

第二章 本體論的探討

本章旨在探討本體論的相關文獻，作為建立本體的基礎。首先在第一節中由哲學上的本體論開始引言，進而延伸到資訊科學界的本體。第二節則探討本體的定義、成份、認定標準、並給予範例說明。第三節探討本體在不同領域與不同層面的應用。第四節探討不同觀點的本體分類方式。第五節探討本體的發展原則與各種不同的本體發展方法論。第六節探討本體表徵的內部基礎與各種本體語言。第七節探討本體的評鑑項目與技巧。第八節探討本體的整合與自動建構本體的學習方法等。最後在第九節作一個小結。

第一節 本體論與本體

一、本體論 *Ontology*

英文的 *Ontology* 源自於拉丁文的 *Ontologia*，而 *Ontologia* 又源自於希臘文的 *onto* 與 *logos*。*on* 是指目前正在發生的事情，也就是存在的事情 (*being*)；*logos* 是指宇宙萬事萬務的本質，*logy* 則是指基礎的科學知識或理論，合起來就是指「宇宙間正在發生的事情」或是「宇宙間的基礎知識」。因此，*Ontology* 就是在討論宇宙間萬事萬物存在本質的一種知識體系。*Ontology* 在中文的翻譯上有「本體論」、「存在論」、「存有論」、「實有論」等多種，但是較被兩岸學術界所接受的翻譯為「本體論」。本體在日常生活中具有多樣的意義，例如「基本」、「基礎」、「根本」、「根據」、「本質」、「實質」、「真實」等，因此，*Ontology* 翻譯成本體論，最能彰顯其意義與內涵 (楊學功，2002)。

二、本體 ontology

Ontology 起源於亞里斯多德的哲學思想，因此也常常被稱為「哲學之頂」(the summit of philosophy)，主要目的在探討宇宙間萬事萬物存在的本質。而這些本質可以分成十種不同的屬類，最後被普遍接受的結論認為，萬事萬物存在的本質可以由十個抽象的概念組成，即物質 (substance)、數量 (quantity)、品質 (quality)、關係 (relation)、行動 (action)、情感 (passion)、時間 (time)、地點 (place)、氣質 (disposition)與習性 (habits)等 (Corazzon, 2007)。也就是說，任何一個物件都可以用「是什麼」(is what)來加以表示，例如「某一個東西是物質」、「某一個東西是數量」等。「是什麼」是一種追本溯源的探究、是連結兩個物件的連接詞，因而可以建立上下之間的關係。持續的探究根源與建立關係就可以瞭解一個領域的知識組成。因此 Guarino 和 Giaretta (1995)建議採用大寫的 Ontology，並且不加冠詞，來表示哲學上探究根源的理論，而採用小寫的 an ontology 或 ontologies，來表示探究根源的結果，也就是領域的知識組成，兩者之間應該有所分辨。因此，本研究將以本體論或 Ontology 表示探究根源的理論，以本體或 ontology 表示探究根源的結果。

我國碩博士論文資訊網中對於 Ontology 的譯名也非常的分歧，以 ontology 為關鍵字查詢其譯名所獲得的結果，包括知識本體論、學習本體論、實體論、本體分類論、知識領域、本體知識、知識庫、本體知識庫、映拓邏輯、本體論分析法、概念模型、知識地圖、本體知識架構與分類論等十幾種名稱 (國家圖書館, 2007)。最常被用到的名稱為本體論與知識本體論兩者，但是大部份的論文都沒有清楚的界定

理論與實體的分別，使得本體論可能指的是哲學上探究根源的理論，也可能指的是所探究出來的結果。為了能夠清楚的分辨兩者之不同，宜使用「本體論」來表示 Ontology、以「本體」來表示 ontology。

第二節 本體的定義

一、分歧的本體定義

在本體的定義方面，許多專家學者的看法有很大的差異，有些定義極為鬆散，有些定義卻又非常嚴謹。茲將具有代表性的本體定義按照時間序列敘述如後。首先 Neches et al. (1991)認為一個本體是由某個主題領域、該領域的名詞、名詞與名詞之間的關係、以及組合這些名詞成為句子的推理規則等所組成。這個定義將本體視為某一個領域的知識基礎，是具有人工智慧的知識庫，儘管這個定義並未描述術語之間是何種關係，也未提及本體的表徵方式，但與其他後來的許多本體的定義而言，已經算是相當嚴謹的定義。

最常被引用的本體定義是 Gruber (1993)所提出，Gruber 認為「本體是一種概念化的顯性規範」(An ontology is an explicit specification of conceptualization)，這個定義強調了「概念化」、「規範」、與「顯性」三個特性；概念化 (conceptualization)是由某一個概念認知到真實事物的過程 (Maedche & Staab, 2002)。圖 2-1 所示就是概念化的過程，首先由一個符號或稱為字彙來喚起一個概念，再根據規範來參照真實的事物，就可以獲得具體的實例 (instance)，或稱範例。例如有些人可能認知到 Protege 是馬自達汽車的一種車款，有些人可能認知到

Protégé 是一種本體編輯工具，有些人會認知到 Protege 是一種 DVD 複寫器。不同的人可能認知到不同的事物，這是因為符號本身並沒有加上其他的規範，因此一個符號可能代表不同的概念，但是當加上了規範之後，符號所代表的事物就會變得更加明顯。所以本體就是由概念到實例的正規化表示，因此本體也常被稱為正規本體 (formal ontology) (Guarino & Giaretta, 1995; Guarino, 1997; Noy & McGuinness, 2001)。

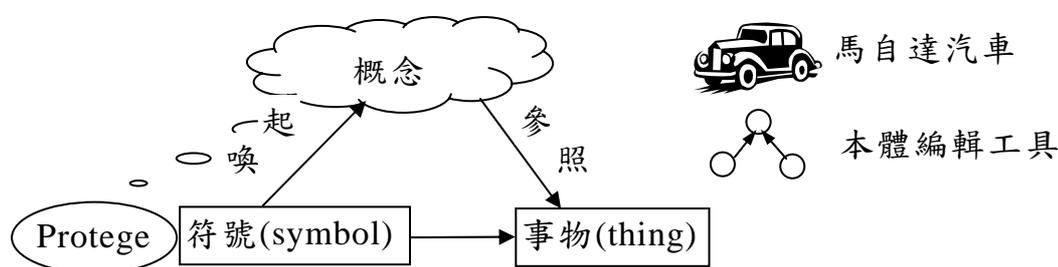


圖 2-1 概念化的過程(資料來源：Maedche & Staab (2002))

Alberts (1993)將本體視為特定領域知識的概念分類，並定義這些概念的語意解譯，使得概念所代表的意義變得更加明確。由這個定義來看，本體又類似於分類學 (taxonomy)，但是分類的方式是根據語意，而不是根據語法。

Knowledge Based Systems, Inc. (1994)認為本體應該包括三個成分，某一個領域的名詞所組成的名詞目錄、結合某些名詞成為正確敘述的規則，以及預定的推理模式等。因此，本體就如同人類所熟知的資料字典一般、再加上文法規則與行為模式，這個定義強調本體和資料字典極為相似，但是仍然有差異存在。資料字典是將所有的單字或片語集合在一起，並且透過自然語言給予定義，相反的，本體則是以正規語言來定義。

Takeda, Iino, 與 Nishida (1995)提出了一個非常鬆散的本體定義，認為在真實世界中，為了達成某一個特定的目的，而對某一件事情達成一致的共識，就是本體。這個定義只觸及了本體的表象，並沒有陳述本體的組成，也未深入到本體的內涵。

Mizoguchi 與 Ikeda (1996)提出一個相當嚴謹的定義，認為本體是根據本體論所發展出來的一種概念的組成架構。也就是針對某一件事情的概念、關係、屬性與值等，進行清楚明白的描述。其描述方式可以是自然語言的描述、圖形結構的描述、樹狀結構的描述、或是正規語言的描述等。這種型的本體，不單是存在於傳統的哲學之中，也存在於任何事務與任何領域之上。這個定義點出了本體的成份為概念、屬性、概念之間的關係、與值等四種成份，但是允許使用不同的方式來塑模本體，和 Gruber (1993)的定義不同之處在於後者要使用正規的法則。

van Heijst, Schreiber, 與 Wielinga (1997)則以知識工程的角度，將本體定義成某個特定領域中概念化知識的層次規範。這個定義比較特殊的地方，是在概念化的層次，也就是要將知識劃分層次，由抽象知識、一般知識、到具體知識加以分層。由知識工程的觀點來看，本體顯然可以由上層的概念，透過層次規範來導出實例，也就是說本體一定要涵蓋實例，如果只是到達概念的層次則不應算是本體。

Poli (2002)則持一般性的觀點，將本體定義成一種項目理論 (theory of items)，目的在處理組成這個世界的所有項目的資訊，包括實體的資訊與抽象的資訊、存在的資訊與不存在的資訊、實際的資訊

與理想的資訊等。而這些資訊又可以按照其基本的組成分成三個維度的資訊：符號的資訊 (semiotic information)、語意的資訊 (semantic information)、與本體的資訊 (ontological information)。符號的資訊主要是分析用來傳達資訊的各種符號，也就是語言的分析，屬於語法的層級，或稱為標記的層級；語意的資訊則是分析資訊的分類型式，例如標準的分類、認知的分類、社會的分類等，都是屬於語意的資訊，也就是符號所代表的內涵；本體的資訊則是由本質上的分類來分析，也就是由概念上來分析。

二、本體的認定標準

在前述諸多有關本體的定義中，有的把本體看成是一個分類架構，有的把本體當成是字典，有的把本體當成是索引典，有的則是將本體視為知識基礎，有的將本體看成是知識工程，有的將本體看成是項目理論、甚至是鬆散到將本體看成是對事務的一致觀點。這些定義只能部分反應了本體的內涵，而 Jacob (2003) 針對多樣的本體定義提出了評論，認為本體論與本體雖然是極為熱門的一個研究主題，但是大家似乎還是對本體論與本體的名稱感到迷惑。因為過去對於本體的解釋實在太多，包括目錄 (category)、分類學 (taxonomies)、分類綱要 (classification schema)、控制字彙 (controlled vocabularies)、階層架構 (hierarchies)、字典 (dictionaries)、索引典 (thesauri)、術語集 (terminologies) 等。在國內部份也顯示相同的現象，許多分類系統也稱為本體，如果鬆散到將這些分類系統也視為本體，實在是違背了本體是領域知識組成的內涵，也降低了本體的發展潛力。因此許多文獻強調本體中的概念必須要合乎 isa 的關係，並安排成為階層架構

(Ceusters, Smithb, Kumarb, & Dhaenaa, 2004 ; Jacob, 2003 ; Noy & McGuinness, 2001)。除了 isa 的關係之外，還允許「部份-整體」(partof) 關係存在(Noy & McGuinness, 2001) 。isa 與 partof 的主要差異在於 isa 是一種父子關係、也就是超類別與子類別的階層架構關係，也是一種繼承的關係，在上層的概念比較一般化，在下層的概念比較具體化，到了最底層時就變成了一個具體的實例。例如人類是一種哺乳類；而 partof 是一種整體-部分的關係，也就是一種涵蓋關係，例如引擎是汽車的一個部分。

合乎 isa 關係的子概念會繼承超概念的所有屬性，而且還可以定義自己的屬性。例如人類 isa 哺乳類、哺乳類 isa 動物，如果以階層架構而言，哺乳類是動物的一種。因此動物是超概念，而哺乳類是子概念，根據這樣的定義可以定出圖 2-2 簡單的本體，最上層的概念為動物，第二層的兩棲類、哺乳類、與爬蟲類等，都是屬於動物，合乎 isa 的規範，第四層的張三、李四、與王五等，則是屬於實例。

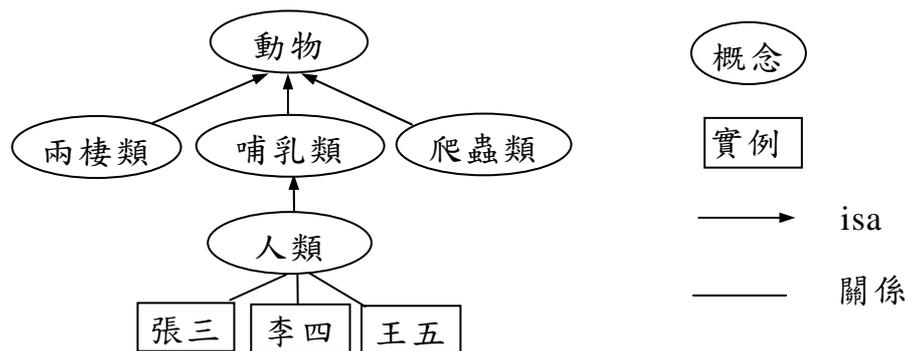


圖 2-2 本體的範例 (資料來源：本研究)

由於 ontology 有許多不同的定義，使得本體的內涵出現很大的落差，因此，Guarino 與 Giaretta (1995) 分析了許多有關 ontology 的研究報告，彙整出七種 ontology 的解釋：

- 1、把 ontology 當成是一門哲學上的學科。
- 2、把 ontology 當成是一種非正規的概念系統。
- 3、把 ontology 當成是一種正規的語意說明。
- 4、把 ontology 當成是一種概念化的顯性規範。
- 5、把 ontology 當成是一種邏輯理論，用來表現概念系統。
- 6、把 ontology 當成是一種邏輯理論中所用的字彙。
- 8、把 ontology 當成是一種邏輯理論的後設規範 (meta-level specification)。

在這七種解釋之中，第一種和其他六種有明顯的不同，第一種解釋把 ontology 當成是一門哲學上的學科，和人工智慧與電腦科學的 ontology 應該有所分辨。因此 Guarino 與 Giaretta (1995) 建議採用大寫的 Ontology，並且不加冠詞，來表示屬於哲學的學科；而另外採用小寫的 ontology 再加上冠詞 an，即一個本體 (an ontology) 或許多個本體 (ontologies) 來表示應用領域的本體。第二種解釋是採用非正規的方式，也就是以自然語言來描述概念系統，概念本身是一種語意的層次；第三種解釋則以正規的方法來做語意上的規範，所以第二種與第三種解釋都是在「語意」層次，並沒有牽涉到「語法」的規範。第五種到第七種解釋都和邏輯理論有關，也就是有「語法上」的規範，其中第五種解釋是把 ontology 看成是一種邏輯理論，其作用在表現語意的概念；第六種解釋是將 ontology 看成是邏輯理論所使用的字彙，也就是所有的名詞、屬性與關係等的名稱；第七種則是邏輯理論的後設

