建構動態資料檢集器以檢索蒐集網路資訊*

朱雨其　蔡篤賢　吳元彰　鄭宏昇　楊鍵樵
國立台灣科技大學　電子工程系
台北市基隆路四段四十三號
TEL: (02)2733-3141 ext 7214
<ycchu@sun.epa.gov.tw　ccyang@et.ntust.edu.tw>

摘要

本文倡議一種整合性系統模式來建置動態資料檢集器，用以檢索及蒐集網路環境下的資料。資料檢集器以代理人程式爲骨幹，佐以資料整合、資料存取等組件，以及本體引擎、學習引擎等單元，構成具有可調性及整合性效果的運算模式。資料檢集器根據各種不同的需求，在網路上以動態或週期性地方式蒐集資料，是以可以作爲資料倉儲的前置性工具，以提升資料倉儲的資料品質。我們以環境品質資料的檢集工作爲例，實作一雛型資料檢集器，用以說明相關的概念及方法，並藉以驗證其可行性及正確性。

關鍵詞：資料檢集，代理人程式，本體論，機器學習，層次分析法，資料整合

ABSTRACT

We propose an integrated system model to implement an adaptive data extractor (ADE) for retrieving and gathering information on the Internet. ADE adopts multi-agent as system's infrastructure. In cooperative with the components of integration, access, ontology engine, as well as learning engine, ADE forms a computing model in adaptive and integrated characteristic. Based on diverse requirements, ADE may perform information gathering dynamically and periodically. Moreover, ADE can serve as a front-end component of data warehouse for improving the data quality in data warehouse. We illustrate ADE related concepts and methodologies, and justify ADE's feasibility and correctness by implementing an ADE prototype which can retrieve and gather environmental quality data on the Internet.

Keywords: data retrieval and gathering, agent, ontology, machine learning, analytic hierarchy process, data integration

*本項研究係由國科會 (NSC89-2213-E011-003) 及環保署 (EPA-88-U3L1-03-003) 補助研究計畫經費

1　緒論

晚近由於資訊科技的迅猛進展，使我們可以獲得資訊的能與量都大爲提昇；惟如何有效、正確、且快速地獲取資訊，並進一步有效率地管理與運用，才能真正展現企業組織的力量。傳統的資料檢索 (information retrieval) 系統是指一種固定性的系統，經由一定的輸入介面讓使用者由外部提出查詢，然後再由系統在所連結的資訊源中搜尋，找出系統認爲與查詢相關的文件資料回應給使用者。這類系統在傳統的圖書館資料查詢及近年 WWW 的搜尋引擎方面的應用相當普遍。一般而言，資料檢索的方式可以概分爲*全文檢索*與*分類檢索*兩種；以全文檢索的方式所找到的資料較爲廣泛，不容易有遺珠之憾。以搜尋引擎來說，找回的資料通常會以關鍵字在文件中的出現頻率次數來決定相關性的高低，並加以排序供使用者作參考判斷。但這種簡單的方法並不一定適用於所有的情形，且相對於資料的廣泛性，所得資料集中可能參雜有許多不相干的文件 (同字複義、異詞同義等情況)，正確比率因而降低，執行速度上也會受到影響。以分類檢索的方式所找到的資料則正好相反，資料的正確性提高，但相對的，一些相關的資料也有可能因爲分類的狹隘或分類者的主觀看法而被忽略掉。在檢索過程中需要使用者的不斷介入才能找到正確的分類，也對使用者造成不便。再者，由於使用者對系統所提出的查詢大部分是即時性的，資料檢索系統對於查詢所得資料的再利用方面通常都不會作整體性的考量。隨著網際網路上流傳資料量的日漸龐大，資訊源的變動成爲一種動態的變數，不僅資料隨時都在更新，更有許多異質性的資訊源不斷地增加或被移除，因此傳統上只針對固定資訊源作資料搜尋工作的資料檢索系統，如果不加以改進，便不容易滿足使用者的需求 [2]。

資料蒐集 (information gathering) 系統則可以視爲是一種動態的系統，當使用者對系統提出資料蒐集的需求後，由系統中的使用者代理人程式 (user agent) 負責解釋需求的意

義並據此找出要達成的目標工作，再交付給資源代理人程式 (resource agent)，到外部的資訊源作蒐集工作，工作完成後再帶回系統作整理回應。資料蒐集系統也可以利用資源代理人程式主動去蒐集資料，這種作法不僅擴大了資料蒐集的範圍，更能在到達資訊源開始蒐集資料的，同時便可察覺到資訊源的變動，所蒐集到的資料都能有時效性的保證 [2, 9]。但由於資訊源範圍的擴大，資料量更龐大且在資料的正確性質方面較難要求，所以在資料檢索系統以全文檢索方式時所造成的缺點在資料蒐集系統上也可能發生。目前一些個人性的資料蒐集代理人程式 (personal information agent)[7] 的研究是以使用者檔案 (user profile) 的方式對這方面的問題作改善。代理人程式在出去蒐集資料之前會先讀取使用者檔案，得知使用者的要求領域後再開始蒐集的工作，如此一來便可達到資料檢索系統在使用分類索引時所達到的效果，找到小量而且與使用者需求高度相關的資料。但由於代理人程式都是在讀完使用者檔案後才開始工作，較不具變動性，因此這樣的構想較適用於長期固定性的資料蒐集工作，對於多變性的即時性查詢工作較不可行，也因此使用者檔案的如何有效更新，遂成爲資料蒐集代理人程式亟待克服的一項課題。

