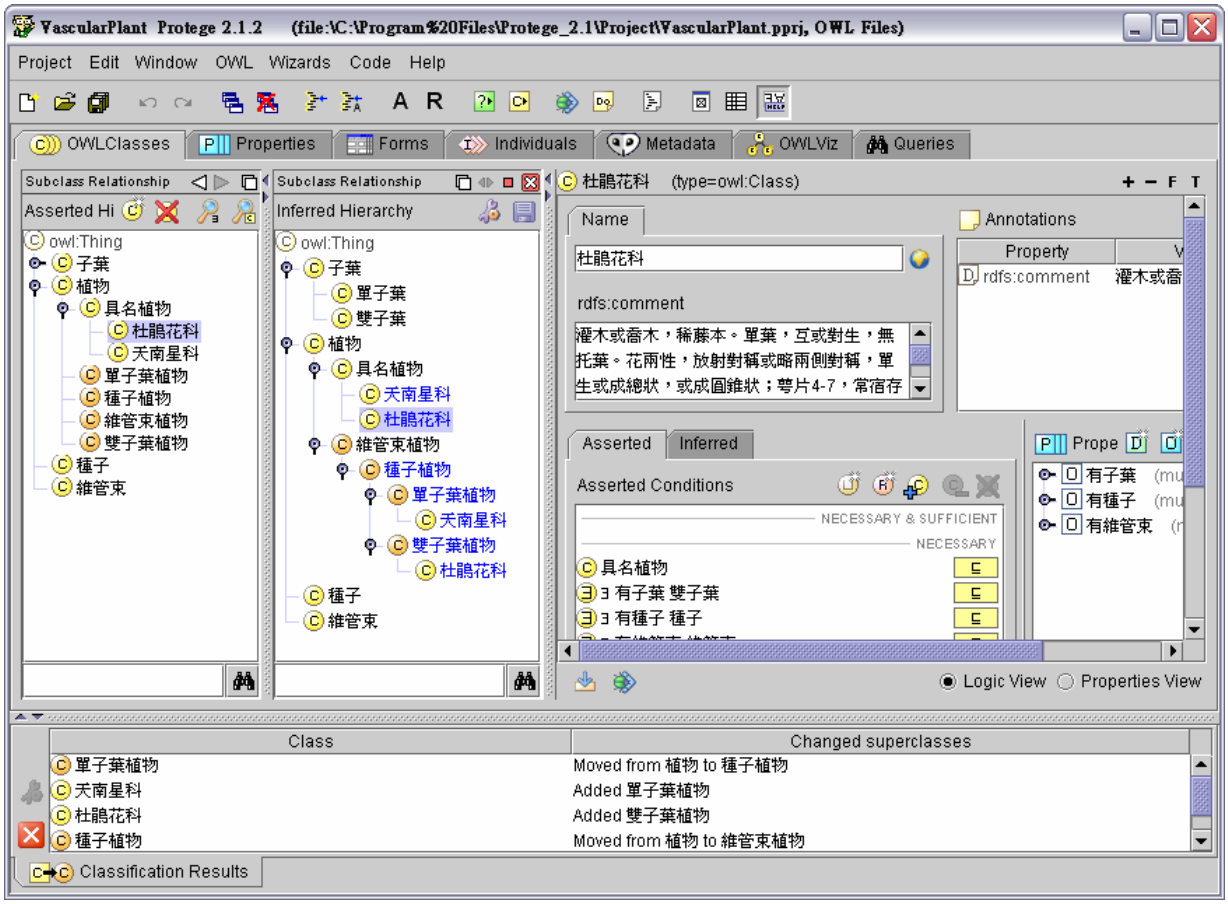


圖 6：使用 Protégé 及 OWL Plugin 編輯 Class

- 圖 6 左側子視窗是建置概念名稱的架構(asserted hierarchy)，例如編輯一個特定概念杜鵑花科植物在「具名植物」下，它代表階層從屬關係，因此杜鵑花科植物也將擁有其父階層定義的特性。
- 圖 6 左二的子視窗利用推論器(Racer)的推論結果(inferred hierarchy)，其中杜鵑花科被增加在種子植物下的雙子葉植物建置本體的另一項工作是定義屬性，它同樣也可以用階層表達從屬關係。
- 圖 6 右側由數個子視窗組成，如用於命名及說明概念(Name 及 comment 視窗)、相關的屬性(property 視窗)、及概念的定義條件(asserted condition 視窗)，定義條件即爲利用描述邏輯說明概念的地方，例如本例我們對杜鵑花科植物的簡單描述如下：