規範、或是稱為詮釋規範，用來規範邏輯理論的架構與成分。在這七種解釋之中，第四種解釋就是 Gruber (1993)的定義，也是最常被引用的定義，在 Gruber (1993)與 Guarino (1995)的定義中，本體是指概念化的顯性規範，也就是概念的正規化，這樣的定義有極大的解釋空間，正規化到什麼程度才算是本體呢？

Guarino 與 Giaretta(1995)雖然列出了七種有關 ontology 的解釋，但是仍然未能清楚的定義何謂本體。McGuinness (2002)進一步認為要合乎兩個條件：合乎 isa 架構、與利用某種正規化語言加以塑模，才能稱之為本體，也就是說，本體必須要具備「正規 isa」 (formal isa) 的架構。

Uschold (1996)也利用概念正規化的程度來認定是否為本體，概念正規化的程度分成四個等級，要稱為本體至少要達到半正規化或高度正規化的程度：

- 1、高度非正規化 (highly informal)：完全以自然語言的方式來表示，沒有任何結構上的限制或規範。
- 2、結構化非正規化 (structured informal)：完全以自然語言的方式來表示，但是在結構上必須受到限制或規範。
- 3、半正規化 (semi-formal)：使用正規語言來表示，但是對於名詞的定義並未嚴謹規範。
- 4、高度正規化 (rigorously formal)：使用正規語言來表示，並且很嚴謹的定義名詞、屬性與語意等，還要確保其正確性與完整性。

McGuinness (2002)的「正規 isa」架構和 Uschold (1996)的「半正

規化架構」，都是用來決定是否為本體的依據。雖然在名稱上不盡相同，但是其內涵是相同的，都是以是否使用正規法則來作為認定的依據。但是這樣的認定方式並不周延，主要的原因是由發展本體到應用本體有不同的層次，只要是合乎 isa 或 partof 的階層架構，就可以稱為本體。至於本體的表徵方式，只是使用的工具不同而已。例如使用自然語言、圖形結構、邏輯語言、資料庫系統、甚至是電腦語言等都可以。如果一定要限制非使用正規法則不可，則在發展階段中的自然語言描述或圖形描述都不算是本體，等到帶入正規法則之後才叫本體，似乎有一點牽強。但是以本體的嚴謹性而言，利用正規法則確實可以讓本體更加周延，也才可能融入語意網的應用。

綜合言之，本體是針對某一個領域，為了達成人與人、人與系統、系統與系統之間的相互溝通，而以正規化的方法，將領域的概念、概念與概念的階層關係、概念的屬性與屬性的值等明確的加以規範，以便將正確的資訊，在正確的時間，以正確的方法，傳送給正確的個體。在這個定義之中，共有四層的意義：

- 1、本體的領域：本體必定是針對某一個領域來發展，這個領域可能是針對極為特定的領域，所發展出來的領域本體，或是針對一般的領域所發展出來的通用本體，完全看本體的應用層面來決定。
- 2、本體的目的：本體的目的是人與人、人與系統、系統與系統之間相互溝通的媒介，能夠相互溝通才能進一步擴展其應用。
- 3、概念關係的明確規範：為了達成相互溝通的目的，需要就該領域的所有概念加以語意定義，並且概念之間的關係也要加以明確的規範，以便在概念化的過程中獲得明確的實例。
- 4、正規化的語言：使用正規符號或正規語言來規範概念與概念之間

的關係，除了人可以瞭解之外，機器也可以閱讀，使本體與資訊系統的應用相互結合。

三、本體的成份

本體的組成除了概念之外，還具有其他的成份，來豐富本體的內涵，正如同本體具有多樣的定義一般。本體的成份所使用的名稱也相當的分歧，根據 Kiryakov, Dimitrov, 與 Simov (2001)的研究顯示，本體的表示方式有許多種，每一種表示之中所用的名詞也不盡相同。

表 2-1 不同本體表徵方式所使用的本體成份名稱

表徵方式	各層次的名稱			
語意式	concepts	properties	attributes	instances
框架式	classes	slots	facets	instances
描述邏輯式	concepts	roles		individuals
物件導向式	classes	attributes		objects
Cyc 式(Cycorp 公司)	collections	predicates		individual
Knowledge Based Systems, Inc 式	kinds	properties	attributes	

資料來源：整理自 Kiryakov, Dimitrov, & Simov (2001)

表 2-1 所示為較常用的表示方式，其中又以語意式與框架式兩者最為普遍。本體的基本組成元素是概念 (concept)，也稱為類別 (class)。Cyc 公司使用比較特殊的名稱叫做集合 (collection)。Knowledge Based Systems 公司更使用種類 (kind)。概念與類別是最常

被採用的名稱，當本體以自然語言表示時通常使用概念，但是當本體帶入本體工具，或應用在資訊系統中時，則以使用類別居多，因為在物件導向的語言中都使用類別的名稱。

表 2-1 的名稱確實容易造成混淆，本體最基本的要求應該是眾人的共識，才能夠據以溝通。但是這些不同的名稱本身卻使得最基本的本體共識都難以達成，因此，本體的成份可以簡化成三組名稱。當本體以自然語言表示時使用 concept、attribute、value 與 instance；在物件導向環境或本體工具中，使用 class、slot、facet 與 instance；在中文環境下，則使用概念或類別、屬性、值與實例等；至於關係則是指概念與概念之間的 isa 或 partof 關係 (溫瑞烘、莊謙本；2004)。

總結而言，哲學思想的 Ontology 是一種存在的理論，稱為本體論，包含萬事萬物存在的探究，近代的 ontology 被認為是知識的基本架構，是依據本體論所發展出來的事物的表徵，本體論是一個理論，本體是一個由概念到實例的表徵，可以說是一個產品，兩者有顯著的區隔，但是 ontology 實際上的基礎還是在於哲學上的 Ontology 理論，兩者之間最主要的差異在於亞里斯多德的哲學本體論，在十個概念範疇之間並沒有特別強調相互之間的關係 (relation)，而今日的本體卻極為強調概念與概念之間的關係 (Schulze-Kremer, 2002)。

第三節 本體的應用

一、本體在溝通上的應用

個人、組織與軟體系統之間都需要相互溝通，但是由於背景與需求之不同，導致對同一件事情的觀點也有差異。因此應該要減少或去除概念上或名詞上的混淆，以達成共同的瞭解，這就是要建立本體的最主要理由 (Uschold & Gruninger, 1996；阮明淑、溫達茂，民 91)。

Noy & McGuinness (2001)也是從溝通的基礎上，認為本體可以延伸到下列的應用：

- 1、本體可以提供人們或軟體代理人對於資訊的共同了解：由於人們對於某個名詞的認知可能不同，導致不同的解讀，影響到人們的相互溝通與資訊的流通。例如在全球資訊網上，許多與醫藥相關的網頁都會提供有關醫藥的資訊，假如每一個網站都是基於相同的本體，則所提供的訊息都會採用相同的名詞，且其名詞的定義也是相同，對於大眾而言都會有共同的資訊傳達，對於資訊搜尋的軟體代理人而言，也可以獲得較有意義的資訊 (Musen, 1992; Gruber, 1993)。
- 2、本體有助於領域知識的重複使用：在人類知識發展的過程中，經常是利用舊有的知識來發展新的知識。因此，知識的重複使用對於知識的擴展與累積，佔有極為重要的地位，尤其在二十一世紀的今天，各種知識再也不能完全獨立。跨領域的知識整合變得極為普遍，如果對於某一個領域的知識，都是採用一致的格式或概念，對於知識的傳達與表現也必定有相當的助益。
- 3、清楚的表達領域的知識：本體除了是名詞的集合與定義之外，還將其關係明顯的表現出來，並以正規化的方法來加以描述。因此本體就是該領域的知識基礎，是初學者瞭解該領域的起點，對於想要擴展該領域的知識與應用的專家，也有極大的幫助。

4、將領域性知識與操作性知識分開表示：本體是用來表示某個領域中的概念知識，也就是領域性知識，而這個概念知識可以在不同的領域之中應用，也就變成操作性知識，因此，本體可以將領域性知識與操作性知識加以分開表示。

前述的這些應用，不管是人與人的溝通、人與機器的溝通、代理人之間的溝通等，都必須植基於共同的基礎，也就是有共通的字彙，而本體是最好的溝通工具。

二、本體在知識工程上的應用

建立本體的最終目的是為了應用，因此 Knowledge Based Systems, Inc. (1994)就以知識工程上的應用為例，來說明建立與應用本體的理由如下：

- 1、標準化 (standardization)：在一個知識工程計畫中，往往會牽涉到許多不同領域的工程人員，這些不同領域的人員經常要相互溝通與共同合作，才能完成一項計畫。相互溝通與共同合作必須要在一致的基礎上進行，本體的建立就是要給予所有術語或名詞標準化的定義，以便建立共同的基礎。
- 2、再使用 (reusability)：在工程應用上，經常有許多的工作是一再的重複，若每次均需從頭開始建立工作模式，既浪費資源也減低效率。所以建立本體就是為了節省重複塑模的花費，一旦某一個領域的本體被建立之後，就如同決定了該領域的標準，而可以拿來重複使用。

三、本體在全球資訊網的應用

McGuinness (2002)的研究指出，發展小型且簡單的本體，並不需要多大的花費，甚至還可以從網路上的本體儲存庫中取得，這種小型本體大多應用在全球資訊網頁上，說明如下：

- 1、提供控制的字彙：本體可以提供一般使用者、資料庫管理者、與程式設計者等，利用其字彙作為命名的依據，例如變數名稱、資料庫綱要與應用程式界面的名稱等。
- 2、作為網站架構：本體就是一個分類系統，利用這樣的分類作為網頁的架構，可以一層一層的擴展，不但方便網頁的管理，而且可以讓使用者很快的瞭解其知識架構，也很容易找到要瀏覽的項目。
- 3、提供瀏覽支援：整個網頁的內容可以利用本體的名詞加以標記，形成後設標記 (meta-tagged)，也就是網頁內容的摘要標記，有助於讓使用者瞭解網頁的重點內容，提供網頁的瀏覽支援。
- 4、提供擴充搜尋支援：本體的架構非常適合應用在搜尋的擴充上，當使用者想要搜尋某個名詞的定義及其相關的內容時，就可以透過本體的架構，找出其擴充的內容。
- 5、釐清意義：當一個名詞出現在不同的地方，而且又有不同的意義時，可以透過本體的階層架構，釐清名詞所代表的意義。

四、本體在生物資訊界的應用

Schulze-Kremer (2002)指出，在 1997 年之前，生物資訊界與分子生物界幾乎不知道本體這個名詞。但是在網際網路盛行之後，許多和生物相關的資訊，如基因的、細胞的、結構的、性質的、及其他型態

的生物資訊來源眾多，單就一項 DNA 而言，就有許多不同組織、不同範圍、不同目的的資料庫，提供相關的資訊查詢。但是這些資訊來源可能因為名詞的差異，例如同義詞、別名、慣用語等；或因語法的差異，例如語法結構、拼音；或因語意的差異等，造成查詢結果的混亂，使用者必須花費極大的心力加以篩選，更不用說是資料的共享與整合了。因此，在生物資訊界與分子生物界，都有共同的需求，希望建立共同的溝通標準，將本體與生物資料採礦、生物資訊整合、生物資源共享、與其他的應用聯結在一起。

Stevens, Wroe, Lord, 與 Goble (2004)也指出，生物資訊一直都受到極大的重視，可能是因為生物資訊的資料量大、複雜性高、反覆無常、異質性也大，而且過去累積的生物資訊都較為分散，由於缺乏共同的字彙，導致生物資訊在名稱上、實體上都有很大的差異。但是過去數年來，生物資訊界已經逐漸利用本體的概念，包括：

- 1、利用本體定義資料庫綱要或知識基礎：本體的概念、屬性與關係等，可以作為定義資料庫綱要的基礎。
- 2、利用本體查詢生物資料庫：本體本身的概念架構就包含分類的機制，生物資訊本體的內容就可以作為社會大眾查詢之用。
- 3、利用本體作為異質生物資訊之間相互操作的中介：不同的系統平台、不同的語言平台、或不同的生物資訊之間的相互溝通，需要利用本體作為中間的操作介面。
- 4、利用本體作為智慧型搜尋生物資訊的鷹架：利用本體所具有的概念、關係與同義詞等，可以提升搜尋回覆率與精確度。
- 5、利用本體作為相關社群的共同參考依據：生物資訊的專有名詞很多，對於許多名詞的解釋也有極大的變異。因此，生物資訊相關

的社群需要有一個共同的參考基準，就是生物資訊本體。

- 6、利用本體作為資料庫或技術文獻的語意註記：一般而言，生物資訊的資料庫綱要或內容未必有語意的註記，本體的概念與架構就可以當作語意的註記。

五、本體在軟體工程的應用

軟體的重複使用有助於軟體品質與生產力的提升，要達到軟體的重複使用先要設計出可重複使用的軟體資材 (assets)，再設計出新的軟體來重複使用這些軟體資材。在設計過程中，本體可以扮演軟體重複使用的重要角色，因為本體可以促進軟體發展者之間的了解、也可以作為軟體設計的規範、又可以改進資訊的存取。雖然本體對於軟體的重複使用有許多助益，但是最大的缺點是軟體工程中，沒有任何方法可以將本體融入軟體設計程序中。因此，Falbo、Guizzardi、Duarte 與 Natali (2002)建議以領域工程 (domain engineering)與軟體工程 (software engineering)平行且交叉的處理模式，來加強軟體的重複使用，其程序如圖 2-3 所示。

在領域工程方面，首先要進行領域本體的分析，以獲得領域的本體模式，再進行軟體架構的發展，以獲得結構模式，接著發展可重複使用的軟體資材，最後將軟體元件儲存到儲存庫中。在這些程序中，領域模組化、結構模組化、與元件儲存庫等，都是以本體作為基礎。在軟體工程方面，領域模式可以作為系統分析的輸入，結構模式也有助於系統設計，而可重複使用的軟體元件更可以提升系統建構的品質與效率。