本文旨在融合資料檢索技術及資料蒐集過程的優點，倡議一種整合性系統模式，我們稱爲資料檢集系統 (Information Retrieval and Gathering Syatems, IRGS)，根據這種系統模式，我們設計了動態資料檢集器 (Adaptive Data Extractor, ADE)，用以在 WWW 環境中根據使用者需求來檢集資料。經過檢集所得到的資料，可以配合資料倉儲的設計以及資料品質的控管，直接匯入資料倉儲加以儲存，加以再利用；例如某些需要長期性統計的資料利用，或是根據過往查詢結果的經驗，在查詢工作進入後，比對找出相同或類似的例子直接加以利用 (案例庫 (Case Base) 的應用)，對日後處理查詢工作的速度方面得以有效的提昇，甚至可以用這些資料作爲系統學習的比對依據，也有提昇系統效率的效果，另外藉由本體庫的引入，對特定領域查詢所得的資料正確性亦將提高。

本文之組織如下：第二節我們討論資料檢集系統的相關功能及資料檢集器的系統架構與運作流程；第三節則闡述有關資料檢集器的學習機制及其演算法，以驗證系統的可調性 (adaptability)；第四節說明實作的構建，我們並以 "環境影響評估 " 工作所需的環境品質相關資料檢集爲實例說明實作情形。第五節作結論並說明未來的研究方向。

2 動態資料檢集器

我們認爲在開放式環境下的資料檢集系統必須有鎖定，探索及整合資訊的能力，系統在設計上應滿足下列的需求：

- 必須有事前在不同的資訊源中搜尋資訊的能力，並且應避免需要使用者介入修正才能滿足工作需求的情形發生。
- 提供使用者正確的最新相關資訊，這需要由系統週期性的對各個資訊源的改變情況作監控與更新來達成。
- 要能在使用者要求限定的時間內儘力完成工作。

根據上述的設計準則，我們採行代理人程式技術作爲動態資料檢集器 (ADE) 的基礎架構，佐以領域相關的本體庫作爲代理人程式的一種先驗知識。在實際運作時，每一個不同的資訊源都會被分配到一個不需外界干涉即可自己下決定工作的獨立代理人程式；其次，這些代理人程式還具有與其他代理人程式互動，彼此合作幫助滿足使用者需求完成工作的能力。 ADE 主要的工作目標係接收經由轉換器以及本體庫所處理過的查詢需求工作，將所接收到的工作作分析，分解成數個不同的要素，再將這些不同的要素當成工作需求分配給幾個負責不同工作的代理人程式，分別到不同的資訊源中蒐集符合的資訊。例如關聯性資料庫，網頁，或遠端伺服器中可供存取的檔案，待蒐集完成後再傳回到 ADE 內作整合與學習，並依查詢需求的型態作不同的處理之後回應。

2.1 ADE 的組件與其功能

圖 1所示爲 ADE 的內部組件與查詢工作進入後的作業流程梗概。其中**本體引擎** (ontology engine) 分爲依領域相關本體論所構建之本體庫 (ontology library) 及推理驅動器 (reasoning driver) 二個部分；當使用者需求輸入時，本體引擎即依據本體庫中之語彙關係等知識，對需求作初步處理，而後將結果匯給 ADE，是以本體引擎可以視爲是 ADE 的前置性工具。有關應用本體論作爲資料檢索輔助性工具的作法已有相當文獻論述 [2, 4, 6, 8]，惟大部分都將本體論與系統作過於緊密的結合，使得系統的可調性及獨立性降低，我們將本體引擎與 ADE 分別成獨立的組件，是以在實務運用可以更具彈性，使得 ADE 可以成爲領域無關 (domain independant) 的組件。

分派整合管理器 (distribution and integration manager) 負責的工作是將經過本體引擎處理之後的查詢工作，依照其中的關鍵字，將整個查詢工作分解成數個不同方面的子查詢工