  杜鵑花科$\sqsubseteq$*具名植物*

  杜鵑花科$\sqsubseteq\exists$*有子葉(雙子葉)*

杜鵑花科⊑∃*有種子(種子)*

杜鵑花科⊑∃*有維管束(維管束)*

- 圖 6 下方的視窗是針對推論後，各概念在架構上的改變說明，設計者也可以在此調整階層關係。

概念及屬性的命名與階層化，通常是建置 Ontology 的初始工作，完成後須對概念下達明確的定義，前述概念杜鵑花科植物的各項邏輯描述，可藉圖 7 的畫面協助完成，圖 7 左是可用的屬性，右側是邏輯符號，下方則是表達將被限制的概念或其組合。例如想對杜鵑花科植物加上「維管束」的描述，而這項限制式僅是必須而非唯一的要件，因此邏輯式如「*杜鵑花科*⊑∃*有維管束(維管束)*」，其中「*有維管束*」爲屬性，「*(維管束)*」爲另一概念，描述邏輯可組合爲複雜的限制式，可參考本研究第 4 節的通用模式。Protégé 是我們編輯 Ontology 的平台，爲使本體內容符合 OWL 格式，須另行儲存。表 4 是將本例以 OWL 格式呈現的展示，我們摘錄其中的杜鵑花科的概念，其中<rdfs:subClassOf>標記內爲各項邏輯描述，例如「*杜鵑花科*⊑∃*有維管束(維管束)*」即爲

```
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
        <owl:ObjectProperty rdf:ID="有維管束"/>
    </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#維管束"/>
    </owl:Restriction>
 </rdfs:subClassOf>
```

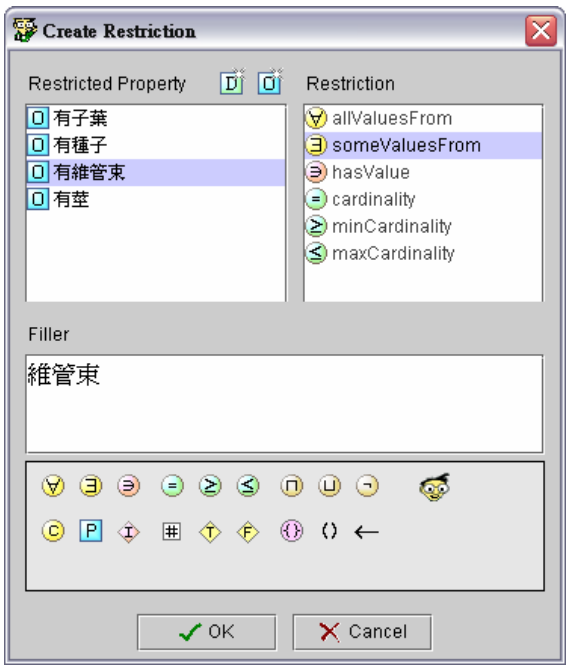


圖 7：編輯 Class 的描述邏輯式

表4: 以OWL格式呈現杜鵑花科概念

```
<?xml version="1.0" encoding="BIG5"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    ..........
    ..........
  </owl:Class>
  <owl:Class rdf:ID="杜鵑花科">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    灌木或喬木，稀藤本。單葉，互或對生，無托葉。花兩性，
    放射對稱或略兩側對稱，單生或成總狀，或成圓錐狀；萼片 4-7，
    常宿存；花冠 4-7 裂，漏斗形或鐘形；雄蕊與花冠裂片同數或爲其兩倍，
    花藥 2 室，孔裂或縱裂；子房上或下位，常 5 室。蒴果或漿果。臺灣有 6 屬。
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:someValuesFrom>
          <owl:Class rdf:about="#雙子葉"/>
        </owl:someValuesFrom>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="有子葉"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="具名植物"/>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="有種子"/>
        </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#種子"/>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="有維管束"/>
        </owl:onProperty>
        <owl:someValuesFrom rdf:resource="#維管束"/>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  ..........
  ..........
```