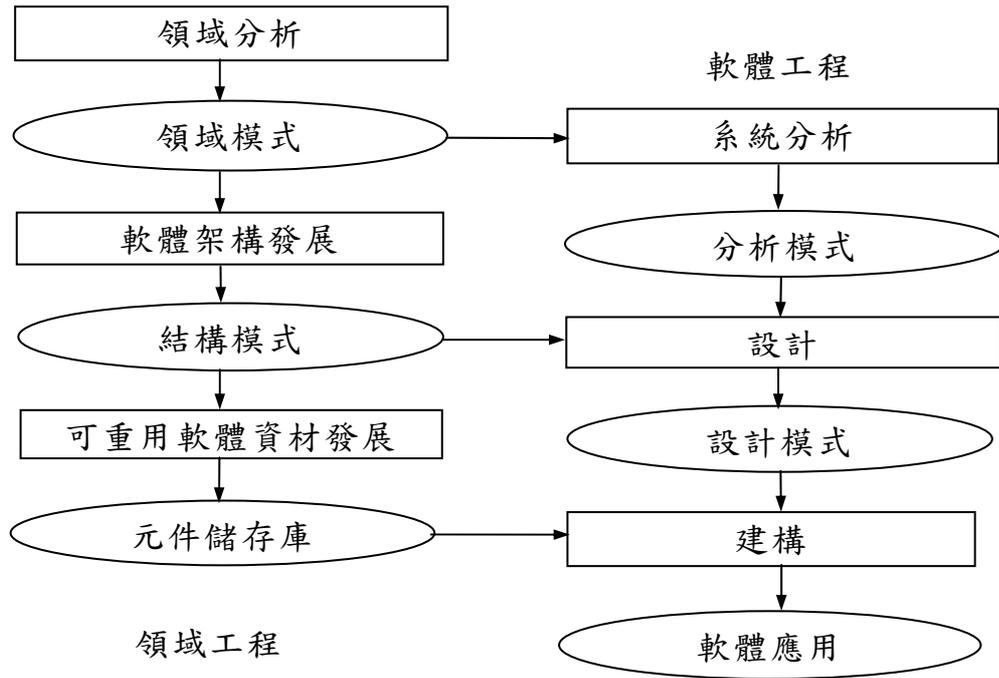


圖 2-3 以本體為基礎的軟體工程處理模式 (資料來源：Falbo、Guizzardi、Duarte 與 Natali(2002))。

六、本體在語意網的應用

本體的應用層面極為廣泛，已經擴展到各行各業的應用，但是最重要的應用應該是下一代的網際網路，也就是語意網的應用。今天網路上的搜尋引擎已經極為方便，只要使用者輸入關鍵字查詢，往往可以得到成千上萬筆資料，但是這些資料是否為使用者所需要，仍然是個未知數，使用者還要花上許多時間一一過濾。這是因為網頁上或者儲存庫中的資料，通常都欠缺註解資訊，也就是欠缺語意資訊，只能提供人類閱讀，而無法讓機器瞭解。但是這種情況在未來將會改變，因為目前有許多標記語言 (markup language)，如 XML (eXtensible Markup Language) (XML, 2000)、RDF (Resource Description Language) (Lassila, 1998；Lassila & Swick, 1999)、RDFS (Resource Description Language Schema) (Brickley & Guha, 2000)、DAML (DARPA Agent

Markup Language)(Hendler & Mcguinness, 2000)等，逐漸被使用者所接受，利用這些標記語言和本體相互結合，可以使目前的文字網頁變成語意網頁 (semantic web)，就可以讓網頁內容提供機器閱讀並瞭解。

在資訊擷取的應用方面，也常利用本體的語意來作語意搜尋，與一般傳統的資訊擷取技術利用統計方法比較起來，本體在語意搜尋上扮演兩個角色，一是透過本體的概念階層架構將搜尋的結果加以分類，另一是透過本體的語意來作推論式的延伸搜尋，其目的是要改善搜尋的回覆率與精確度，目前的實際應用情形如下：

- 1、Corese：Corese (Acacia-Corese, 2007)是 Conceptual Resource Search Engine 的縮寫，是一個植基於概念圖 (conceptual graphs)和 RDF 的語意搜尋引擎，可以由 RDFS 與 RDF 的文件中進行語意搜尋，其應用環境涵蓋知識管理、eLearning 與 eHealth 等。Corese 以 Java 實作，是一種開放式的軟體。
- 2、OntoSeek：OntoSeek系統 (OntoPortal, 2007)採用有限的語言表示法來描述結構化的資源，例如黃頁或產品目錄，並透過大型語彙本體WordNet作媒合，可以改進資訊擷取的回覆率與精確度。
- 3、SHOE：SHOE (Simple HTML Ontology Extensions) (SHOE, 2007)利用類似XML的語言來標記網頁，並利用本體作為背景知識，再利用人工智慧的技術來搜尋網頁，儘管所標記的網頁有限，但是已經證明語意搜尋的效能。
- 4、XSearch：XSearch (CiteULike, 2007; Cohen, Mamou, Kanza, & Sagiv, 2003)是一種語意搜尋引擎，使用XML正規化的本體作為基礎，並且搜尋由XML表徵的文件中，傳回語意相關的文件。

另外，本體也極為適合應用在語意式的工作媒合系統上，將於第三章中專章探討。

七、本體在知識管理的應用

資訊科技 (information technology ; IT) 與知識管理 (knowledge management ; KM) 兩者密不可分。IT 扮演轉換知識、連結知識、進而創造知識的角色，使得 KM 系統更形豐富，而這些功能的達成可以使用本體作為知識管理系統的基礎，其應用包括 (Abecker & van Elst, 2004)：

- 1、以本體支援知識的顯示：利用本體作為知識顯示的架構，再加上適當的瀏覽與進階搜尋機制，可以讓使用者更方便的存取知識。
- 2、以本體支援知識搜尋、擷取與個人化：建立領域本體 (domain ontology) 可以擴展傳統的關鍵字搜尋到先進的語意搜尋，有助於知識的擷取，而個人的學習歷程檔案本體 (profile ontology) 也可以讓知識管理系統做到個人化的知識管理。
- 3、以本體作為資訊蒐集與整合的基礎：利用領域本體與資訊本體 (information ontology) 作為資訊擷取、資訊萃取、與資訊整合的基礎，以便產生新的知識。

本體還可以應用在虛擬組織的內容管理上，Mika, Iosif, Sure 與 Akkermans (2004) 發展了一個虛擬企業有限公司叫 EnterSearch，以網頁方式傳播資訊科技與能源方面的知識，可以提供研究報告、報導、或專書等。EnerSearch 原來的設計和大部份企業的網站沒有兩樣，參觀者以主題分類方式瀏覽，所有出版品均以表格方式呈現，並可以選

擇不同的排序方式，同時還有一個關鍵字搜尋引擎 EnerSearcher。但使用者並不滿意這樣的安排，主要的問題出在搜尋資訊的效率與精確度。使用者希望系統能夠引導他們快速的找到需要的資訊，並排除不相關的資訊。除此之外，固定式的網頁架構限制了瀏覽的選擇，這種完全由供應端決定一切的作法，和需求端為主的思考不相吻合，違背了企業界顧客至上的信條。因此，在 1999 到 2002 年之間所進行的「歐洲本體到知識計畫」(European On-To-Knowledge project)，發展了以本體為基礎的中小企業解決方案，包括本體語言、本體工具、與本體發展方法等，希望應用在中小企業的知識管理系統之中，以提升競爭力。為了確認此解決方案的成效，EnerSearch 加入了該計畫並轉型成為一個知識管理系統，目的在組織企業的知識，以減少知識工作者與內容管理者的負擔。EnerSearch 有一個資訊萃取程式 OntoExtract，將萃取到的資訊建立成本體。儘管本體的正規化不高，所建立的關係也不佳，但是因為本體的潛力，使得語意式的搜尋變得可行，改進了資訊擷取與網頁顯示的效率，也讓使用者的滿意度大大提高。

本體還可以被使用在知識入口網站的建置上，OntoWeb (Oberle & Spyns, 2004)是歐盟的語意主題網頁 (OntoWeb, 2007)，也稱為本體網頁，於 2004 年 5 月完成建置，是一個以本體為基礎的知識入口網站，也是一個知識管理、電子商務、與生物資訊的資訊互換平台，其互換的機制為 RDF+RDFS。OntoWeb 的核心概念有兩個，一個是 DOGMA 本體 (Developing Ontology Guided Mediation for Agents)，另一個是語意入口網 SEAL (SEmantic portaAL)。DOGMA 本體由兩個部份組成，本體庫 (ontology-base)與本體承諾層 (commitment layer)，前者是領域的概念集合，後者是領域的規則；SEAL 則是由不同的資料來源中萃

取資訊，並儲存到資訊儲存庫中，並藉由本體架構提供瀏覽、查詢等功能。

八、大型本體的應用範例

早期所發展的本體大都屬於大型的本體，用來提供某個領域標準化的字彙，作為溝通的基礎。例如國際疾病標準分類 (International Classification of Diseases) (WHO-ICD, 1992)；聯合國標準產品與服務分類 (the United Nations Standard Product and Services Classification) (UNSPSC, 1998)；美國病理學家學院 (College of American Pathologist) 所發展的系統化醫藥名詞集 (Systematized Nomenclature of MEDicine; SNOMED)，已經成為國際健康術語標準 (IHTSDO, 2007)；國家醫藥圖書館 (National Library of Medicine) 所發展的統一醫學語言系統 (Unified Medical Language System) (NLM, 2007)，都是極為成功的本體應用範例。

本體的應用除了在醫學領域大放光芒之外，也在軍事領域佔了一席之地，例如美國夏威夷太平洋指揮部的虛擬資訊處理代理人研究 (Virtual Information Processing Agent Research; VIPAR) (Potok, Elmore, Reed, & Samatova, 2002)，就是以本體作為基礎，每天以完全自動化的方式，萃取17種網路英文日報的重要資訊，其做法是先分析每一種英文日報的HTML格式，再發展出不同的描述本體作為基礎，再由智慧型代理人自動萃取資訊，節省許多的人力與物力。我國中央研究院中英雙語知識本體詞網 (中研院, 2007)，也是以本體為基礎的應用。

第四節 本體的分類

一、以資訊處理觀點來做本體分類

Guarino (1997)以資訊處理的觀點，將本體分成四個類型：

- 1、上層本體 (top-level ontology)：最具有一般性概念的本體，屬於最高階的本體，也可以稱作高階本體。這種本體並不是針對某個特定的領域，而是各個領域都需要用到的概念。因此有較為抽象的內涵，通常和人類的日常生活息息相關。例如有關時間、空間、語言、文字等的本體等，都是屬於上層本體。
- 2、領域本體 (domain ontology)：領域本體是具有特殊概念的本體，係針對某一個特定領域的知識內涵所建立。通常和人類的學習生活、經濟生活、社會生活、或就業生活等相關。例如各種學科、職業、工程、醫藥、醫學等領域的本體。
- 3、任務本體 (task ontology)：任務本體也是具有特殊概念的本體，通常和人類的工作或職業相關，用以描述特定的工作任務或工作活動為主。例如銀行交易、醫師診斷、教師教學等本體，都是屬於任務型的本體。
- 4、應用本體 (application ontology)：結合了領域本體與任務本體兩者，前者提供領域相關的字彙或名詞，屬於理論的層面，後者則以執行特定的任務為主，屬於應用的層面，大部分本體的發展都是以應用為目的。

二、以主題觀點來做本體分類

Uschold (1996)則以本體的主題來劃分本體，也就是將表徵本體置於最上面一層，第二層是領域本體，最底下一層則是問題解決本體：

- 1、表徵本體 (representation ontology)：也稱為後設本體或稱詮釋本體 (meta ontology)，是專門用來描述本體的本體，以提供結構與字彙給領域本體與問題解決本體。
- 2、領域本體 (domain ontology)：針對某一個領域的一般性概念進行描述，和特定的問題或特定的任務無關的本體。領域本體就是該領域的知識組成。
- 3、問題解決本體 (problem-solving ontology)：也稱為任務本體 (task ontology)，或方法本體 (method ontology)，通常描述可能產生的問題與問題解決的方案、或執行任務的方法等。

三、以知識重複使用的觀點來做本體分類

Mizoguchi 與 Ikeda (1996)則以知識重複使用性高低的觀點，將本體劃分成四種：

- 1、通用本體 (general/common ontology)：或稱為一般本體，和日常生活事務有關的本體，例如時間、空間、事件、事務等之本體，由於這些項目沒有領域之分，因此，本體的重複使用性最高。
- 2、領域本體 (domain ontology)：和特定領域有關的本體，主要是提供該領域的一般性知識，其重複使用的特性低於通用本體。
- 3、任務本體 (task ontology)：用來描述問題解決的架構，包括任務執行時的推理、任務安排、資源配置、任務目標與限制等，由於任務不同，其問題解決方式也一定不同，因此任務本體的重複使用性也不高。

4、工作場所本體 (workplace ontology)：以特定的問題解決為主，本體的內容應該包括問題解決所需要的知識、處理程序、重要事件，資源配置、效能、產出、可能碰到的問題與替代的解決方案等。由於工作場所本體會因為領域不同而有極大的差異，因此，其重複使用性最低。

四、以普遍性的觀點來做本體分類

Knowledge Based Systems, Inc. (1994)以普遍性的程度，將本體分成三個層次，按照普遍性由高至低說明如下：

- 1、領域本體 (domain ontology)：普遍性最高，主要是提供該領域的一般性資訊。例如半導體製造領域的領域本體，將包括產品種類、製造技術、材料與工具等之本體；資訊領域的本體可以包括硬體領域與軟體領域，再往下繼續分類。
- 2、實務本體 (practice ontology)：次普遍性的本體為實務本體，是領域本體的延伸，繼承了領域本體的所有性質，並且是該領域中所有相似的實務所組成的本體。例如半導體製造領域中類似的生產線，就可以共同發展成為實務本體。
- 3、特定場所本體 (site-specific ontology)：最缺乏普遍性的本體，是實務本體的延伸，當然也是領域本體的延伸。例如生產特定半導體晶片的生產線，就可以發展該生產線的本體。

相較於其他許多本體分類而言，這種分類方式係以領域本體的普遍性最高，實務本體是領域本體中的特定實務，而特定場所本體又是某個實務本體中的特定任務解決本體。因此，都是屬於領域本體的延

伸，對於一般性的本體，例如時間、空間、事務、地點等，並未著墨。

五、以知識工程的觀點來做本體分類

van Heijst, Schreiber, 與 Wielinga (1997)則以知識工程的角度，按照知識概念化的程度來劃分本體：

- 1、領域本體 (domain ontology)：針對個別領域所發展的本體，可以作為建構知識本位系統的基礎知識或資訊架構，包括領域共通化與標準化的名詞與定義、領域中特定的名詞、定義與方法等。由於一個知識本位系統的建構，需要不同背景的人員共同合作才能完成，其溝通的基礎就在於領域本體中的名詞與定義，因為領域本體不僅僅是文獻或規範而已，而且還可以作為品質控制的機制。
- 2、核心本體 (core ontology)：在知識工程系統的建構中，知識成份的重複使用是極為重要的一個考量。而一個知識工程系統之中可能包含數個相近領域的知識工程，這些領域必定要用到許多共同的本體，這些共同的本體就稱為核心本體，包含共同的名詞、定義與方法等。例如在法律上有許多不同的專業，每一種法律專業之中有共同的知識，這些知識的組成就是核心本體。又如在醫學的專業上分科很細，但不管是哪一個分科，都會有共同的醫學知識，這些共同知識所組成的本體也是核心本體。

