

國立政治大學

資訊科學系

碩 士 論 文

本體論為基礎的統計資訊整合 - 以政府公開資訊為例

Ontology-Based Statistical Data Integration for Open
Government

研 究 生： 梁世麒

學 號： 98971016

指 導 教 授： 胡毓忠博士

中華民國一百零一年十一月十日

本體論為基礎的統計資訊整合 -以政府公開資訊為例

學生：梁世麒

指導教授：胡毓忠博士

國立政治大學 資訊科學系

摘 要

現代的民主國家無不致力於深化民主的價值，政府運用人民所繳納的稅金進行相關施政，在政府運用國家資源的同時，也應該提供各項施政的統計資料以便說明及用來監督政府施政的成效，政府提供的資料所涵蓋的領域及格式非常多元，若要加以運用產生具有附加價值的資訊，往往單一來源的資料無法滿足需求，必須透過多方的合併參照才能凸顯在資料背後所隱含的價值，因此使用者在運用前必須先針對不同來源的統計資料進行多方的蒐集、參考及比對，最後才能彙整成為有用的資訊，而政府將各種的資料進行公開之後也會快速累積出龐大的資料量，若要透過人工的蒐集比對其困難度也越來越高，因此如何能動態地從不同來源中萃取出有意義的內容便是一個相當大的挑戰，本研究運用語意網技術來解決此一困難，透過單一平台來進行多元資料的彙整查詢，在此平台上使用者可以依其需要選擇特定資料維度或計量單位作為整合條件，並針對特定或不特定的對象進行查詢，最後透過彙整後的結果來提高資料本身的價值，本研究最終目的為提供系統化的方法將政府公開統計資料進行有意義的萃取、彙整及再利用。

Ontology-Based Statistical Data Integration for Open Government

Student: Shih-Chi Liang

Advisor: Dr. Yuh-Jong Hu

Submitted to Department of Computer Science

College of Science

National Chengchi University

ABSTRACT

For enhancement of the value of democracy, the governments are expected to publish statistical data to explain and monitor the performance of policy implementation while they utilize the national resources and the tax for the policies. The data provided by official departments usually contain multiple domain information with diverse formats, which cause the difficulty to generate value-added information from single source. The embedded values could be revealed only by cross-reference of multiple sources. Valued information must be collected, cross-referred, and compared from different sources. In addition, after the government publishes the data, the database would be accelerated to accumulate. The difficulty of manual data collection and comparison would be enhanced consequently. Therefore, it is challenge to extract valued content from different sources dynamically. The study utilized semantic web technology to integrate the inquiry of diverse data with single platform. Users can select specific data dimension or measurement unit based on their requirement as the condition and inquire on specific or unspecific objects. The value of data could be enhanced with the integrated results. The ultimate purpose of this study is to provide a systematized method to extract, integrate and reuse government's public statistical data.

誌 謝

在求學的過程中經歷了結婚以及女兒的誕生，因此第一個要感謝的就是默默支持我的太太，即便在忙到不可開交需要我一起分擔家務的時候，仍能支持我繼續完成學業，再來就是感謝我的母親不斷地鼓勵、提醒我不可鬆懈，同時也感謝胡毓忠教授在研究過程中給了我許多具體的建議讓我能順利完成論文，此外也要感謝實驗室同學們，楊竣展、鄭迪嶸、楊協達、黃雅玲、鄭國平、吳穩男等，不論是論文內容的建議或是各式問題的支援給了我非常大的幫助。

最後，在學習做研究的日子中得到了很多人的幫助才能順利完成，因此滿懷感恩的心祝福大家，謝謝。2013年1月14日



目 錄

中文摘要	1
英文摘要	2
誌謝	3
目錄	4
表目錄	6
圖目錄	7
演算法目錄	9
1 導論	10
1.1 研究動機	10
1.2 研究目的	10
1.3 各章節概述	11
2 研究背景	12
2.1 資訊公開法與政府資訊公開	12
2.2 政府資訊再利用	14
2.2.1 美國	14
2.2.2 英國	15
2.2.3 澳洲	16
2.2.4 台灣	16
3 相關研究	18
3.1 利用語意網與 OLAP 技術整合公部門統計資料	18