(三). 本體知識庫的應用

我們承前例的 Ontology 建置範例，再予增加更多的概念及屬性後，應將具體的實例

分別納入各概念中，例如國內杜鵑花科包含有金毛杜鵑、紅毛杜鵑、西洋杜鵑等十數種，它們原則上繼承概念的屬性及定義，若有相異或其他額外的屬性及定義，亦可再予描述，圖 8 是針對 Ontology 的知識擷取應用，本例是訂定查詢具有雙子葉特性的維管束植物。值得注意的是各概念在雙子葉特性上，是一項由推論獲得的隱性知識，而圖 8 右側的結果視窗，顯示了來自於各概念的具體實例。

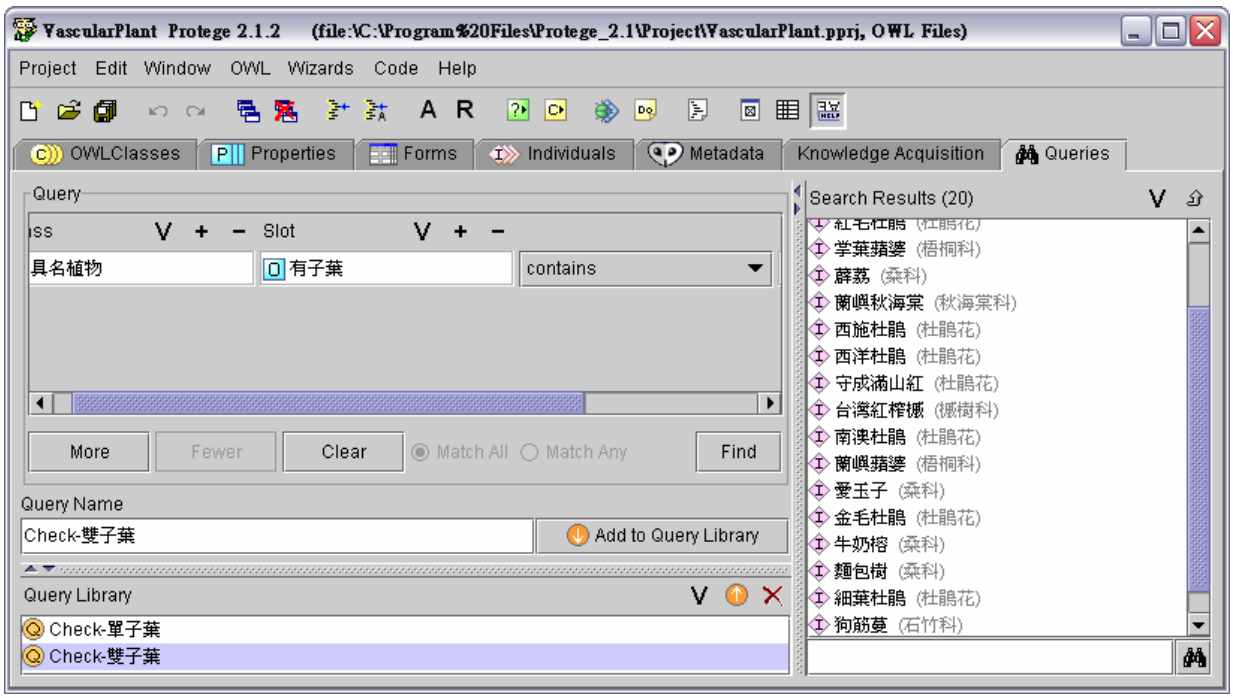


圖 8：知識擷取應用

六、 研究討論與結論

(一). 研究討論

由第五節的案例可發現，利用 Ontology 所建立的系統較傳統系統複雜，特別是本體知識庫的建置過程也較傳統知識庫的設計更爲瑣碎，但這種以 XML 爲資料格式的 Ontology，可滿足目前資訊技術對開放式環境的要求。因此，Ontology 能在知識庫的分享、再利用及整合等，皆較傳統知識庫有明顯的改善。Kayed and Colomb (2005)曾指出：以 Ontology 爲核心的系統，能將系統的複雜面由使用端轉移到伺服端，雖然伺服系統的開發較困難，但卻可以重複多次使用，例如傳統系統是以關鍵字查詢，使用者可能須輸入多個關鍵字才可獲得完整結果，而 Ontology 的系統則將解譯語意的複雜面留給伺服端的知識庫處理。

本研究爲能協助知識庫開發者收集對真實世界的認知，並進行轉換爲資訊系統所接受的知識表達，因此提出本體知識庫發展程序，建置過程的關鍵工作即爲概念及其關係的獲得，爲符合如圖 2 的知識庫架構圖，我們利用正規概念分析(FCA)將形而上的概念分析轉換爲具體型式的術語概念，其中 FCA 藉演算法(如 attribute exploration 及 Duquenne-Guigues 等)，可分析概念之間的共通性，因此可逐步推導各層概念結構，再