六、以本體和應用系統之間相依的關係來做本體分類

Zajac (2001)以本體和應用系統之間相互依存的關係來分類，可以分成顯性本體 (explicit ontology)、隱性本體 (implicit ontology)與中

性本體 (neutral ontology)，敘述如下：

- 1、顯性本體：顯性本體是指本體以單獨存在的方式顯性儲存，應用系統需要時才存取本體。
- 2、隱性本體：隱性本體則是本體以隱藏的方式存在，也就是和應用系統綁在一起。
- 3、中性本體：中性本體則是本體可以單獨存在或與應用系統結合。

Zajac 進一步根據本體的應用層面，將本體劃分成工程本體 (engineering ontology)、語言本體 (linguistic ontology) 與網頁本體 (web ontology) 三者。工程本體是一種隱性本體、語言本體是一種顯性本體、而網頁本體則是屬於中性本體，說明如下：

- 1、工程本體：工程本體是植基於真實世界的觀點，主要目的是在系統設計階段時，協助資訊系統設計人員設計應用系統的綱要 (schema)。例如產品的模式化綱要、資料庫的綱要、資料庫整合的後設綱要 (meta-schema)、電子商務的交易綱要等。利用工程本體的高度正規化屬性，就可以將本體當成軟體工具一般，來設計事件 (event)、狀態 (state) 與行動 (action) 等，甚至於自動產生程式碼。因此，工程本體是以隱性方式整合在應用系統之中。
- 2、語言本體：語言本體是由概念、概念的定義與概念之間的關係等所組成，透過文字化方式輸入這些概念、定義與關係等，並且儲存在本體儲存庫中，就可以建立語意的表示。應用系統需要進行語意處理時，才去存取語言本體的儲存庫，並且根據定義進行語意解譯或媒合。因此，語言本體通常是和應用系統脫勾，不像工程本體是和應用系統整合在一起，也就是屬於顯性的本體，例如許多線上的辭典、索引典等。

3、網頁本體：網頁本體通常比工程本體與語言本體還要大，但是在結構上則比較簡單，主要是提供分類結構的索引或連結、網頁的進階搜尋、使用者介面的設計等。因此在結構上，通常採用主題式的分類，而不是採用概念式的分類，例如一本有關汽車的書，如果按照主題式的分類，則這本書應該在汽車分類之下，但如果按照概念式的分類，則這本書應該是在書的分類之下。在角色上，網頁本體可以像工程本體一樣，作為設計的依據或和應用系統整合在一起，也可以像語言本體一樣，單獨儲存在本體儲存庫中，以便應用系統隨時呼叫，也就是說，網頁本體是一種中性本體。

七、以正規化的程度來做本體分類

Poli (2002)以概念與概念之間關係表徵的正規化程度，將本體分成三種型態：

- 1、描述本體 (descriptive ontology)：描述本體一方面是指組成某一個領域的所有資訊的集合，描述本體另一方面也是指本體的表徵是以敘述的方式存在，也就是以文字敘述的方式來描述所有領域資訊的集合。
- 2、正規本體 (formal ontology)：由於描述本體是以自然語言的方式來表示某一個領域的資訊。正規本體則是進一步將描述本體加以分離、過濾與整理。例如可以根據事物 (thing)、處理(process)、型式 (form)、整體 (whole)與部份 (part)等性質來選擇資訊，進一步做分類篩選、編碼等，就成了正規本體。
- 3、正規化本體 (formalized ontology)：正規化本體則是進行更嚴格的編碼，並加以評鑑本體的表示性 (expressive)、認知性

(cognitive)、與計算性 (computational)。也就是說，正規化本體必須要能清楚的表達其內涵、也要具有知識性、以及可以利用電腦加以處理。

八、以本體在知識庫系統中扮演的角色來做本體分類

Studer, Fensel, Decker, 和 Benjamins (1999)以本體在知識庫系統中所扮演的角色來分類本體：

- 1、領域本體 (domain ontology)：領域本體是某一個特定領域的知識組成，例如電子、醫學、機械、或數位領域等。
- 2、一般本體 (generic ontology)：也稱為常識本體 (commonsense ontology)，是關於一般性的知識，也就是和領域無關的知識，例如時間、空間、狀態與事務等。
- 3、表徵本體 (representational ontology)：表徵本體也是和特定的領域無關，係用來陳述本體應該如何表示，可能有許多不同的名稱，例如框架本體 (frame ontology)定義了本體的框架式表示法，包括框架 (frame)、屬性 (slot)、屬性限制 (slot constraint)等。

九、以使用性與再用性來做本體分類

發展一個本體需要耗費人力物力，因此許多的本體發展方法論都會先尋求是否有現成的本體。利用現成的本體直接拿來應用，可以省下許多發展本體的時間與金錢，所以本體的再使用也是許多研究者極為重視的課題。Klinker, Bholal, Dallemagne, Marques, 與 McDermott (1991) 以本體的使用性與再用性，將本體劃分成圖 2-4 的九種本

體，最底層的表徵本體是本體中的本體，也就是其他八種本體的綱要，任何本體的發展都必須以此綱要作為規範。由於表徵本體和任何領域都毫無關聯，很少融入應用系統之中，其使用性最低，但是再使用性最高。應用領域任務本體是為了某個特定領域的特殊任務而發展，必定會和應用系統結合，其使用性最高，但是其再用性反而變得最底。也就是說，越一般化的本體，其使用性越低，但再用性越高。



圖 2-4 以使用性、再使用性來分類本體 (資料來源：Klinker, Bhola, Dallemagne, Marques, & McDermott, 1991)

十、以本體的共享性來做本體分類

本體首要的任務就是要讓領域內外的人，對於領域知識的描述與宣告有所共識，但是要達到一致的共識是一件不容易的事。有些本體被特定的社群發展出來，以適應特定的應用。這種本體的出發是以解決特定的任務為主，其使用頻率可能只有很少的次數，共享性並不是最重要的考慮因素。但是有些本體具有全國性的標準，例如職業分類、產品分類、服務分類、工作分類與線上詞典等。這些本體被發展出來，確實獲得大眾的共識與應用。Tolksdorf (2007)就以本體的共享性 (shareability)將本體劃分成：

1、個人本體：個人本體是個人所發展的本體，只能反應個人在某個

特定領域的觀點，這種本體即使被發表出來，其衝擊也是有限。但是在發展的過程中，如果能夠遵循本體發展的理論，仍然可能獲得領域內同儕的支持。

- 2、應用本體：過去有許多應用本體被發展出來，大部份都是研究計畫或特定專案的產品，這種應用性的本體反應了研究團隊或研究社群的觀點，即使這些本體被發佈在網際網路上，共享性和再用性也都受到限制。這是因為應用本體和原來的計畫可能是緊密相依的，目前大部份的領域本體都是屬於這一個層次。
- 3、開放式本體：開放式本體是透過大眾或整個社群所共同發展，是一種合作式發展的本體，已經整合多數人的想法，並獲得多數人的同意。這種本體大都是屬於大型的本體，並且和大眾有所關連，其重要性和應用性都很高，才能引起大眾的參與，例如對醫學界與對社會大眾都極有貢獻的基因本體 (Gene Ontology)。
- 4、標準本體：大都由公立機構或重要的組織所發展，通常會透過大眾的確認程序，以期獲取大家的共識，是一個領域共同的標準，因此被稱為標準本體。大多數的標準本體是在電子商務領域，例如聯合國標準產品與服務編碼本體 (UNSPC, 2007)、提供全球供應鏈標準分類的 RosettaNet 本體 (RosettaNet, 2007)與北美工業分類系統本體 (NAICS, 2007)。

十一、本體分類方式的彙總

根據前述的探討，本體的分類方式相當的多樣，彙總如表 2-2 所示，儘管分類的方式很多，但是比較普遍的兩個分類為上層本體與領域本體。上層本體是屬於一般化的本體，是每一個領域都需要的本

體，其應用性較低，但再使用性較高。領域本體是一種特殊性的本體，係針對某一個特定的領域所發展，領域本體的範圍可大可小，可以小到某一個問題解決所需要的本體，也可以大至到一個職業群集的本體，端賴應用層面的大小而定。

表 2-2 本體分類方式的彙總

分類觀點與提出者	分類的方式
使用性與再用性(Klinker, Bholra, Dallemagne, Marques, & McDermott, 1991)	應用領域本體、應用領域任務本體、領域本體、領域任務本體、一般領域本體、任務本體、上層本體、一般本體、表徵本體
普遍性(Knowledge Based Systems Inc., 1994)	領域本體、實務本體、特定場所本體
本體主題(Uschold, 1996)	領域本體、問題解決本體、表徵本體
知識重複使用(Mizoguchi & Ikeda, 1996)	工作場所本體、任務本體、領域本體、通用本體
資訊處理(Guarino, 1997)	上層本體、領域本體、任務本體、應用本體
知識工程(van Heijst, Schreiber, & Wielinga, 1997)	領域本體、核心本體
本體角色(Studer, Fensel, Decker, & Benjamins, 1999)	領域本體、一般本體、表徵本體
本體和系統依存關係(Zajac, 2001)	顯性本體、隱性本體、中性本體

本體應用層面(Zajac, 2001)	工程本體、語言本體、網頁本體
正規化程度(Poli, 2002)	描述本體、正規本體、正規化本體
本體共享性 Tolksdorf (2007)	個人本體、應用本體、開放式本體、標準本體

資料來源：本研究

不管本體的分類方式如何、組成內容如何，本體的概念在真實世界中一定要存在，一個不存在的概念無法加以定義，也找不到子概念，更不可能有屬性、實例與關係。所有的概念都需要有屬性，而且概念與概念之間要有相互關係存在，才能夠將兩個概念相連在一起，這樣的關係就是 isa 或 partof 的關係。

第五節 本體的發展

一、本體發展的原則

建構本體對於許多應用系統的發展，是一個極為重要的步驟，但是目前的本體發展方法似乎還停留在藝術的層次，尚未達到科學的方法。儘管如此，由許多過去的本體發展經驗中，仍然具有相當的參考價值。首先本體是由許多概念所組成，而概念是一個抽象的東西，其命名往往會影響到人們對概念的理解。因此，概念的命名必須要遵循一定的規則，才能夠使概念的名稱有一致的基礎，不但容易將概念的名稱插入本體之中，也比較容易辨別是否有相同或類似的概念。Schulze-Kremer (2002)更進一步提出了概念的命名原則如下：

- 1、使用單數型態：完全使用單數型態來命名，避免使用複數型態而導致誤解，並且使用小寫字母代表類別，第一個字母使用大寫來表示實例，縮寫字全部用大寫字母。如果本體需要帶入本體工具之中，還要注意本體工具是否有特別的限制，例如是否可以使用引號、減號、或底線等。
- 2、儘量使用現成的、約定成俗的名稱：如果沒有現成的名稱，則最多以四個字組成名稱，每一個字也應該使用簡單的名稱，使其淺顯易懂。
- 3、子類別的命名：可以用其父類別的名稱之後附加名稱，以利辨別相互關係。

Uschold 與 Gruninger (1996)以設計企業模式本體的經驗，提出本體的設計原則如下：

- 1、清晰性 (clarity)：本體要能夠有效的傳達其意圖，就必須要清楚的表達所有的概念、實例與關係等，絕對不能有絲毫的模糊空間，才不至於引起擴張的解釋。對於可能造成誤解的地方，最好能夠詳細說明，並提供範例。對於整體的架構，應以正規理論來加以表達，但還是需要提供自然語言的定義與文件的說明。
- 2、一致性 (coherence)：本體需要具備內部一致性，包括理論基礎的一致性、名詞定義的一致性、相互關係的一致性、文件的一致性等。
- 3、延展性 (extensibility)：本體提供某個領域的概念架構、名詞字彙與定義，有其特定的應用範圍。但本體還是應該具備延展性，也就是在既有的基礎上，可以擴充其字彙與關係，來延展其應用範疇。要達到延展性，就必須使本體合乎兩個準則：不做過多的承

諾與避免編碼誤差。前者是指在設定的應用範圍內發展本體，而不要過度的擴張，後者則是將概念與關係轉換成正規表示法時，要避免編碼的錯誤。

Borgo, Guarino, and Masolo (1996)、Gomez-Perez, & Benjamins (1999)等，也提出了五個本體的發展原則，這些原則都是一般性的原則，在真正發展本體時，未必能夠提供實際的決策支援，但是卻可以提醒本體發展者利用這些原則來檢視本體的內涵：

- 1、本體中所有的類別必須要有所差異，也必須要相互分離，不要將具有相同或相近意義的概念放在同一階層或上下階層，而應該要以同義詞的方式表示。
- 2、階層架構的正確性與多樣化，以提升多重繼承機制的效率，與分類的多樣化。
- 3、減少相似概念之間的語意差異，並且將相似的概念放在同一組，以相同的述詞來描述。
- 4、所使用的名稱要採標準化的命名方式，以求一致，並且最好讓名稱具有意義，可以由名稱來瞭解大致的語意或概念。
- 5、減少模組與模組之間的聯結現象，如果一個類別有兩個以上的超類別，這可能造成應用上的困擾，或是造成無窮的參考。

綜合前述，本體發展的一般性原則可以歸納如下：

- 1、有意義的命名方式，舉凡概念、關係與屬性等的命名，都應該採用大眾化的名稱，避免太生澀或學術性的名稱，並且讓超類別與子類別的名稱具有聯結關係。
- 2、確認關係的適合性，應該分別由上而下、由下而上，並且隔一層、

隔二層等依次檢查概念的層次關係與意義。

- 3、確認概念的一致性，也就是要檢查同一層的概念在意義上是否有落差現象，避免造成概念所在的層次不正確。
- 4、確認本體提供足夠的資訊，本體是為了應用而發展，勢必會和應用系統結合。因此，必須以應用的層面來考慮是否可以提供足夠的資訊，而不致於太過簡略或瑣碎。

二、發展本體需要克服的困難

要建立一個最佳的本體極為不容易，有太多的困難需要加以克服，這些困難有些是源自於建構本體過程中的疏失，有些則是對於應用領域認識不夠所造成。整體而言，在發展本體時，需要克服的困難如下 (Schulze-Kremer, 2002)：

- 1、超類別與子類別之間的關係不易確認：由於本體的發展方法中，並沒有公論最佳的方法可以確定何者才是最佳的子類別。因此可能會造成任意決定子類別，這就可能導致本體的一致性不夠，因此必須詳加檢查類別之間的關係。
- 2、遺失本體的組成元素：就是在發展過程中遺失了類別、屬性、或關係等。本體中的類別由上而下應該是由抽象而逐漸具體，但是如果中間缺少了類別，就會造成本體重要元素的遺失。例如由「動物」直接到「人」，中間就缺少了重要的類別「哺乳動物」。
- 3、關係的混淆：類別之間的關係無法確認是一對一、一對多、或多對多的關係，也會造成概念的遺失，或是把關係與屬性誤認為同樣一件東西。
- 4、過於詳細：本體中最重要的元素為概念，最底層則為實例，如果