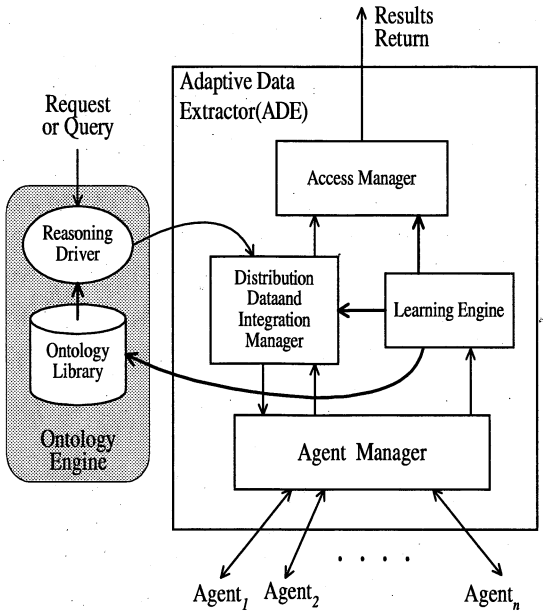

圖 1: ADE 系統架構及組件

作。舉例來說，如果一個有關於環保空氣品質方面的查詢工作進入 ADE，根據空氣品質的本體學，這個查詢工作可能會被分解爲有關於汽機車排氣量，天候，計測區域，計測時間等等不同領域的子查詢工作，而這些子查詢工作也可能會再依需要分解成更細的子查詢工作，最後形成一棵目標樹，再依照分解的結果交給**代理人管理器** (agent manager) 處理。代理人管理器則負責接收分派整合管理器的子查詢工作，將每個工作依需要及可能的限制，配上一個蒐集計畫 (gathering plan)，這裡的蒐集計畫指的是一串需連結的資訊源位址，然後將這樣的一個集合{需求，限制，蒐集計畫}分配連結給負責工作的代理人程式，將所有分配到工作的代理人程式派出去開始蒐集資料，代理人程式的派遣管理亦是由代理人管理器所負責，等所有的代理人程式都將蒐集結果攜回後，管理器會將結果傳回給分派整合管理器以及**學習引擎** (learning engine)。學習引擎負責的是將由代理人管理器所傳回來的結果與之前相同領域中的經驗作比較學習，例如找出是否各個領域或關鍵字之間是否有新的相關性或相似性，如果有新的發現，則將所得到的結果回存到本體庫以及分派整合管理器中作更新的工作，增加下次工作的效率與正確性，有關學習工作的設計將於第三節詳細說明。**存取管理器** (access manager) 則是負責將所得的結果，依查詢工作的性質不同 (即時性或常態性) 分別轉爲合適的格式回覆給使用者端或資料倉儲方面的負責者。

2.2 合作性資料檢集

運用代理人程式以合作方式進行資料檢集的工作可以大略分爲問題解法 (problem solving) 與系統環境 (system environment) 兩部分來作說明 [9]。問題解法部分包含了數種在資料獲取問題中可能有所改變的部分，包括查詢工作的實際內容，被要求負責工作的代理人程式的個數及種類，資料品質的條件限制與要求等等，這一部分的敘述通常可由使用者本身根據問題的內容不同而作相對的設定。相反的，環境部分通常不是使用者所能控制，環境的敘述包括了通訊網路，網路上目前可提供的計算伺服器，資料伺服器，以及資料庫的個數狀況等情形，而且隨時都有變動的可能。

爲了分派整合管理器在進行分散式問題解法及將結果傳給代理人管理器時的運作方便，我們將上述相關的工作敘述轉換爲集合參數的形式。首先是有關環境部分的轉換，設 Ψ 爲一個三元素的集合，表爲 $\Psi = (\tau, \theta, \Upsilon)$，代表環境中某個單位實體對於問題的可利用性，$\tau$ 表示這個實體主要適用於那個領域的查詢工作，可能只是一個關鍵字，也可能是一段較複雜的關鍵字連結敘述，θ 則是介於 0 到 1 之間的一個實數，用來表示這個實體對於此領域的查詢工作能回應的資料品質，最後 Υ 所表示的則是這個實體所能提供的資料量多寡，(可以用百分比或模糊觀念 (Low, Medium, High) 的方式來表現。另外，我們以 $C(\gamma, l)$ 來表示由出發地 γ 到目的地實體 l 所必需花費的成本代價。根據目的地性質的不同，計算方法也可能有所不同，例如在資料庫環境時是以每筆資料的花費爲計算單位，在計算伺服器的環境下則可能是以 CPU 的每一工作秒花費爲計算單位。其次要轉換環境中較高層級的部分，也就是組成整個資料檢集環境的各個網路情況。設 $N = (\Psi, \kappa, \iota, \delta)$ 表示一個網路的環境狀況，其中 Ψ 如上所述，表示整個網路的性質，κ 表示這個網路環境中的計算伺服器集合，ι 是資料伺服器的集合，δ 則是資料庫的集合。

舉例來說，若某一網路環境 N_1 與目前的資料蒐集環境完全無關的話，則 N_1 內的各個元素都應爲空集合，表示爲 $N_1 = (\phi, \phi, \phi, \phi)$，若網路環境 N_2 具有兩個計算伺服器，三個資料伺服器，以及 n 個資料庫，則可將 N_2 表示爲 (Ψ, 2, $\{K_{2.1}, K_{2.2}\}$, $\{I_{2.1}, I_{2.2}, I_{2.3}\}$, $\{D_{2.1}, D_{2.2}, \ldots, D_{2.n}\}$)。

有關問題解法部分，首先在執行查詢工作的轉換。設 $Q = (\phi, \pi)$ 表示一項查詢工作，ϕ 表示此查詢工作的內容，π 則是此查詢工作所需的各種參數集合，ϕ 可以再被分解爲數個子查詢的集合 $(\phi_1, \phi_2, \ldots, \phi_n)$，每一個子查詢工作可分配給一個或數個代理人程式來負