3.2	語意式整合查尋引擎	19
4	整合方法塑模	21
4.1	原始資料對應轉換	21
4.2	整合知識庫	23
4.3	查詢語法改寫與執行	28
4.4	查詢結果重整	30
5	整合程序與實驗平台設計	33
6	平台實作與驗證	39
6.1	建立多維度整合平台	39
6.2	測試範例說明	40
6.2.1	範例一:2010年1月亞洲國家與歐洲國家入境台灣的觀光人數	41
6.2.2	範例二:2009年與2010年製造業家數與生產力指數年增率比較	45
6.2.3	範例三:2011年與2012年生產力指數年增率與受雇員工工資比較	48
7	結論與未來展望	51
	參考文獻	52

表 目 錄

6.1	觀光類重要參考指標	41
6.2	來台旅客人數統計	42
6.3	營利事業家數及銷售額	45
6.4	各行業勞動生產力指數年增率	46
6.5	製造業受雇員工與工資	49



圖 目 錄

3.1	利用 OLAP 架構轉換統計數據	18
3.2	語意式整合查尋引擎架構	20
4.1	統計資料描述類型示意圖	21
4.2	資料庫對應至 Data Cube	22
4.3	電子化表格對應至 Data Cube.	23
4.4	整合知識庫描述字彙	24
4.5	水平階層關係鏈	25
4.6	垂直階層關係鏈	25
4.7	子類別關係鏈	26
4.8	計量單位換算關聯性	27
4.9	整合知識庫範例	28
4.10	Data Cube 查詢語法	29
4.11	Data Cube 整合示意圖	29
4.12	查詢語法改寫用關聯性範例	30
4.13	原始查詢語法改寫為子查詢語法	30
4.14	重複性資料移除	31
4.15	計量單位換算	31
4.16	水平維度轉換	32
4.17	垂直維度合併	32

5.1	多維度整合平台架構與整合流程	33
5.2	多維度整合程序	34
6.1	資料源註冊功能	39
6.2	整合查詢功能	40
6.3	範例 1-資料查詢結果	44
6.4	範例 1-查詢結果重整	45
6.5	範例 2-資料查詢結果	47
6.6	範例 2-查詢結果重整	48
6.7	範例 3-資料查詢結果	50
6.8	範例 3-查詢結果重整	50



演算法目錄

1	查詢語法改寫與執执行程序	36
2	查詢結果重整程序	37
3	子集合內資料項彙整運算程序	38



第 1 章 導論

1.1 研究動機

現代化的民主政府，必須具備課責 (*Accountability*)、透明 (*Transparency*) 與公開 (*Openness*) 三大原則，為了致力於建立此一架構，政府必須保障人民擁有法治基礎的資訊取用權。政府在運用國家資源進行施政時，為了各種不同目的進行研究、蒐集及分析所產生的資訊，亦或是將國家資源運用在經濟建設、社會福利或國家安全等不同用途所留下的完整記錄，過去這些資料都掌握在政府的內部進行使用或研究，除非是政府機關自行公佈，一般公民要取得這些內容並不容易，但隨著網際網路的快速興起，大部分的民主國家透過「電子化政府」平台的推動並配合「政府資訊公開法」所建立的法治基礎，政府機關除了提供新的多元服務窗口外，也提供了新的機會將政府擁有資料普及地向大眾公開。

目前政府進行資訊公開的過程中可以發現一些問題，從資料分析的角度來看，雖然開放式政府 (*Open Government*) 的公開資訊提供了廣泛多元的統計資料 (本研究爾後的內容若沒有特別說明則所指的「資料」泛指一般的「統計資料」)，但其提供的資料格式大部分是適合使用者直接進行閱讀，並不容易利用電腦系統自動蒐集、判讀與處理，除此之外，即使部分使用結構化的格式進行發佈，但其使用的類型也是相當分散與多樣，若要同時整合運用這些不同格式的資料也會有相當的困難度，因此若能透過資訊技術的協助，讓這些多元的資料能夠提供自動或半自動的方式進行查詢、蒐集與彙整的功能，最後有助於使用這些內容的大眾快速取得可分析運用的彙整資料，這樣的運用方式除了可讓政府相關施政更加透明外，在資訊透明化的前提下有效提高政府的課責能力及深化民主政治的內涵，讓政府擁有的公部門資料帶來更多的附加價值。