本體想要嘗試列出所有的概念或實例，就會造成太過詳細、或有過多的元素等。

5、把重要的資訊當成是註解欄位或描述文字，造成重要的資訊遺失。

三、IDEF5 本體發展方法論

Knowledge Based Systems, Inc. (1994)指出，大型本體的發展需要具備相當的技能與經驗，主要是因為領域的知識往往缺乏完整的記載，這些知識只存在於少數專家的腦海中。宣告性的知識 (declarative knowledge) 還比較容易以顯性方式加以描述，但程序性的知識 (procedural knowledge) 就不容易以顯性方式來表徵。因此，一個大型本體的發展必須有計畫、有組織、有管理的循序漸進、重複思考、詳細討論、細心檢視與深切反省。過程中間需要領域專家的參與及團隊合作，整個發展程序的五大活動如下：

- 1、組織與定義計畫 (organize and define the project)：大型本體的發展，其應用層面必定廣泛，因此，通常以計畫的方式進行，包括組織計畫與定義計畫。
- 2、蒐集資料 (collect data)：發展本體需要廣泛的蒐集資料，並且也是一個持續且互動的過程。資料的來源可以由相關的文獻中分析、訪問該領域的專家、與觀察組織的活動或現象等。資料的來源則是和領域有關的知識，如名詞、名詞定義、概念、種類、物件、實例、關係、問題與解決方案等。
- 3、分析資料 (analyze data)：資料的分析是為了建立初步的本體，分析的項目包括：列出領域中的物件、確認在範圍之內的物件、將具有共同性質的物件加以分類等。

- 4、發展初步的本體 (develop initial ontology)：剛開始發展本體時，可能因為對於本體發展方法的運用不純熟，或是對於領域知識瞭解不夠透徹。因此可以先發展初步的本體，也就是由初步的概念開始，這些初步的概念會因為資料的獲得，或概念的釐清，而變成固定且成熟的概念。初步的概念包括初步的類別、初步的屬性、初步的值與初步的關係等，最後再畫出分類的圖示，以明確的表示本體的架構。
- 5、精練與確認本體 (refine and validate ontology)：初步的本體發展完成之後，還需要加以確認名詞及其定義是否合適、種類與關係是否正確。確認的方法可以透過蒐集更多的資料、並實際測試其正確性。

IDEF5 的本體發展方法包括五個大型的活動，每一個活動都需要許多人力的支援，這些參與的人力最好都具備領域的知識與本體發展的經驗，但在發展方法中並未提及如何進行資料的分析，也沒有正規化的步驟，也未談到和應用系統結合的方法。

四、TOVE 本體發展方法論

企業的營運與管理往往需要依靠資訊系統，過去的企業資訊系統主要以儲存企業的靜態資料為主。例如人事、薪資、庫存、生產、訂單與流通等資料，但是二十一世紀的企業，不但要能夠處理靜態的資料，還要能夠處理動態的資料，甚至是語意的資料。也就是說，整個企業的模式都應該是企業資訊系統的內涵，為了達成這樣的企業模式，Gruninger 與 Fox (1995)發展了「多倫多虛擬企業」(TOronto

Virtual Enterprise ; TOVE) , TOVE 的中樞是一個「常識企業模式」(common sense enterprise model) , 也就是一個企業本體 , 用來支援 TOVE 的運作。Gruninger 與 Fox 由發展企業本體的經驗 , 提出了 TOVE 本體發展方法論 , 也被稱為企業塑模方法論 (Enterprise Modelling Methodology) , 其本體發展的步驟如下 :

- 1、情境刺激 (motivating scenarios) : 本體的發展主要是基於應用上的需求 , 當企業碰到問題時 , 就必須要尋求解決的方案 , 這就是一種情境刺激的反應。解決企業問題的方案 , 可以是短期的問題導向解決 , 也可以是長期的根本導向解決 , 後者的解決方式必定和企業的本體知識有極大的相關。因此 , 為了解決企業的問題 , 較好的方式是建立企業的本體 , 作為企業資訊系統的基礎。
- 2、定義非正規的能力問題 (informal competency questions) : 本體的發展是先要瞭解應該解決那些問題 , 才能讓本體具有解決這些問題的能力。這些問題首先以非正規的方法、也就是以自然語言描述法來加以表示 , 就稱為非正規的能力問題。由於建構本體是一件耗時費力的工作 , 本體一旦建立 , 就必須要全盤解決企業的所有問題。因此 , 在形成非正規的能力問題時 , 必須要思考整個企業要解決的問題有哪些 , 思考的方向可以由企業的組織架構與企業活動兩方面著手 , 列出所有過去發生過的問題與未來可能發生的問題 , 並且列出每一個問題的解決方案 , 這些問題及其解決方案就是本體的範圍。
- 3、定義第一階邏輯規範-術語集 (specification in first-order logic: terminology) : 本體是針對領域的概念、概念的屬性與概念之間關係的一種表徵 , 所以在建構本體時必須先找出概念、屬性與關係等。要找出這些成分 , 必須要分析前一步驟所獲得的非正規能

力問題，以便定義需要那些概念、那些屬性與那些關係等，以便形成術語集

- 4、定義正規的能力問題 (formal competency questions)：一但本體中的物件、屬性與關係決定之後，進一步要以正規的方式加以規範，也就是要決定屬性的型態、值、基數或限制等，並以正規的法則來表示。
- 5、定義公理規範 (axioms specification)：公理是利用一階邏輯來規範名詞的定義與解譯的限制，公理就是能力問題的答案，也就是建立本體的最終目的。由於公理的定義方式不止一種，取捨的根據是要看公理的表達是否清晰，以及公理是否足以回答能力問題而定。表達愈清晰、回答能力問題愈強的公理，當然是較佳的公理。
- 6、完全理論 (completeness theorems)：TOVE 發展本體的最後一個步驟是要定義完全理論，也就是要定義達成能力問題的條件，只要本體能夠合乎所定的條件，就表示本體足以回答所定的能力問題，因此，完全理論是評鑑本體是否達成目的的依據。

TOVE 的本體發展方法論是由應用的角度出發，考量本體必須具有解決企業問題的能力，因此，首先要列舉出所有本體應該回答的問題，再據以發展本體，為了達到動態的應用，本體還需要有公理與推理機制。

五、骨架式本體發展方法論

Uschold與Gruninger (1996)指出，過去雖然有許多本體及其應用被發展出來，但是卻缺乏標準化的本體工程理論，因而提出了一個骨

架式 (Skeletal Methodology)的本體發展方法論，也稱為綜合式的本體發展方法論，步驟如下：

- 1、確定本體的目的與範圍：首先要確定本體的領域、目的與範圍，也就是要決定應用層面，是要應用在人與組織之間的溝通、系統與系統之間的相互操作、系統工程上的應用、還是在知識基礎上的應用？不同的應用型態會影響到本體的發展。因此，確定本體的目的之後，才能決定要不要發展到實例，還是只要發展到概念，要不要有推理機制等。另外，還應考慮潛在的使用者及使用的範圍。一般採用腦力激盪與分組的技巧來確定本體的目的與範圍。
- 2、建立本體：在確定目的與範圍後，就可以開始發展本體，過程中應注意三個面向，即抓住重點、正確編碼與整合現存的本體等。
- 3、評鑑本體：一旦建立本體後，必須就本體的整體架構進行評鑑，項目包括軟體環境、資訊量與應用需求等
- 4、建立文件：本體的最終目的是應用，因此應該建立完整的文件說明，包括發展理念、領域、概念、關係、階層架構與應用的指引等，讓有意應用此本體的使用者有所依循，對於後續的新增或維護，也有相當大的助益。

在骨架式的本體發展方法中，整合現存的本體是在新的本體發展完成之後才進行，其目的是要將過去所發展而且仍在使用的本體，融入到新發展的本體之中，也就是本體的整合。雖然兩個或多個本體的整合也是很重要的工作，但是以本體發展的角度而言，要從無到有發展一個新的本體，是一件耗時費力的工作，所以本體發展者應該先尋求是否已經有相同或相關的本體。如果可以採用現存的本體或參考舊有的本體，必定可以收事半功倍之效。另外，建立文件雖然是放在

最後一個步驟，並不是意味本體發展完成之後才來建立文件，而是在本體發展的過程中，同時建立文件。

六、軟體開發生命週期本體發展方法論

Lopez (1999)認為，許多基於本體的應用系統，其中有關本體的發展也可以採用 IEEE1074-1995 的標準程序來發展，也就是將應用系統的發展與本體的發展結合在一起，其程序如下：

- 1、軟體生命週期模式程序 (software life cycle model processes)：生命週期是指整個應用系統開發的程序，由於計畫性質的不同、開發時程的長短、所使用的資料庫系統、開發工具、團隊人力、與系統整合等因素的影響，可能要選擇不同的開發模式。因此在這個階段中，就是要詳細評估應該採取哪一種開發模式，並且確定模式中每一個步驟應該進行的活動。在發展本體時，也應該根據本體的領域、目的、範圍、與應用等，選擇適合的生命週期模式。
- 2、計畫管理程序 (project management processes)：計畫管理是指整個計畫由開始至完成的管理，首先要建立計畫管理的架構、決定每一個過程中的管理機制、並經由監督、控制、回饋、修正等方式，來提升計畫的品質。同樣的，在發展本體時，也應該建立品質管理的機制，並且要嚴謹的監督、控制、回饋與修正等。
- 3、軟體發展導向程序 (software development-oriented processes)：這一階段是真正設計發展的階段，包括設計、發展、安裝、操作、應用、維護與淘汰等過程。
- 4、整合程序(integral processes)：應用系統上線之後，還要進行整合測試，以確定整體的正確性與效能，作為改進的依據，以提升應

用系統的品質。其他如操作訓練、系統轉移、與其他系統的整合、文件撰寫與應用推廣等，這些活動也同樣適合在本體的發展上。

七、Methontology 本體發展方法論

Gomez-Perez, Fernandez, 與 de Vicente (1996)以資料字典 (data dictionary)的觀念出發，提出了本體發展方法論 Methontology，其四個步驟如下：

- 1、掌握領域知識並發展需求規範文件：發展本體的第一個步驟是要掌握住領域的知識，並且建立資料字典，也就是要蒐集領域的重要概念、概念的定義、概念與概念之間的關係、屬性與實例等，再將其記錄在資料字典中。建立資料字典之後，必須進行確認工作，也就是要針對每一個概念，檢查資料字典中各個欄位的內容是否簡潔、正確、有沒有遺漏、實例屬性與類別屬性之間的一致性、並且確定實例與類別都是屬於該概念。
- 2、將所有的中間表徵 (intermediate representation)記錄在不同的表格中，表格名稱與欄位定義如下：
 - (1)、資料字典表：欄位有概念名稱、同義字、縮寫字、描述、實例、類別屬性與實例屬性等。
 - (2)、常數表：欄位有常數名稱、描述、值與量測單位等。
 - (3)、實例屬性表：欄位有實例屬性名稱、描述、型態、量測單位、精度、範圍、預設值、基數與來源等。
 - (4)、類別屬性表：欄位有類別屬性名稱、關係與量測單位等。
 - (5)、實例表：欄位有實例名稱、描述、屬性與值等。

八、SEAL 本體發展方法論

Maedche, Staab, Stojanovic, Studer, 與 Sure (2001)以發展語意入口網站 (SEmantic portAL; SEAL)的經驗，提出本體的發展取向，分成四個階段：

- 1、本體的開始 (kickoff)：包括需求規範與分析輸入來源等。
- 2、本體的加強 (refinement)：包括發展基礎的本體、知識萃取與正規化等。
- 3、本體的評鑑 (evaluation)：包括檢查需求、測試標的應用與分析使用型態等。
- 4、本體的維護 (maintenance)：本體所提供的知識應該盡量符合真實的世界，因此本體需要隨著知識的變遷，加以增強或刪除，而整個發展過程也應該加以文件化。

SEAL 的本體發展取向與其他的本體發展方法論有一個最大的不同之處，就是採用由上而下的方式來決定上下層的概念關係，並且以「有子題」(has subtopics)的分類關係，而非一定要有「是一種」(isa)的繼承關係。因此，SEAL 取向所發展的本體在嚴整性上比較欠缺，如果要合乎 isa 的架構，只要在訂定上下的階層關係時，採用 isa 的架構即可。除此之外，SEAL 的本體發展方法論極為注重本體的評鑑，除了靜態的檢查是否合乎當初所訂定的需求規範之外，還相當重視動態的評鑑，也就是使用者的回饋。

九、KIS/OBM 本體發展方法論

Wang 與 Xu (2000)以建立知識整合系統的經驗，提出本體的建構方法，稱為知識整合系統/本體建立方法，即 KIS/OBM (Knowledge Integrated System/Ontology Building Method)，其步驟如下：

- 1、知識鏈建構：蒐集領域的知識，並加以分析、分類，最後將這些知識加以串聯起來成為知識鏈。
- 2、構成本體：根據知識鏈建立結構化的本體，也就是要辨識領域知識的概念、屬性、關係與值等。發展工作是由系統發展者負責，包括建立領域字典、訂定概念等級、定義常數與定義公式等。
- 3、正規化本體：使用正規化的本體語言工具來模式化本體。
- 4、評鑑本體：評鑑本體的組成與表現，以便進一步改進。

十、知識工程本體發展方法論

在本體的發展上，並沒有一致公認唯一且最佳的方法，不同的人對於同一個領域所發展的本體，可能會有極大的差異，這是因為每個人的思考方向不同所致。由於沒有唯一的方法，也不可能在一的發展中就達到完美，當然也無法獲得唯一的結果。因此，本體的發展是一個持續不斷思考與精煉的過程，在此過程當中，必須要周詳的考慮發展本體的目的，使得本體更能夠接近現實物件的實體與邏輯架構，也唯有在本體接近現實的狀況下，才能發揮其應用的效果。儘管本體的發展沒有唯一的方法論，但是 Noy 與 McGuinness (2001)提出一個知識工程發展方法論 (Knowledge-Engineering Methodology)，係按照圖 2-5 的步驟，以重複的方式，持續的思考精煉與檢查修正，來發展一個較佳的本體，說明如下：