責，參數集合 π 則可包括對於蒐集資訊源的範圍限制，使用者所要求的資訊回饋，以及各項因素如資料品質，蒐集時間，花費多少的比重或數量限制等。其次是對代理人程式的敘述轉換，設 $A_i = (\Delta, \alpha, \phi)$ 表示一個獨立的代理人程式，Δ 表示目前這個代理人程式所看到的問題觀，α 則表示代理人程式在目前環境中所需交換資料的其他代理人程式集合，ϕ 則是目前所負責的查詢工作。

每個代理人程式從初始之後便會隨時有一個本身被給予的問題區域觀，以及部分的問題全域觀，因爲每個代理人程式的全域觀並不完全，所以所蒐集到的資料在最後整合的階段會有不正確，不一致或是過期的可能，因此必須藉由代理人程式之間的通訊交換資料，來使彼此具有更完整的問題全域觀，使最後的結果具有連貫性與一致性。

3 本體庫及學習機制

3.1 本體庫的製作

本體庫的主要作用在於提供系統面對查詢工作時所需的語意轉換服務，以及在面對各個不同的應用領域時提供概念基礎，使系統在面對各式的查詢工作時都能有所認識，進行正確且有效率的表現。本體庫內的元素主要可分爲兩類，分別爲原始詞 (primitive) 與非原始詞 (non-primitive)[6]，原始詞指的是具有單一定義的元素，而非原始詞指的則是可能同時具有多種定義的元素，本文採行原始詞來建立內部資料本體庫，而以非原始詞來建立外部語彙本體庫。

本體庫的建立大致上可分爲兩部分，第一部分是針對不同的應用領域，建立出各自的本體架構；我們引用*觸發對* (Trigger Pair) 的方式來增強本體架構 [3]，以期擴大語意轉換的功能。當關鍵字 S 在某份文件資料中被發現時，如果相關字 T 在一定機率以上將會出現在後文中時，我們稱存 S 爲觸發字，T 則被稱爲被觸發字，(S,T) 爲一觸發對。令 $P(S), P(T)$ 爲所有文件中含有 S 或 T 的關鍵字的文件出現機率，$P(S,T)$ 則爲同時含有 S 與 T 的文件出現機率，則以 MI(mutual information) 來作爲觸發對 (S,T) 的相關性測量，其公式如下：

$$\mathbf{MI(S,T)} = \log \frac{P(S,T)}{P(S)P(T)} \qquad (1)$$

觸發對 (S,T) 與 (T,S) 並沒有對偶性，兩者所產生的相關性並不會相同。由此方法以某個關鍵字對所有文件作過測試後，便可得到一系列的相關字集合，以相關度的大小作排序。

第二部分則是建立起不同本體庫之間的連結關係。舉例來說，假設某個資料檢集的需求“台灣北部地區在 1999 年 1 月的空氣品質資料以及水質資料”，此時如果地理領域的本體與空氣、水質領域的本體間沒有相關的連結，則系統難以了解所要蒐集的資料範圍應該有多大 (例如哪些縣市才算是台灣的北部地區)。如果蒐集資料的來源是網路上的網頁，尙可直接以「台灣北部地區」這類關鍵字來增加資料命中的可能。但若資料源是資料一筆一筆依縣市劃分清楚的資料庫，則代理人程式將無法以這個關鍵字對資料庫下達查詢，因此在各領域間建立連結是很重要的工作。其次，建立連結後，查詢工作不論由那一個領域的本體結點切入，都可以自由的依照需求，找到其他領域的相連結點，增加了系統的活用性，也實際協助代理人程式間通訊工作的實踐。

3.2 學習機制的設計

當代理人程式由外部將資料帶回系統後，將分別傳到分派整合管理器和學習引擎作處理；學習引擎負責對得回的資料作一評估，找出由各個資訊源所得資料對於這一次查詢的權重 (藉由各個資訊源所得的資料是否符合此次查詢需要來評斷)。對於所得的重要性比例可作一經驗性的累積，對以後查詢工作進行時各資訊源的使用比例產生影響。我們採行*層次分析法* (Analytic Hierarchy Process, AHP) 來進行這樣的分析 [10]。運用層次分析法進行決策工作時，大體上可分爲四個步驟進行：(1) 分析系統中各因素之間的關係，建立系統的層次結構。(2) 對同一層次的各元素關於上一層次中某一準則的重要性進行兩兩比較，構造出兩兩比較矩陣。(3) 由判斷矩陣計算被比較元素對於該準則的相對權重。(4) 計算各層元素對系統目標的合成權重，並進行排序。

層次結構的建立，首先應將問題條理化，以構造出一個層次分析的結構模型。接著把問題分解爲稱爲元素的組合部分，這些元素又可按屬性分成若干組，形成不同的層次，同一層的元素會成爲下一層某些元素的準則，具有支配的關係，同時它也受上一層某個元素的支配，稱爲遞階層次結構。

我們參照 [11] 的方式建構兩兩比較的判斷矩陣，假設以上一層的元素 C 爲準則，所支配的下一層元素爲 $U_1, U_2, \cdots, U_n$，欲確定這些下層元素對於準則 C 的相對重要性 (權重) 的話，有下列兩種方法：