利用描述邏輯將概念階層給予具體邏輯式的定義，完成知識庫的 TBox 建置。本研究確認正規概念分析與描述邏輯可協同完成 TBox 的建立，但在 ABox 部份，因涉及個別實例的細節，這項工作仍須由專家與知識工程人員配合完成，此爲本研究的主要限制。

另外，本研究將植物學門的維管束植物做爲本體知識庫範圍，在實證的過程中有下列幾項經驗提出：

1. 建置程序不同：傳統資訊系統循著需求、分析、設計等步驟後，系統雛型已大致成型，需求者與工程技術人員的互動是相對較鬆散的。建置 Ontology 則是以問題導向並配合「專家」的知識爲主，因此必須藉綿密的互動及反覆修正，才可獲得有意義的 Ontology。本研究將建置本體知識庫的程序建議如第三節的五大階段，其中設計階段是建置過程最大的挑戰，我們認爲應以領域專家擔任主導工作。
2. 建置內涵不同：傳統資訊系統是以應用程式搭配資料爲主，而本體知識庫則是以建置概念架構與邏輯關係爲主。領域專家應儘可能將概念利用增加屬性方式予以區隔，以利概念的獨立。惟當概念及屬性增加時，其交錯的關係不利人工判斷，因此須藉工具協助分析，降低人爲的疏失。由概念描述轉換到邏輯式的下達，亦是領域專家與資訊技術人員間對認知的傳承與交替，爲降低轉換時的瓶頸，本研究在第四節建立常用的描述邏輯通用模式(1~5)及其補充，提供邏輯定義的依據。
3. 應用方式不同：傳統資訊系統是以建置的內容物爲運作範圍，本體知識庫則是包含擴充的推論結果爲範圍，並形成可操作的知識庫。有別於傳統以關鍵字或關聯資料庫的比對查詢，本體的內容物是經專家所訂定的知識，因此對知識庫的查詢爲知識擷取，故使用方式也不同。

(二). 研究結論

本研究主要探討以 Ontology 爲基礎的知識庫建置中，如何解決領域專家、知識工程人員及資訊系統等在建置過程中的問題。由於本體知識庫建置方式具有高度的人爲依賴性，因此建置的技術及策略並沒有固定的型式，本研究將建置過程區分爲規劃、設計、測試修正、佈署及整合擴展等階段進行，爲獲得 Ontology 所須要的概念及其關係，我們利用正規概念分析法將真實世界的認知予以收集，並發展常用的邏輯類型模式，以降低轉換爲資訊系統格式的障礙。本研究另提供植物學門的維管束植物爲驗證實例，由結果顯示：本研究確可協助領域專家、知識工程人員及資訊系統，提供建置本體知識庫的參考程序。

參考文獻

[1]. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Parel-Schneider, P.,(eds) *The Description Logic Handbook*, Cambridge University Press, Cambridge, UK, 2003

[2]. Chandrasekaran, B., Josephson, J.R., and Benjamins, V.R., "What are ontologies, and why do we need them," *IEEE Intelligent Systems* (14:1) 1999, pp:20-26

[3]. Decker, S., Melnik, S., Harmelen, F.V., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., and Horrocks, I., "The Semantic Web: The Roles of XML and RDF," *Internet Computing* (4:4) 2000, pp:63-74.

[4]. Diego, C., Giacomo, G.D., and Lenzerini, M., "Description Logics: Foundations for Class-based Knowledge Representation," *Proc. of the IEEE Sym. on Logic in Computer Science* 2002, pp:359-370

[5]. Dieter, F., Horrocks, I., Harmelen F.V., McGuinness, D., and Patel-Schneider, P.F., "OIL: Ontology Infrastructure to Enable the Semantic Web," *IEEE Intelligent System* (16:2) 2001, pp:38-45.

[6]. Donini, F.M, Lenzerini, M., Nardi, D., and Schaerf, A., "Reasoning in description logics," In G. Brewka (ed) *Foundation of Knowledge Representation*, CSLI Publications, 1996

[7]. Feigenbaum, E.A., "The art of artificial intelligence: Themes and case studies of knowledge engineering," Proceedings of Fifth International Joint Conference on Artificial Intelligence 1977, pp:1014-1029.