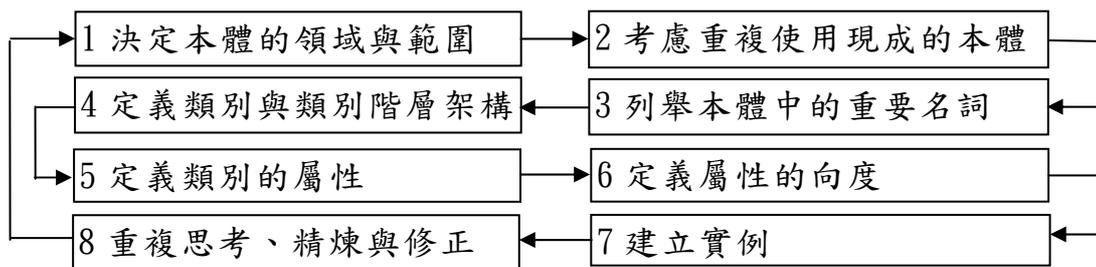


圖 2-5 知識工程發展方法論(資料來源：Noy & McGuinness (2001))

- 1、決定本體的領域及範圍：大部份的本體都會和某一個領域相關，因此發展本體首先要決定領域、未來的應用方向、所應提供的資訊、應該解決的問題、維護與更新的人員與方式等。這些事項都釐清之後，就可以決定本體的涵蓋範圍，也就是說，本體的範圍需要和其應用作對應。如果要有廣泛的應用，那就要有廣泛的本體作基礎。假如應用層面較窄，那就不需要發展一個涵蓋面廣泛的本體。要決定本體的範圍，首先要決定本體的「能力問題」，所謂能力問題是指本體應該回答那些問題，利用這些問題就可以決定應該提供那些資訊，而本體的領域與範圍就可以確定。
- 2、考慮重複使用現成的本體:在決定了本體的領域及範圍後，可以透過文獻或網路搜尋方式，尋求現成的本體，並了解其發展背景、架構、儲存方式、與應用層面等。如果可以符合需求，則應用現成的本體可以省下很多自行發展的時間與花費，並且獲得較佳的效能，即使不能夠符合需求，也可以作為自行建構本體的參考。目前網路上有許多現成的本體可供參考，例如「美國史丹福大學知識系統實驗室」(Knowledge Systems Laboratory, Stanford University)提供了許多現成的本體 (KSL, 2007)、「美國高級國防研究計畫署」(Defense Advanced Research Projects Agency; DARPA)的本體儲存庫(DAML, 2007)，另外也有商業化的本體儲存庫可供

參考，例如「聯合國標準產品與服務碼」(United Nations Standard Products and Services Code; UNSPSC) (UNSPSC, 2007)。

- 3、列舉本體中的重要名詞:本體是由領域中的所有概念組成，概念也稱為類別，類別是一種抽象的名詞，可以代表許多物件的集合。由類別再往下劃分，可能還可以找出抽象的子類別，再繼續往下劃分，最後就會找到具體的物件，稱為實例。因此，本體發展者必須以腦力激盪的方式，盡量列舉該領域中可能的名詞，可以暫時不管這些名詞是否會有重疊現象，也不必理會這些名詞相互之間的關係，其思考方向是由一個名詞開始，以衍伸性的方式來思考有關的名詞。
- 4、定義本體的類別與類別階層架構:類別與類別階層架構是本體的主要成份，可以利用三種不同的方式來加以定義 (Uschold & Gruninger, 1996)：
 - (1)、由上而下定義：先定義最一般化的概念，再往下定義下一層較特殊的概念，依次由一般概念到具體概念的往下定義。
 - (2)、由下而上定義：先由最底層開始，列出所有特殊的名詞，再將某些屬性相同的名詞組合起來，再賦予上一層的概念名稱，依次由具體而抽象的往上定義。
 - (3)、混合式定義：結合由上而下與由下而上兩種方式，首先列出較突出、較明顯的概念，再將其中的某些概念加以組合，以由下而上的方式賦予上一層的概念，再取出這些概念中的某一個，以由上而下的方式，定義較特定的概念。

這三種定義方式，並沒有優劣之分，完全要看發展者對該領域的瞭解程度。如果對該領域有完整的概觀，則比較容易抓到最

上層的概念，採用由上而下的方式應是較佳的選擇；但是如果發展者一時無法先由抽象的概念開始，則可以先列出具體的實例，再由下而上加以整合；混合的方式較適合一般的發展者，因為由上層開始可能會太抽象或太巨觀，由下層開始可能又會太瑣碎或太具體，由中間開始反而較容易描述一個領域，然後可以往下找出具體的概念，往上可以整合出抽象的概念。

- 5、定義類別的屬性 (slot):本體中如果只有類別，則無法提供充足的資訊，也無法回答本體的能力問題。因此，在建立了類別之後，就應該要定義類別擁有哪些屬性、或稱為角色、或 slot。
- 6、定義屬性的向度 (facet)：Slot 之中可能還具有不同的向度，可以用來描述資料的型態、數值的範圍、數值的基數與其他特性等。
- 7、建立實例 (instance)：發展本體的最後一個步驟是要建立類別的實例，才能夠將本體由抽象的概念引伸出具體的物件，達成本體的完整架構。
- 8、重複思考、精煉與修正：儘管經過前述的七個步驟發展了初步的本體，但是並不表示所發展的本體就是完整無瑕，必須要進一步加以評鑑，可以採用重複思考本體的適合性、適用性；或將本體應用在資訊系統之中，以便發現其中的問題；或與領域的專家討論。總之，本體的發展一定是一個重複思考、精煉與修正的過程。

使用知識工程發展方法論來發展本體時，在列出領域的名詞之後，接著是要定義類別、屬性、向度與實例。這四個步驟可以藉助表 2-3 的類別表與實例表來加以記錄，類別表中分別列出超類別、類別、與屬性、型態、基數與向度等；實例表中則列出實例的名稱，以及所有的屬性與向度的值。

表 2-3 本體建構所需類別表與實例表的結構

類別表的結構						實例表的結構	
超類別	類別	屬性	型態	基數	向度	類別	實例名稱

資料來源：本研究

十一、本體發展方法論的彙總

本節探討了八種本體發展的方法論，彙整如表 2-4 所示，期間跨越了 1994 到 2007 年之間。在發展步驟方面，大部份都是原則性的指引，大致可以歸納為由需求、發展與評鑑等三個主要的程序。比較詳細具體的發展方法論有 Gruninger & Fox (1995) 的 TOVE、Gomez-Perez, Fernandez, & de Vicente (1996) 的 Methontology、與 Noy & McGuinness (2001) 的知識工程發展方法論三者。在這三種方法論之中，被稱為企業塑模方法論的 TOVE 本體發展方法論，是為了解決企業問題的應用的角度出發，考量本體必須具有解決企業問題的能力，因此，首先要列舉出所有本體應該回答的問題，再據以發展本體，並且需要有公理與推理機制。Methontology 除了有具體的發展步驟外，還提供了表冊，可以隨時記錄本體的各種中間表徵，不但可以作為設計文件加以保存，也可以提供一致性與完整性檢查的依據。知識工程發展方法論則提供了詳細的發展步驟，由決定領域與範圍、尋求現成的本體、列舉概念、定義類別與架構、定義類別的屬性、定義屬性的向度、建立實例、與評鑑修正等。另外知識工程發展方法論與本體工具 Protégé 相互結合，所使用的本體元素名稱為 class、slot、facet 與 instance。

表 2-4 本體發展方法論的彙總

方法論與提出者	本體發展步驟	特點
IDEF5 Method (Knowledge Based Systems, Inc., 1994)	組織與定義計畫、蒐集資料、分析資料、發展初步本體架構、精練與確認本體	合作式團隊工作
TOVE (Gruninger & Fox, 1995)	情境刺激、定義非正規能力問題、定義術語集、定義正規能力問題、定義公理規範、完全理論	以回答企業的能力問題考量
Skeletal (Uschold & Gruninger, 1996)	確定本體目的與範圍、建立本體、檢視、編碼、整合現存的本體、評鑑、建立文件	綜合性的本體發展原則
IEEE1074-1995 (IEEE,1996)	軟體生命週期、計畫管理、軟體發展、整合	融入軟體發展程序
Methontology (Gomez-Perez, Fernandez, & de Vicente, 1996)	需求規範、記錄中間表徵、正規語言實作、評鑑	資料字典、有完整表冊
SEAL approach (Maedche, Staab, Stojanovic, Studer, & Sure, 2001)	本體的開始、本體的加強、本體的評鑑、本體的維護	語意網導向、注重回饋與評鑑
KIS/OBM (Wang & Xu, 2000)	知識鏈建構、構成本體、正規化本體、評鑑本體	知識整合應用
Knowledge Engineering Methodology(Noy & McGuinness, 2001)	決定本體領域及範圍、考慮重複使用現成本體、列舉本體中的重要名詞、定義本體的類別階層架構、定義類別的屬性、定義屬性的向度、建立實例	發展步驟清楚、考慮本體的重複使用

資料來源：本研究

第六節 本體的表徵

一、本體表徵的層次

Mizoguchi 與 Ikeda (1996)和 Gruber (1993)對於本體的定義有相同的內涵，但是對於本體的表徵則有顯著的差異。前者的本體常用自然語言的描述方式，而 Gruber 則強調使用正規的方法。事實上，這兩種定義並沒有衝突之處，只有層次上的分別而已，也就是說，在本體發展過程中，會先使用自然語言的方式來表徵本體，再以正規法則來塑模本體，最後要融入應用系統中時，會進一步選擇適合的本體表徵方式。因此，本體表徵可以由發展到應用的過程來劃分成三個層次：

- 1、概念層 (conceptual layer)：概念層的本體表徵係使用自然語言、圖形結構、或是樹狀結構的方式，也稱為非正規表徵，這種表徵方式一目了然，可以很清楚的看出概念與概念之間的關係。
- 2、正規層 (formal layer)：正規層的本體表徵係使用正規法則的方式來塑模本體，又可以分成邏輯語言與電腦語言兩種類型。前者如描述邏輯 (Description Logic)、框架邏輯(Frame Logic)等，後者如各種本體電腦語言等。
- 3、應用層 (application layer)：應用層的本體表徵則進一步與資訊系統結合，例如以樹狀結構、圖形結構、或是儲存到資料庫系統之中，以便提供本體為基礎的資訊擷取、語意網頁資訊服務、本體支援的資訊萃取、自動本體學習、或其他基於本體的應用等。

在概念層級的表示方式中，圖形結構表示是一種簡單、清晰與易懂的表示方式，例如圖 2-2 的本體表示，可以利用不同的形狀表示或

以標示方式為之。因此，圖形表示的優點在於很容易看出整個本體的架構與組成。

當本體以自然語言來描述時，也是屬於概念層的表徵，就是利用文字敘述的方式，將概念、關係、屬性、與值的限制等描述清楚。例如「能力可以分成專業能力與一般能力」就是一種本體的描述。如果以 isa 的方式來描述，可以描述成「專業能力 isa 能力、一般能力 isa 能力」。這樣的表示方式雖然清晰易懂，但是當本體的內容很多時，以敘述式的自然語言來描述，就會顯得雜亂，不容易將階層架構表示清楚，也無法以視覺化的方式來瞭解本體的架構。此時，可以改成表格的方式來描述本體，例如利用表 2-3 的類別表與實例表。

二、本體語言的層次

由於應用層面之不同，知識表徵的正規化程度也不相同，因此，在本體語言的發展上，也有不同的層級，以適應不同的應用。Meisel (2005)將本體語言劃分成五個層級，如圖 2-6 所示，以實作層的層次最低，語言學層的層次最高，較高階的層級是奠基於較低階的層級，例如 Ontolingua 的內部表示就是 KIF (Knowledge Interchange Format) (Genesereth et al., 1992)，而 Protégé 也可以使用 RDF 來表示。這五個層次的內涵敘述如下：

1、實作層 (implementational level)：實作層是屬於本體語言最低階的層級，主要的作用在定義資料結構或資源，典型的本體語言為 RDF，例如 RDF 可以利用 URI 來指向資源所在的地方。

語言學層：名詞的分類，例如控制詞彙
概念層：概念的塑模，例如 UML
認識論層：概念描述，例如RDFS、OWL、OCML、FLogic、Protégé
邏輯層：一般邏輯敘述，例如CycL、KIF
實作層：資料結構，例如RDF

圖2-6 本體語言的分層（資料來源：Meisel (2005)）

- 2、邏輯層 (logical level)：邏輯層的本體語言是以一般的邏輯正規化來表徵本體的知識，常用的正規化邏輯為第一階邏輯 (First Order Logic)，代表性的本體語言有 CycL (CYC-CycL, 2002)與 KIF (Genesereth et al., 1992)。
- 3、認識論層 (epistemological level)：認識論層的本體語言使用框架式的表示方式，框架提供類似於物件導向的資料結構，可以表示多種不同的情況，是最常用的一種標準化的本體表示方式。由認知的觀點來看，框架式的知識表示方式是最容易被接受的表示方式，因此也稱為認識論層。一般而言，框架式的本體表徵語言至少能夠表示三個層次的本體內涵，即 Frames、Slots 與 Facets。Frames 用來描述領域概念或類別的集合，並且安排成階層架構，Slots 用來描述 Frames 的屬性，而 Facets 則用來描述 Slots 的限制，例如本體工具 Protégé (Noy, Ferguson, & Musen, 2000)。
- 4、概念層 (conceptual level)：利用圖形結構來表徵本體的概念與概念之外的關係，或是利用概念塑模語言如 UML (unified modeling language)。
- 5、語言學層 (linguistic level)：利用自然語言來描述本體的字彙，或以分類方式來表示本體，例如 Ebay、Yahoo、Google 的網站分類

架構，就是語言學層的本體表示法。

由於本體的應用層面愈來愈廣泛，主要是因為本體所儲存的知識內涵扮演了重要的角色，但是僅有本體還是不夠，還需要有本體的語言來將知識內涵表示出來，以便機器可以閱讀與瞭解。目前已經有許多本體的語言被發展出來，包括本體互換語言(Ontology Exchange Language; XOL)、單純 HTML 本體延伸語言(Simple HTML Ontology Extensions; SHOE)、本體標記語言(Ontology Markup Language; OML)、資源描述架構與綱要 RDF+RDFS (Resource Description Framework)、本體推論層(Ontology Inference Layer; OIL)與美國國防部高級研究計畫署的代理人標記語言與本體推論層 (DARPA Agent Markup Language+Ontology Inference Layer; DAML+OIL)，這些語言都是植基於 XML 的語法。

三、本體語言的需求

本體語言 (ontology language)是一種正規的語言，可以用來建構一個本體，也就是用來表徵某個特定領域的知識，達成某個領域的知識塑模，比較高階的本體語言還具有推理能力，可以處理知識的轉換。本體語言可以分成傳統的本體語言與現代的電腦語言，前者如 DAML+OIL、RDF+RDFS、與 OWL 等；後者如物件導向語言 Java、C++與 PHP 等。