1. 如果 $U_1, U_2, \cdots, U_n$ 的重要性可以測量且爲同種單位可作比較之時 (如價錢，重量)，則其重要性可直接以其大小決定。

2. 如果元素之間對於 C 的重要性無法直接定量，則其權重可用兩兩比較法加以確定，如表 1所示，即各元素之間以 1 到 9 的比例標度對重要性幅度給值。

表 1: 權重比例標度的含義

標度	含　義[a]
1	兩個元素具有同樣的重要性
3	兩元素相比，前者比後者稍重要
5	前者比後者較重要
7	前者比後者重要很多
9	前者比後者極重要
2,4,6,8	表示上述相鄰判斷的中間值
倒數	若元素 i 與元素 j 的重要性比為 a_{ij}，則元素 j 對元素 i 的重要性之比為 $a_{ji}=\frac{1}{a_{ij}}$

[a]摘自任善強，周寅亮合著 "數學模型 "

但若僅以這樣的比例標度方法來對元素間的相對重要性來作評判，有時可能會有過於粗糙的缺點產生，且在評判時容易因評判者的不同，得到不盡相同的主觀評判，與真正的客觀環境不一定相符。

在所有元素都作完比較後，所得到的重要性數值可構成一個兩兩比較判斷矩陣 $A=(a_{ij})_{n*n}$，且具有 $a_{ij}>0, a_{ji}=\frac{1}{a_{ij}}, a_{ii}=1$ 的性質。可知具有 n 個元素的判斷矩陣只需得到上 (下) 三角的 $\frac{n(n-1)}{2}$ 個元素即可，因此只需作 $\frac{n(n-1)}{2}$ 個比較判斷。其它有關相對權重、合成權重等 AHP 的詳細運算及操作方式請參閱 [11]。

將 AHP 運用在資料檢集工作時，我們以整體效率作為評定的目標層，第一準則層分別為資料數量，存取時間成本，資料品質三項。底下又可細分到第二準則層，分別為資訊源所擁有的總資料筆數，資訊源內命中資料的筆數 (以上兩項受第一準則層的 "資料數量 "準則支配)，代理人程式蒐集資料完畢後由資訊源帶回資料所需的時間 (受 "存取時間成本 "準則支配)，命中資料比率 (命中資料筆數 / 總資料筆數)，資料正確性 (命中資料內使用者認定正確的資料比率)，以上兩項由 "資料品質 "準則所支配。最下方的方案層部分則有各個不同的資訊源。為了建構各層之間所需的判斷矩陣，需找出各個準則之間的重要性排名；方案層與第二準則層之間只要根據客觀的數據標準便可找出重要性的比率，至於準則層之間以及第一準則層與目標層之間的重要性排名，則需仰賴查詢工作是否有特殊限制，例如，若查詢工作有搜尋時間或資料筆數的限制時，第一準則層的 "存取時間成本 "準則或 "資料數量 "準則的重要性便會提高；或者使用者在初始系統時根據本身需求所作的主觀判斷來決定，再藉由一致性檢驗來判斷是否需要修正重要性排名。在各層之間的判斷矩陣決定之後，接下來便是進行層次分析法的運算，找出方案層各資訊源對於目標層的總權重並先加以記錄。若僅由一次的查詢工作所作出的權重判斷，可能會因為查詢工作的某些特殊限制或是網路的目前流量大小所影響而有失客觀，因此各資訊源之間的權重判斷將以相類似 (相同地區，相同領域) 的查詢工作進行數次後所得的權重結果平均而成。最後判斷出的權重結果被記錄起來，並由此決定系統在之後接受到類似的查詢工作時，在各個資訊源所要分配的代理人程式個數或搜尋資料時間的多少，藉以減少在較差資訊源所花費的系統資源與時間，進而提升系統整體的效率。

4 實作構建與釋例

環境保護近年已成為社會大眾廣泛關心的課題，惟由於環境問題錯綜繁雜，其相關資料的蒐集統合仍存在相當的困難 [13]。我們以 "環境影響評估 "(Environmental Impact Assessment, EIA) 這項業務為例，通常其首要的工作就在於相關環境資料的蒐集。這些資料均散佈在各業務主管機關如環保署、氣象局、內政部、農委會、主計處等。往昔由於網路與資料庫連結技術或是網路頻寬的限制，想要在線上擷取或查詢這些資料，並不容易。近年來由於網際網路的便捷，資料的傳遞與交換已經不構成問題，但是各機關間的資料儲存方式、資料表現格式、語意，及資料的整合等問題卻形成另外一種障礙。

我們可以利用 ADE 來執行環境影響評估所需相關資料的檢集，假設初步的資料需求如表 2。

表 2: 環境資料檢集需求

資料項目	來　源
空氣品質	環保署資料庫及 HTML Table
紫外線量	環保署 HTML Table
水體品質	環保署資料庫及 HTML Table
氣象資料	氣象局 HTML Table
地理圖形	內政部一系列的地圖圖形庫