[8]. Fox, M.S. and Gruninger, M., "Enterprise Modelling", *AI Magazine* (19:3) 1998, pp: 109-121.

[9]. Gillam, L., Tariq, M., and Ahmad, K., "Terminology and the construction of ontology," *Terminology*, (11:1) 2005, pp: 55-81.

[10]. Gruber, T.R., "Towards Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human-Computer Studies* (43:5-6) 1995, pp:907-928.

[11]. Gruninger, M., Atefi, K., and Fox, M.S., "Ontologies to Support Process Integration in Enterprise Engineering", *Computational and Mathematical Organization Theory* (6: 4) 2000, pp: 381-394.

[12]. Horrocks, I., Patel-Schneider, P.F., and Harmelen, F.V., "From SHIQ and RDF to OWL: the making of a Web Ontology Language," *Web Semantics: Science, Services and Agents on the World Wide Web* (1:1) 2003, pp:7-26.

[13]. Hui, B. and Yu, E., "Extracting conceptual relationships from specialized documents," *Data & Knowledge Engineering*, (54:1) 2005, pp:29-55.

[14]. Jiang G., Ogasawara, K., Endoh, A., and Sakurai, T., "Context-based Ontology Building Support in Clinical Domains using Formal Concept Analysis," *Journal of Medical Informatics* (71:1) 2003, pp:71-81.

[15]. Kayed, A. and Colomb, R.M., "Extracting ontological concepts for tendering conceptual structures," *Data & Knowledge Engineering*, (40:1) 2002, pp:71-89.

[16]. Kim, H.M., Fox M S., and Gruninger, M. "An ontology for quality management - enabling quality problem identification and tracing," *BT Technology Journal*, (17:4) 1999, pp:131-140

[17]. Knublauch, H., Dameron O., and Musen, M.A., "Weaving the Biomedical

Semantic Web with the Protégé OWL Plugin," *Proc. Of 1st Intl. Workshop on Formal Biomedical Knowledge Representation*, 2004

[18]. Kralingen, B.W., Visser, P.R.S., Bench-Capon, T.J.M., and Herik, H.J., "A principled approach to developing legal knowledge systems," *International Journal of Human-Computer Studies* (51:6) 1999, pp:1127-1154.

[19]. Lehmann, F., (ed) *Semantic Networks in Artificial Intelligence,* Elsevier Science Inc., New York, USA, 1992.

[20]. Lin, J., Fox M.S., and Bilgic, T., "A Requirement Ontology for Engineering Design," *Concurrent Engineering: Research and Applications*(4:4) 1996, pp:279-291.

[21]. López de Vergara, J.E., Villagrá, V.A., and Berrocal, J., "Applying the Web Ontology Language to management information definitions," *IEEE Communication magazine* (42:7) 2004, pp:68-74.

[22]. McGuinness, D.L., Fikes, R., Hendler, J., and Stein, L.A. "DAML+OIL: an ontology language for the Semantic Web," *IEEE Intelligent Systems* (17:5) 2002, pp:72-80

[23]. Minsky, M., "A framework for representing knowledge," In Winston, P.J., (ed) *The psychology of computer visions*, McGraw-Hill, New York, 1975.

[24]. Noy, N.F. and McGuinness, D.L., "Ontology Development 101: A Guide to Creating Your First Ontology," *Stanford Knowledge Systems Laboratory Technical Report KSL-01-05*, 2001.

[25]. Noy, N.F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R.W., and Musen, M.A., "Creating Semantic Web contents with Protege-2000," *IEEE Intelligent Systems* (16:2) 2001, pp:60–71.

[26]. Sugumaran, V. and Storey, V.C., "Ontologies for Conceptual Modeling: their creation, use, and management," *Data &Knowledge Engineering* (42:3) 2002, pp:251-271.

[27]. Tamma,V., Phelps, S., Dickinson, I., and Wooldridge, M., "Ontologies for supporting negotiation in e-commerce," *Engineering Applications of Artificial Intelligence*, (18:2) 2005, pp: 223-236.

[28]. Uschold, M. and Grueninger, M., "Ontologies: Principles, Methods and Applications," *Knowledge Engineering Review* (11:2) 1996, pp:93-155.

[29]. Van der Vet, P.E. and Mars, N.J., "Bottom-up construction of ontologies," *IEEE Transaction on Knowledge and Data Engineering*, (10:4) 1998, pp: 513-526.