本體要和資訊系統結合，就必須選擇某一種本體語言來進行塑模，因此，本體語言就是介於本體與應用之間的中介者，扮演非常重

要的角色，要能夠發揮本體的應用性，本體語言必須合乎下列需求：

- 1、定義明確的語法：本體語言和一般電腦語言一樣，必須要有非常明確的語法定義，才能讓本體語言工具的發展者寫出語法剖析器，也才可以讓一般使用者寫出合乎語法的陳述。
- 2、定義明確的語意：由於本體語言都需要有表達語意的能力，這種語意的定義也必須要非常明確，不管是人還是機器來解譯語意，都可以得到相同的結果。
- 3、足夠的建構子 (constructor)：建構子是本體語言中用來表示知識的運算子，由於知識的組成包羅萬象，本體語言要有足夠的建構子，才能將本體的知識表示出來。
- 4、有效的推理支援：當本體被使用在知識工程方面時，本體語言就必須具備有效的推理支援，因為推理能力可以檢查本體與知識之間的一致性、類別之間的關係、將實例作分類等。
- 5、充分的表達能力：本體的成份有類別、屬性、關係、實例與值等，其中有各種不同的資料型態、範圍限制等，本體語言必須要有充分的能力來加以表達。
- 6、方便的運算表示：本體語言的一般運算、邏輯運算、或推理運算等，都需要有簡潔與方便的運算表示。

在眾多的本體語言之中，到底應該選用那一種才較適合呢？

Gomez-Perez & Corcho (2002)認為本體所包含的知識有七種不同的成分與層次，因此本體語言處理這七種成分與層次的能力，就變成了選擇本體語言的重要考慮依據：

- 1、處理概念的能力：包括定義概念的能力、劃分概念的能力與註解概念的能力等。

- 2、處理屬性的能力：包括定義概念屬性、實例屬性、區域屬性與整體屬性等。概念屬性屬於概念所獨有，實例屬性屬於實例所獨有，區域屬性是附屬於某個概念，其應用僅限於該概念所涵蓋的範圍；整體屬性則適用於所有的概念。
- 3、處理屬性不同向度的能力：屬性經常有不同的向度，如預設值、資料型態、基數與值的限制等。這些向度可以提供概念與屬性更豐富的資訊，本體語言處理向度的能力也應加以考慮。
- 4、處理分類的能力：本體本身就是一個分類系統的延伸，因此在選擇本體語言時，應該考慮的分類處理能力包括：
 - (1)、是否允許概念包含子概念：就是一般性概念包含特殊性概念。
 - (2)、是否允許概念劃分不完整：本體語言是否可以讓概念與實例儲存在同一層級。
 - (3)、是否允許最終為子概念：本體的組成並不一定要到最底層的實例，本體語言是否允許只儲存到概念。
 - (4)、是否允許多個概念組成的本體：本體語言是否可以同時儲存兩個以上互不相屬的概念。
- 5、處理關係與功能的能力：包括是否允許定義多個關係或功能、是否可以設限參數的資料型態、是否可以設限參數值以進行一致性檢查、是否可以設定操作性定義以推導屬性值。
- 6、處理公理的能力：公理是指不必經過驗證而永遠成立的敘述，可以用來限制資訊、確認正確性、推導新資訊等。如果本體要用在推理方面，就要考慮處理公理的能力。
- 7、處理實例的能力：包括是否可以定義概念的實例、是否可以定義關係的實例。

第七節 本體的評鑑

一、本體評鑑的 5W1H

本體發展出來之後，必須付諸實際的應用，本體可能以隱性的方式直接和應用系統綁在一起，或者是單獨儲存並直接拿來應用，或者是獨立存在但提供應用程式需要時存取。不管如何，本體的發展一定要拿來作為應用的基礎。因此，本體的正確性與內涵必須要經過檢視，何況從發展本體的整個過程來看，必定事先經過評估、研究與需求分析，而本體是否合乎組織的需求、是否合乎使用者的需求、是否可以回答本體的能力問題等，都需要靠本體的評鑑才能知曉。因此，本體評鑑也是本體發展中極為重要的一環。

Gomez-Perez (1995)認為本體的評鑑應該先考慮 5W1H，即：

- 1、為何評鑑 (why)：為了確保本體的正確性、完整性與簡潔性，必須進行評鑑，使得應用系統發展者、終端使用者等，都可以獲得知識的共享與重複使用的好處。
- 2、評鑑什麼 (what)：評鑑的內容包括本體的定義、架構、內涵、文件與軟體等。
- 3、何時評鑑 (when)：評鑑是一個持續且重複的過程，因此整個本體的生命週期中都應該評鑑。
- 4、如何評鑑 (how)：評鑑應該講求方法與技術，因此要訂定標準程序、發展評鑑工具、設定評鑑構面與指標等。
- 5、誰來評鑑 (who)：本體的發展團隊應該負責主要的評鑑，但也可以由其他本體專家、或使用者來評鑑。

6、何處評鑑 (where)：任何地方都可以評鑑，例如本體發展計畫的實驗室、甚至是透過網路等。

二、本體評鑑的項目

在評鑑的項目方面，Gomez-Perez (1995)認為應該包括定義 (definition)的評鑑與文件 (documentation)的評鑑兩者，敘述如下：

1、定義的評鑑：針對本體的定義進行評鑑，其步驟如下：

- (1)、檢查本體的架構：藉以了解本體的架構是否合乎設計規範。
- (2)、檢查定義的語法：檢查語法結構及關鍵字的使用是否正確。
- (3)、檢查定義的內容：包括內容的一致性、完整性、與簡潔性；
一致性是指定義與真實世界的情境一致；完整性是指資訊的內容完整；簡潔性是指所有資訊都是有用的資訊。

2、文件的評鑑：本體的文件對於應用的推廣、本體的整合、應用系統的發展與使用等，都有幫助。因此，在發展本體的過程中，都應該建立完整的文件，並且要加以評鑑。文件的內容應包括本體的發展理念、應用方向、發展過程與參考手冊等。

Gomez-perez (1995)的本體評鑑也只是原則性的規準，如何檢視本體的內容以確定本體的正确性與一致性呢？由於本體是一種分類系統的延伸，類別與類別之間必須合乎 isa 或 partof 的關係，而這兩種關係是本體發展中最不容易達成的要求。因此，Gomez-perez (2003)進一步具體提出了評鑑的方法，列出三種容易犯的錯誤，必須在本體評鑑中特別加以檢視：

1、不一致性 (inconsistency)：不一致性是指本體的組成並未在一致

的基礎上，因此，必須檢查本體是否有循環錯誤、劃分錯誤或語意錯誤。

- 2、不完整性 (incompleteness)：當類別階層架構中間欠缺某些類別、或同一層中欠缺某些類別時，就造成了本體的不完整性。例如「生物←人」就欠缺某些類別，「生物←動物←哺乳動物←人」應該是較好的架構。不完整性意謂著本體可能會遺失許多寶貴的資訊。
- 4、累贅 (redundancy)：累贅是指本體中有些重複定義的類別存在，又有底下四種情況：
 - (1)、類別上的累贅：類別上的累贅是指類別階層架構中，存在有重複的定義，例如「動物←寵物←狗」與「動物←狗」，動物與寵物兩者就有重疊的定義。「狗是一種動物」合乎分類架構，但狗是一種寵物可能就有待商榷，並不是每一種狗都是寵物，因此「寵物」就是一個多餘的定義。
 - (2)、實例上的累贅：實例上的累贅是指類別與實例中，存在有重複的定義，例如「動物←狗←牧羊犬」與「動物←牧羊犬」，「牧羊犬」同時是動物的實例也是狗的實例，因此，「動物←牧羊犬」就是多餘的關係。
 - (3)、有相同的類別定義但名稱不同：例如「電腦語言」與「程式語言」，兩者代表同一個概念，不應該定義成兩個類別，應該訂定一個類別，另一個以同義詞方式表現。
 - (4)、有相同的實例定義但名稱不同：例如「Visual Basic」與「VB」，這也是相同的實例定義，只是名稱不同而已，也應該只訂定一個實例，另外意義相同的實例應以同義詞表示。

OntoWeb (2003)以本體作為知識基礎的觀點認為，本體在付諸實

際應用前，必須針對下列十一個構面進行本體的評鑑，才不致於傳達錯誤的知識：

- 1、一般性規準 (generic criteria)：一般性的規準是指本體的基本需求，又包括兩個項目，一個是塑模能力，也就是所使用的塑模語言或正規化的方法是否可以清楚的表達本體；另一個是支援工具是否可以使本體發揮作用，例如編輯工具、儲存工具、查尋工具、維護工具與整合工具等。
- 2、表現 (performance)：包括本體的大小、所需要的儲存資源、所能提供的查詢型態與本體的內涵等，也就是本體結構所能夠提供的整體表現。
- 3、實用性 (practicalities)：包括應用實績、應用範圍、與外部的聯結與支援等，如果本體可以提供大眾化的應用，其實用性就愈高。
- 4、可用性 (usability)：可用性是指本體所具有的使用能力，包括是否可以合乎特定領域的需求、所能回答的能力問題有那些、重複使用的特性如何、提供查詢的型態與詳細程度等。
- 5、表達性 (expressivity)：表達性是指本體所要表達內容的程度，如果一個本體中的概念愈多、定義也極為清楚、正規化的表示完整，則該本體的表達性就愈高，也就愈能夠彰顯本體所涵蓋的知識。
- 6、正確性 (accuracy)：正確性是指本體概念化的正確程度、關係的正確程度與分類的正確程度等，正確性愈高就表示是好的本體。
- 7、一致性 (consistency)：一致性是指定義或關係是否一致，如果類似的概念中卻有相互衝突的定義，那就缺乏一致性，可能導致類似的輸入，卻產生相反的輸出。
- 8、完整性 (completeness)：完整性是指本體中的概念是否都有定義，是否提供了完整的關係，也就是定義與關係的完整性。

- 9、簡潔性 (conciseness)：簡潔性是指本體中是否儲存有不必要的概念與關係，如果都是必要的概念與關係，則本體的簡潔性就高，多餘的概念或關係會影響本體的效能。
- 10、擴充性 (expandability)：擴充性是指本體的擴充能力，如果本體很容易加入新的概念、定義與關係等，而又不會破壞原有的架構，則本體的擴充性就高。
- 11、敏感性 (sensitivity)：敏感性是指改變本體中的定義或關係對於整體的影響，如果僅作了少許定義或關係的改變，卻造成了許多連鎖變化，就表示敏感性太高，因此，本體的敏感性不能過高。

Mizoguchi (1998)以本體作為知識基礎的角度，提出了下列七個問題，用來評鑑所建立的本體是否可以稱之為本體，如果答案是「是」而且愈是肯定，就表示所建立的知識愈接近本體，也才可以拿來做實際的應用，其問題如下：

- 1、所建立的知識是否為領域成員的共識知識？
- 2、人們是否可以參考其名詞的定義？
- 3、所建立的知識是否可以作為電腦代理人相互溝通的知識？
- 4、所使用的語言是否可以清楚的表示其內涵？
- 5、所建立的知識是否可以重複應用在不同的問題解決情境？
- 6、所建立的知識穩定嗎？
- 7、所建立的知識是否可以用在知識本位系統、資料庫系統的綱要、物件導向系統的設計與發展等？

Uschold 與 Gruninger (1996)則以重複使用本體的觀點，認為要選擇現成的本體來應用，應該以下列三個構面來進行評鑑：

- 1、正規化 (formality)：本體的表徵方式，是否能夠符合應用的需求，是否可以很容易的加以存取。
- 2、目的 (purpose)：原來發展本體的目的與應用方向如何，是否可以配合新的應用。
- 3、主題 (subject matter)：發展本體所屬的領域與其特性如何，是否和設想的應用吻合。

總而言之，本體的發展就如同一般應用系統的發展一樣，在整個發展生命週期之中，都可以也應該要隨時隨地進行評鑑。評鑑的項目以當下的工作重點為主，必須檢視能否達成預設的目標，等到本體發展完成時，再進行一個總體的評鑑，其項目如下：

- 1、本體成熟度：本體成熟度就是本體準備好的程度，成熟度愈高就表示可以推出實際的應用。成熟度的衡量可以朝兩個方面思考，一是本體可以回答的能力問題足夠嗎？通常在設計本體時就會列出所有本體應該回答的問題，只要一檢視這些問題就可以決定本體回答能力問題的成熟度；另一個衡量的成熟度是資源的成熟度，本體要拿來做實際的應用，相關的配套措施也要完備，例如本體儲存庫、網路頻寬與本體存取機制等。
- 2、本體的容錯度：本體的容錯度是以資訊處理的角度，來看本體的內涵容許錯誤的程度，包括不被預期的輸入處理、以及處理概念精確的程度。本體中可能有多層次的概念，而資訊處理時是否可以存取到最接近的概念，都會影響到資訊輸出的可接受度。如果輸出的結果不如預期，可能的原因是概念的劃分層次太過一般化或是太過特殊化、或是屬性與向度的資訊不夠詳實，這些資訊進而可以用來檢視本體的整體架構。

- 3、使用介面的友善度：本體的最終目的是大眾化的應用，因此，由使用的角度出發，使用介面的友善程度會影響到本體的傳達與推廣，外部使用者的評鑑將有助於改進使用介面的友善度。

三、本體評鑑的技巧

當前的資訊系統已經由資料處理進步到概念處理，為了要解譯概念的語意，就必須要有領域的知識作為輔助，而表徵領域知識的最佳資料結構就是本體。但是一個領域的知識可以用不同的本體來加以塑模，此時，選擇最佳或較佳的本體就變得非常重要。通常是透過事先定義好的評鑑規準，來決定本體的品質。Brank、Grobelnik、與 Mladenec (2005)提出四種本體評鑑的取向：

- 1、黃金標準取向 (golden standard approach)：要評鑑的領域本體可以和該領域公認的黃金標準本體作比較。
- 2、應用導向取向 (application-based approach)：評鑑本體在應用系統中的表現。
- 3、資料驅動取向 (data-driven approach)：與領域的大量文件作比較，評鑑本體的涵蓋面。
- 4、人工評鑑取向 (human assessment approach)：由人工判斷本體是否合乎事先所定的設計標準、評鑑標準與需求標準等。