上述的資料為查詢資料的主要蒐集範圍。另外某些查詢在提出時可能還會附有某些限制條件需要滿足，例如：地理圖形資料的比例尺 (精度) 以及是否包括村里界、道路及水

系等圖層。空氣品質資料及氣象資料則必須含括某個時間範圍內的明細資料以及月平均值及年平均值等彙整性的資料。

4.1 本體庫與本體域的建立

建構資料檢集系統的第一步在於本體庫的建立，因爲在本體庫建立起來後，ADE 才能對於所負責工作的領域有一定程度的認識，用以派出合適的代理人程式，本例針對特定的工作目標執行工作，對前述各個資料項目及處理查詢工作的語彙部分來建立本體庫，佐以本體庫之間的連結關係，成爲整體的“本體域”(domain ontology)。

各個本體庫內所含的內容一開始是藉由觸發對的方式或現有的資料內容制定，惟因「地名」多半具有單一性，爲簡化起見，我們將其屬於原始詞，是以地理資料領域尙勿需利用觸發對的方式找相關語彙。其他內部資料領域的本體庫節點建立，由於結點的內容如站名與地名同樣具有單一性，因此由環保署，氣象局以及內政部所提供的資料來初始內建。至於外部處理查詢工作的語彙部分，則經過觸發對的評估及量測工作後可以設定如下：

1. 空氣品質 ⇒{ 空氣，大氣，氧，塵埃，PSI，懸浮微粒，二氧化硫，二氧化氮，一氧化碳，臭氧 }

2. 紫外線量 ⇒{ 紫外線，紅外線，幅射，射線，電磁波，UV，陽光，臭氧，曝曬，曬傷 }

3. 水體品質 ⇒{ 河，川，水，溪，流，懸浮固體，酸鹼，濁，溶氧量，氨氮 }

4. 氣象資料 ⇒{ 氣象，天氣，氣候，降水，溫度，溼度，風，水氣，氣壓，季節 }

接下來則是建立本體庫之內的上下連結，使本體庫成爲一樹狀結構，在各領域的本體庫建立完成後，再對不同的本體庫作比較，建立起不同本體庫節點之間的聯結關係，這部分的連結工作初始亦由系統建立者來進行，待新的節點在工作後被發現時，才由系統進行更新工作，本範例之本體域初步建立如圖 2。

本體域建立完成之後，將其與實際的分派整合管理器以及代理人程式管理器相互結合，並開始接受查詢工作的進行。查詢工作在進入分派整合管理器之前會先經由語彙本體庫作比對，將分析所得的結果傳到分派整合管理器，讓分派整合管理器知道這個查詢工作所需的領域範圍，然後接下來介由地理本體庫的比對，了解查詢工作所需的地理範圍，找出所有在地理範圍內的地理本體庫節

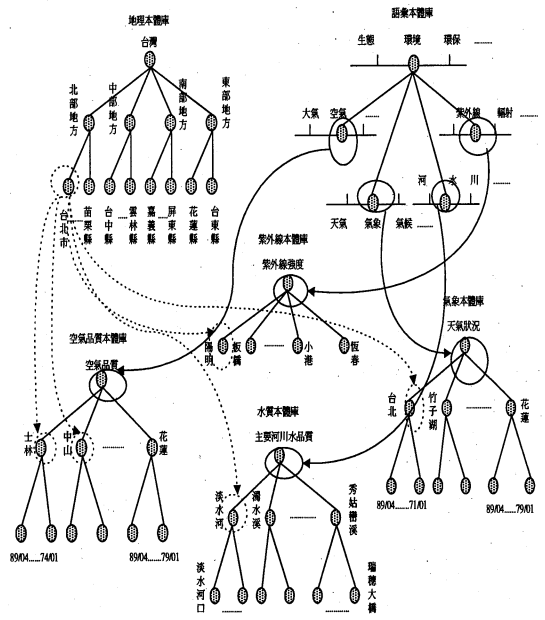

圖 2: 本體域示意圖

點，當作傳遞內容交給所需工作範圍的代理人程式管理器。代理人程式管理器才從所接到的這些地理本體庫節點找到其所聯結到的領域內本體庫節點，並根據節點內容 (監測站站名，地名) 以及一開始查詢內可能附有的限制 (查詢資料的年月，所得資料的需要筆數等)，開始分配工作給代理人程式，由代理人程式開始在不同的資訊源中開始蒐集資料的工作。

4.2 範例與實作測試

例一：檢集「八十五年四月間台北市的空氣品質」資料。
首先語彙本體庫分析出查詢工作領域在空氣品質範圍，接下來由分派整合管理器內的地理本體庫得知地域範圍爲「台北市」，接下來便將台北市這個節點與查詢工作內的限制時間範圍傳給負責空氣品質領域的代理人程式管理器，找出全部與台北市相連的空氣品質本體庫節點，根據節點的內容爲關鍵字，每一節點分派一個代理人程式開始出外蒐集資料，資訊源範圍主要集中在環保署網站的各網頁[1][2]。圖 3 顯示 ADE 執行後所檢集得的資料結果畫面。
例二：檢集「北部地區在八十五年四月間空氣與河川品質」資料。
分派整合整理器在分析查詢工作後，由內部的地理本體庫得知「北部地區」包括有台北市，台北縣，基隆市，宜蘭縣，桃園縣 ...