除了前述的四種評鑑取向之外，Brank、Grobelnik、與 Mladenec (2005)更建議評鑑應區分層次：

- 1、語彙、字彙、或資料層次：焦點放在概念、屬性、實例與事實的命名是否合適。

- 2、階層架構或分類層次：焦點放在概念之間的 isa 架構與分類的適當與否。
- 3、其他語意關係層次：除了 isa 關係之外，本體可能還包含其他關係，應該單獨進行評鑑，如果本體是由半自動或自動產生，可能還應該包括回覆率與精確度的評鑑。
- 4、環境或應用層次：一個本體可能是本體集中的一個，可能參考其他本體，也可能被其他本體所參考，或是和應用系統結合。這一個層次的評鑑應針對本體在整個環境中的角色，或本體在資訊系統的應用結果。
- 5、語法層次：當本體以某種語言來表徵時，應檢查是否合乎語法規則、是否形成無限迴路定義。
- 6、結構、架構與設計層次：本體的結構、架構與設計是否合乎事先定義的標準、是否具有更進一步發展的彈性。

Brank、Grobelnik、與 Mladenic (2005)更進一步建議不同層次的評鑑可以採用不同的取向，如表 2-5 所示，每一個層次都可以利用人工來評鑑，另外三種取向只是評鑑項目或參考基礎之不同，但是評鑑的方式是利用人工評鑑還是採用自動化的方式，在文中並未描述。不管如何，如果要按照表 2-5 的方式來作本體評鑑，事實上還有進一步的規準需要被發展出來，也就是必須發展出每一個層次的評鑑指標，以及每一個適用取向的對應指標，才能夠進行評鑑作業。如果要發展成為資訊系統，以自動化的方式進行本體評鑑，更需要定出演算法，才能實作成為系統，以語意網的發展趨勢而言，發展本體評鑑工具以便作自動化的評鑑，應是許多研究者努力的目標。

表 2-5 本體評鑑層次與評鑑取向的關係

層次	黃金標準	應用導向	資料驅動	人工評鑑
語彙、字彙與資料層次	X	X	X	X
階層架構或分類層次	X	X	X	X
其他語意關係層次	X	X	X	X
環境或應用層次		X		X
語法層次	X			X
結構、架構與設計層次				X

資料來源：Brank, Grobelnik, & Mladenic (2005)

由於人工評鑑的耗時費力，因此，發展自動化的本體評鑑工具是許多本體社群的研究重點。Brank、Mladenic、與 Grobelnik (2006) 發展了自動化的本體評鑑系統，是基於黃金標準的本體評鑑取向，可以針對本體的概念與實例作評鑑，也就是針對概念的實例與概念形成的階層架構是否適合作評鑑，和其他以黃金標準取向的本體評鑑方法類似。但是不同之處在於評鑑概念的實例是否適合，也就是要評鑑本體的學習與傳統分群之間的相似度，分群是要將一組實例劃分成許多互斥的群集，再利用不同的技巧來計算自動分群方法與黃金標準分群方法的相似度，相似度越高表示本體的品質越好。

總結而言，本體的評鑑方法和本體的型態與應用有關，過去有許多本體評鑑的方法被提出來，大致可以分成以下數種取向：

- 1、基於一個事先定義的評鑑規準、設計標準、或需求標準等 (Lozano-Tello & Gomez-Perez, 2004) 來做評鑑。

- 2、基於一個黃金標準 (golden standard) 來做評鑑：黃金標準是一個被公認的標準本體 (Maedche & Staab, 2002)。
- 3、基於本體在應用系統中的表現來做評鑑 (Porzel & Malaka, 2004)。
- 4、基於一個領域的語料庫來做評鑑 (Brewster, Alani, Dasmahapatra, & Wilks, 2004)。
- 5、基於不同的層次來做評鑑 (Brank, Mladenic, & Grobelnik, 2006)。

第八節 本體的整合

一、發現本體的方法

許多的本體發展方法都強調盡可能使用現存的本體，或是由現存的本體進行整合或結合 (Uschold & Gruninger, 1996; IEEE, 1996; Noy & McGuinness, 2001; Gomez-Perez, Fernandez, & de Vicente, 1996)。但是如同資訊搜尋的方式一樣，要在全球資訊網中找到並篩選現存的本體並不容易，目前有一些本體儲存庫可以作為搜尋本體的起點，列舉如下：

- 1、Protégé 本體儲存庫：Protégé 本體儲存庫 (Protégé, 2007)是由 Stanford 大學所建置，該本體儲存庫儲存了許多各種主題的本體，其實作格式有三種：框架式 (frame-based ontologies)、OWL 本體 (OWL ontologies)、與其他格式如 DAML+OIL、RDFS 等。每一個本體都伴隨著文字說明，可以瞭解本體的領域與應用層面，目前提供瀏覽與搜尋兩種介面，搜尋成功後並可以自由下載。
- 2、Cyc本體儲存庫：Cyc本體儲存庫是一個綜合性也是最大的常識知識庫 (Cyc, 2007)，最早使用的本體語言為CycL，由於語意網的發

展，Cyc本體的OWL版也在建置之中，整個Cyc本體包含百萬個以上的名詞與斷言，形成人類所使用的常識的集合，因此，需要常識本體者可以考慮Cyc本體儲存庫。

- 3、DAML 本體儲存庫：DAML 本體儲存庫儲存了 282 個本體 (DAML, 2007)，主要的表徵語言為 DAML+OIL、RDFS 與 OWL，並以多種方式呈現。例如 URI、提出日期、關鍵字、開放目錄、屬類、類別、屬性、實例、命名空間等。每一個本體也提供了其他重要的資訊，例如類別、屬性、實例與公理的個數等。
- 4、Onthology 本體儲存庫：2006 年才開始建立的本體儲存庫 (Anthology of ontologies) (Onthology, 2007)，透過一種標準化的機制，稱為本體銓釋字彙 (ontology metadata vocabulary)，來建立本體的描述。除了提供本體所含有的基本資訊外，還提供領域特殊的資訊，可以讓學術界與工業界發展語意網的應用。除了本體的基本資訊外，所有的本體品質還透過使用者評鑑。
- 5、OntoSelect 本體儲存庫：OntoSelect (OntoSelect, 2007)儲存了許多語意網的本體，總共有 1420 個本體資源，按照領域、格式、名稱、語言與標記等來分類，可以提供本體的瀏覽、搜尋與選擇等，也可以用來標記自然語言。
- 6、SchemaWeb本體儲存庫：SchemaWeb (SchemaWeb, 2007) 是一個持續發展中的語意網的本體儲存庫，提供多種分類方式，包括本體名稱、本體描述、命名空間(namespace)、位置、網站位址與聯絡人等。所儲存的文件是符合語義網格式的RDF、OWL與DAML+OIL的本體，提供使用者或軟體代理人查詢，以便即時解決語意網的資料解譯問題。
- 7、Swoogle本體儲存庫：Swoogle本體儲存庫 (Swoogle, 2007)儲存超

過10,000個語意網本體，透過一個搜尋介面，使用者可以輸入關鍵字來查詢本體。

二、本體整合的方法

重複使用現存本體的方式有兩種，一種是本體合併 (ontology merging)，另一種是本體整合 (ontology integration)。前者是將多個相同領域不同主題的本體加以合併，形成一個更大的本體，其操作程序是屬於平形式的，新本體的產生是合併多個本體的結果；後者則是在發展新本體的過程中，利用多個現存的本體為基礎，來建立一個新的本體，其操作程序是屬於階層式的，新本體的產生是發展與整合的結果。在本體的整合程序上 Pinto 和 Martins (2001)提出了下列的步驟：

- 1、確定本體整合的可能性 (identify integration possibility)：本體整合是要在發展新的本體過程中，重複使用現存的本體。因此，首先必須進行可行性研究，針對本體整合所需要的時間、經費、操作性等項目作綜合考量，如果可行才進入本體整合的程序。
- 2、辨識模組 (identify modules)：本體整合可能會使用現存本體的某些模組，或在某些模組中加入新的內容、或對模組內容作更新或刪除，所以需要將現存本體的模組加以分解，模組可能是類別階層架構中的一個分類單位，或是一個主題。
- 3、辨識假設與本體承諾 (identify assumptions and ontological commitments)：瞭解現存本體的假設與本體承諾，有助於瞭解現存本體的內涵與能力問題，也就是要瞭解現存本體可以解決那些問題，對於本體的整合將有所幫助。
- 4、辨識模組中要表示的知識 (identify knowledge to be represented in

each module)：新本體的能力問題和原有的本體未必相同，所以要辨識模組中所需要的知識，以便整合到新的本體中。

- 5、辨識候選本體 (identify candidate ontologies)：透過搜尋私有的本體或公共的本體儲存庫，找到本體的來源，然後根據新本體的需求辨識有關的本體，選擇有關的本體使之成為候選本體，以便進一步分析其適合性。
- 6、取得候選本體 (get candidate ontologies)：為了保有使用本體表徵語言的選擇性，本體整合應該在知識階層整合，不應該在實作階層整合。這樣可以使本體表徵語言的選擇更具有彈性，當現存本體是以某種表徵語言存在時，就必須進行再工程以列出其概念化架構，也就是要回到知識階層的本體表示法。
- 7、分析候選本體 (analyze candidate ontologies)：候選本體的分析是為了決定要參與整合的本體，包括領域專家的技術評鑑與本體工程師的品質評鑑。前者注重候選本體的內容、領域知識的涵蓋範圍，並瞭解需要修改的程度；後者則評鑑候選本體的品質，包括本體的結構、詳細程度與粗細層級等。
- 8、選擇來源本體 (choose source ontologies)：經過前一個步驟的評鑑之後，可以將結果排序，本體工程師可以選擇可用的來源本體，再進一步確認這些來源本體的完整性與相容性，如果達不到要求，可以剔除該候選本體。
- 9、實施整合操作 (apply integration operations)：實施整合操作就是要將現存的本體與欲發展的本體相互結合，可能的結合方式有照原樣整合、修改後整合與補強後整合。整合之後還應視模組化、階層級數、命名方式等標準，決定是否需要再次修改，如果需要則進行修改。

10、分析本體 (analysis of the resulting ontology)：完成初步的標的本體之後，還應針對發展目的、設計標準與評鑑標準等，再次進行評鑑，以確定本體的品質。

第九節 本章小結

本章探討了本體論 Ontology 的理論基礎，Ontology 起源於亞里斯多德的哲學思想，主要在探討宇宙間萬事萬物存在的本質，其結論認為萬事萬物存在的本質可以由十個最具有一般性、也就是最抽象的概念組成。因此，哲學上的本體論是一切知識的本體，因而 Ontology 也被稱為知識本體論。Ontology 中「是什麼」是一種追本溯源的探究，也是一種上下關係的建立，持續的探究根源與建立關係，就可以組成一個領域的知識模型，稱為 an ontology 或 ontologies，也就是本體。

在本體的定義方面，有很多不同的定義被提出來，有的由非常鬆散的角度來定義本體，也有極為嚴謹的定義，大致可以彙整成為下列不同的觀點。將本體視為哲學思想的延伸，是一種哲學上的科目，這種定義混淆了本體論與本體，也就是將 Ontology 與 an ontology 視為是相同的東西；將本體視為一種以自然語言所組成的非正規的概念系統，這種定義過於鬆散，並未提及概念系統組成的要件；將本體視為一種邏輯理論，用來表現概念系統，這個定義將本體視為以邏輯理論為基礎，具有概念推理的功能，是一種比較嚴謹的定義；將本體視為邏輯理論中所用的字彙，這是一個鬆散的定義，只是把本體看成是術語的集合；而最常被引用的本體定義是「本體是一種概念化的顯性規範」(Gruber, 1993)，這個定義強調了「概念化」、「規範」、與「顯性」

三個特性，本體是由概念、關係、屬性、值與實例等共同組成，並以正規方式表徵，使得機器可以閱讀與處理，概念與概念之間的關係必須合乎「是一種」(isa)或「整體部份」(partof)的關係。

在本體的應用方面，已經由早期的人工智慧領域，擴充到其他各個領域，例如溝通上的應用、知識工程上的應用、生物資訊界的應用、軟體工程的應用、知識管理的應用、e-learning 的應用、製造程序上的應用、電子商務的應用等。而最有潛力的應用，應該是未來的語意網的應用。

在本體的分類方式上，文獻中也顯示不同的分類觀點，包括資訊處理觀點、不同的主題觀點、知識重複使用的觀點、概念普遍性的觀點、知識庫知識工程的觀點、本體和應用系統之間依存度的觀點、本體表徵正規化程度的觀點、本體在知識庫系統扮演角色的觀點、本體再使用性的觀點與本體共享性的觀點等。綜合而言，比較常被提到的稱呼是上層本體 (top-level ontology)與領域本體 (domain ontology)，前者是屬於一般性的本體，例如時間、空間、地方等，後者則是屬於特定領域的本體，但不管本體的分類方式如何，本體的最終目的是應用應無疑義。

在本體的發展方法論上，本章也探討了多種不同的方法，包括 IDEF5 方法論、TOVE 方法論、骨架式發展方法論、軟體開發生命週期法、Methontology 方法論、SEAL 方法論、KIS/OBM 方法論、知識工程發展方法論等。各種方法雖然都有其特色，但是本體發展方法還是停留在藝術的層次，迄今未有大家公認的科學發展方法論。綜合言

之，Noy 與 McGuinness (2001)所提出的知識工程發展方法論，是一種廣泛被接受的本體發展方法，其重要步驟如下：列舉本體中的重要名詞、定義類別與類別階層架構、定義類別的屬性、定義屬性的向度、建立實例、重複思考精煉與修正。這些步驟的思考方向是先決定領域知識組成的概念，再決定概念的架構，是一種由廣入深的取向。另外一個方法論是 Methontology (Gomez-Perez, Fernandez, & de Vicente, 1996)，係以資料庫資料字典的觀念出發，其主要特色是提供了各式的表冊可供使用，包括概念分類樹、實例屬性表、類別屬性表、公式表與實例表等。結合這兩種發展方法，也就是採用知識工程發展方法論的步驟，再配合方法論 Methontology 的制式表格，將有助於本體的發展。

在本體的表徵方式上，本章探討了自然語言描述、圖形表示、本體語言等。這些表徵方式以 RDF、RDFS 與 OWL 等較為適合用來表徵一個本體，但其內部還是基於描述邏輯或框架邏輯。在本體的評鑑方面，本章探討本體評鑑的項目、評鑑的技巧、與不同主題的評鑑方法。總而言之，本體評鑑必須要在本體發展與應用的生命週期中，進行評鑑與改進，使得本體能夠保持一致性、簡潔性與完整性。

最後、本章探討了本體的整合與學習，包括現存的本體儲存庫、發現本體的方法、本體整合的方法等。儘管在目前的環境下，本體學習的科技未臻完善，但確實是一個重要的方向，尤其是語意網的成功與否，必須依靠大量的各個領域的本體作為基礎。而本體的建構如果可以透過學習之後，以自動化方式完成，必定能達到事半功倍的效果。