[1] http://www.epa.gov.tw/psi/psidaily.html
[2] http://www.epa.gov.tw/psi/report

等地區，將這些結點分別傳給空氣與河川的代理人管理器，有關河川領域的蒐集資料範圍主要在環保署網頁群及資料庫[3]，至於地理圖檔則係檢集「環境地理資料庫」之向量圖檔，並結合 ERSI 公司的 Arc/Exploer[1] 協助圖資之檢索工作，圖 4 顯示 ADE 執行本例檢集工作後所得資料的結果畫面，有關詳細之實作程序及程式資料詳見 [12]。

我們針對學習引擎內部的層次分析法及相關的運算以實例說明。有關遞階層次結構的爲置依 3.2 節所述之步驟實施，我們設目標層 "整體效率 "爲 A。第一準則層 "資料數量 "元素爲 B_1，"存取時間成本 "元素爲 B_2，"資料品質 "元素爲 B_3，三者對於目標層的相對權重比例定爲 3:1:5。第二準則層 "總資料筆數 "元素爲 C_1，"命中資料 "元素爲 C_2，兩元素對於 "資料數量 "元素的相對權重比例爲 1:2；"回傳資料時間 "元素爲 C_3，由於只有一個元素歸屬於 "存取時間成本 "元素，因此比重設定爲 1，"命中資料比率 "元素爲 C_4，"資料正確性 "元素爲 C_5，兩者對於 "資料品質 "元素的比重爲 1:4。方案層方面，我們取空氣品質領域的兩個網頁[4] [5] 爲例作比較，分別令其爲 D_1 與 D_2，查詢工作則以例二的「北部地區在八十五年四月間空氣與河川品質」作爲測試標的。一開始先依第二準則層的元素找出對應的數據，進而測出相對權重，首先 D_1 與 D_2 的總資料比數分別爲 57 與 66 筆，兩者相差 9 筆，將近 D_1 的四分之一，因此兩者比重定爲 1:5，接下來在命中資料筆數方面分別爲 26 筆與 32 筆，相差 6 筆，亦將近 D_1 的四分之一，因此比重一樣定爲 1:5。回傳資料時間方面，由於兩者皆爲環保署網頁，在網路寬敞時存取速度並無太大差別，唯 D_2 的資料量稍大，其回傳時間會慢一點點，因此兩者比重定爲 2:1。接下來是命中資料比率方面，D_1 與 D_2 的命中比率爲 $\frac{26}{57}$ 與 $\frac{32}{66}$，約爲 0.45 與 0.48，因此兩者比重定爲 1:3。資料正確性方面，D_2 較 D_1 詳細，因此定爲 1:5。以下是判斷矩陣運算過程時得到的部分結果。

$$\mathbf{Level_3} = \left\{ \begin{array}{lcl} P_1^{(4)} & = & (0.17, 0.83) \\ P_2^{(4)} & = & (0.17, 0.83) \\ P_3^{(4)} & = & (0.67, 0.33) \\ P_4^{(4)} & = & (0.25, 0.75) \\ P_5^{(4)} & = & (0.17, 0.83) \end{array} \right\} \quad (2)$$

$$\mathbf{Level_2} = \left\{ \begin{array}{lcl} P_1^{(3)} & = & (0.34, 0.66, 0, 0, 0) \\ P_2^{(3)} & = & (0, 0, 1, 0, 0) \\ P_3^{(3)} & = & (0, 0, 0, 0.2, 0.8) \end{array} \right\} \quad (3)$$

[3]http://alphapc.epa.gov.tw/cgi-bin
[4]http://www.epa.gov.tw/psi/psidaily.html
[5]http://www.epa.gov.tw/psi/report/pols8904.html

$$\mathbf{Level_1} = \{ \ W^{(2)} \ = \ (0.39, 0.14, 0.47) \ \} \quad (4)$$

所有判斷矩陣所計算得到的 $C.R$ 皆爲 0，符合一致性，最後經由 $P^{(4)}P^{(3)}W^{(2)}$ 可得到 D_1 與 D_2 相對於目標層的權重分別約爲 0.2475 與 0.7525，代表對於這一個查詢工作時這兩個資訊源的重要性比例，明顯可見 D_2 這個網頁的貢獻會比較大，有關詳細的資料及結果詳見 [12]。

上述的分析結果，可以匯入相關的組件模組，甚或作爲調整本體庫的參據。我們認爲以這種機制及運算模式對檢集行爲作學習，不論在系統負載及使用者參與方面均較爲輕省便捷。

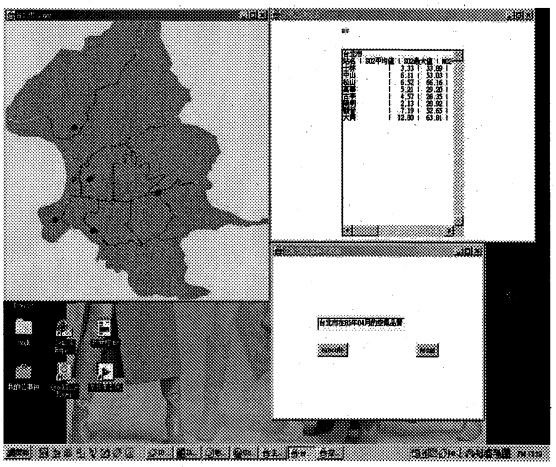

圖 3: 八十五年四月間台北市的空氣品質

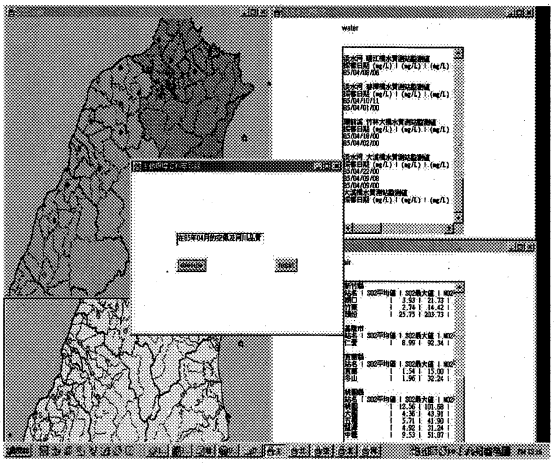

圖 4: 例二資料檢集結果畫面

5 結論

本文針對網路環境下的資料檢索及蒐集等工作，提出一種整合性的系統建置方案。我們的方法調解傳統的資料檢索及資料蒐集方法，以便能夠執行動態及週期性的作業需求，是以資料檢集器可以作爲資料倉儲的前置性工具。本文利用代理人程式及本體庫的配合應用，以平行處理的方式在各個資訊源蒐集所需資料，較之以搜尋引擎進行資料檢索的方式，可以減省判別繁雜分類的時間；而代理人程式的平行處理方式，其效率亦較循序性的搜尋引擎爲佳。其次，若佐以資料品質保證措施 [5]，我們倡議的方法可以加強屬性資料的詮釋性，免除資料誤用及語意的混淆，還能提供診斷性之資訊以期發現錯誤發生的原因及出處。我們對資料檢集器的系統組件、運作功能、學習的策略與機制以及代理人程式的作業行爲等作了相當程度的探討，我們認爲這樣的架構及運作機制，在各種領域相關 (domain-dependant) 的應用上，可以有顯著的功效。本文並以 "環境品質資料檢集" 的應用作爲實踐的範例，並獲致相當的成果。

由於網路環境的資料型態愈來愈多樣化，而且應用的態樣也愈來愈複雜。針對網路上各類數據、文字及圖樣等資訊的整合，以提供智慧型的答詢功能將會日趨普及。我們規劃未來研究方向可以結合 XML 相關標準及工具軟體，發展在網路環境更簡潔及有效的異質性資訊整合作業平台。一般而言，XML 可以作爲一種異質性資料彼此間的共通協定 (common protocol)，這對網路環境下的資料整合有相當的便利性。但是 XML 也存在某些問題，舉例來說，當 XML 資料及系統日趨龐大及複雜時，要如何有效地查詢及管理，而以 XML 所建構的資料與系統，基本上是不具有推理能力的，是以如何結合智慧型代理人程式，用以進行自動化的資料整合或電子商務活動，在學理探討與系統實作二方面都有相當的空間。

參考書目

[1] http://www.esri.com/software/arcexplorer

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval,* Addison Wesley, (1999).

[3] L. Chen and K. Sycara, "WebMate: a personal agent for browsing and searching," *Proc. of 2nd Int'l Conf. on Autonomous Agents and Multi Agent Systems* (AGENTS'98), pp. 132-139, (1998).

[4] R. J. Bayardo Jr. et al., "InfoSleuth: agent-based semantic integration of information in open and dynamic environments," *In Proc. of ACM Int'l Conf. on the Management of Data* (SIGMOD), Tucson, AZ. (1997).

[5] Y. C. Chu, S. S. Yang, and C. C. Yang, "Ensuring data quality in data warehouses through attribute-based metadata and cost evaluation," *In Proc. of 12th Int'l Conf. on Software Engineering and Knowledge Engineering* (SEKE'2000), Chicago, IL. (2000).

[6] D. Fensel, et al., " On2Broker: Semantic-based access to information sources at the WWW," http://www.aifb.uni-karlsruhe.de/www-broker (1998).

[7] V. Lesser et al., "A next generation information gathering agent," *In Proc. of the 4th Int'l Conf. on Information Systems, Analysis, and Synthesis,* Orlando, FL. Also available as UMASS Tech Report 98-72 (1998).

[8] S. Luke, L. Spector, and D. Rager, "Ontology-based knowledge discovery on the World-Wide Web," *In Proc. of the Workshop on Internet-based Information Systems,* (1996).

[9] T. Oates, M. V. Nagendra, and V. Lesser., *Cooperative information gathering: a distributed problem solving approach,* Available as UMASS Tech Report 94-66(ver. 2), (1994).

[10] T. L. Saaty, L. G. Vargas, *The logic of priorities: applications in business, energy, health, and transportation,* Kluwer-Nijhoff, Boston (1982)

[11] 任善強，周寅亮，數學模型，中央圖書出版社，(1998).

[12] 蔡篤賢，主動資料萃取器的應用，國立台灣科技大學碩士論文，(2000).

[13] 楊鍵樵，異質性環境資訊整合與資訊交換技術應用研究，行政院環境保護署科技研究計畫 (EPA-88-U1L1-03-003), (1999)