1.2 研究目的

政府資訊公開已經是成熟民主制度中重要的一個環節，各國政府也不斷致力於將公部門資訊提供給其國民使用，在實際的執行面上已可看出許多成效，各國

政府過去所提供的公開資訊服務中所包含的資料範圍相當的廣泛，而其提供的方式不外乎個別獨立的檔案或是將資料儲存於資料庫中提供查詢，目前因語意網技術的迅速發展，開放式政府所提供資料的類型也擴增到具有語意描述的 *RDF (Resource Description Framework)* 本體論格式，雖然開放式政府提供的格式非常廣泛且多元，使用者可以依不同的需求選擇不同類型的格式進行運用，但這樣多元的格式在要進行資料彙整時便會產生一些困難，對進行彙整使用的人而言，首先必須面對眾多不同類型資料，若所需的部分僅存在於單一資料源則該問題尚不嚴重，但往往單一來源所能呈現的內涵較為不足，必須透過多方資料的合併使用才能發掘隱含的數據意義，因此使用者通常需要蒐集彙整不同來源的內容，而這些資料對象可能是使用不同的格式進行存放，這樣的情況更加深了彙整運用的困難度，使用者通常必須先從不同來源的資料集中取出所需要的部分，若是不同來源所使用的格式相異 (如: 資料庫、統計報表等)，則必須先進行格式的轉換，不同格式的資料轉換為同一格式後才能進行彙整與再製，因此實際上必須透過一連串的處理程序後才能產生具有使用意義的彙整資訊，而當進行彙整查詢的條件有改變時則全部的程序必須重新再執行一次。

鑒於以上所面對的困難，本研究建立一個有效的解決方法，透過單一平台的使用，動態彙整不同來源的統計資料。資料整合的程序在符合其原始描述的意義下，使用者可以依據不同的整合需要，選擇及下達適當的條件進行查詢，整合平台再透過分析其查詢條件從不同的來源中萃取出適當的資料集合，最後再將查詢結果彙整重建為符合查詢條件的整合資料，最後整合平台再利用這些重建後所產生的資料以適當的圖表呈現方式提供。透過本研究的整合平台，不同來源且格式相異的統計資料在進行彙整所面對的困難能夠系統化地修正與排除，同時系統化的整合方式能夠為開放式政府的公開資料帶來更多的附加價值與貢獻。

1.3 各章節概述

本文第二章是關於研究背景的描述及資訊公開目前發展的狀況，第三章是與本研究內容有關的其他研究說明，第四章是本研究設計的整合方法塑模，第五章則為利用塑模的方法所設計的平台架構、整合程序及其他細節說明，第六章為實作及驗證說明，最後第七章為總結及未來展望。

第 2 章 研究背景

2.1 資訊公開法與政府資訊公開

在民主原則下國家的主權是由人民擁有，政府既然是基於人民授權而組成，人民便應該是公共事務的最終決定者，因此為了要能夠使人民充分了解政府施政的成果，政府也就必須要提供充分的資訊給人民檢視與利用。政府施政的公開與透明，是國家邁向民主化與現代化的重要指標之一，隨著資訊化與社會變遷，人民對與公共政策的參與度也越來越高，不論是在監督政府的施政或是從事商業及個人的投資行為，所有的決策都必須仰賴多元且正確的資訊來做為輔助判斷的參考，其中政府可以說是國家內外資訊的最大擁有者，因此為了保障人民知的權利、提供資訊公平的運用以及增進人民對政府的了解與信賴，透過建立一套完善的資訊公開制度為現在民主國家必然之趨勢，政府資訊公開制度的建立從歐洲開始發起，挪威、美國、澳大利亞、加拿大、日本等國也陸續針對政府公開資訊進行立法明定，因此台灣也參考其他國家的立法內容，於民國 94 年完成立法程序並公佈「政府資訊公開法」，該法立法的精神便是要為國家施政的公開化與透明化建立出一個明確的制度，透過政府資訊公開法所建立的法源依據，目前已經規範了政府應主動公開的資訊範圍來確保人民知的權利，除此之外，透過其他相關法令的規範，特定項目的資訊在特殊的目的情況下可以不公開或必須延後公開，例如：國家機密保護法、個人資料保護法等，透過不同位階法源的相輔相成讓政府資訊公開的執行更趨完備。

政府公部門所擁有資訊是具有高度利用價值的國家資源，這些公部門資訊所涵蓋的範圍相當廣泛，除了能夠提供過去、現在及未來的相關施政紀錄外，也包含為了社會安全、國家經濟等目的所產生或蒐集、研究的各種資訊，透過將這些資訊進行公開能夠提高政府的課責性及施政透明度，提供人民能隨時觀察並監督政府各方面的施政是否有健全的發展，另一方面這些公開的政府資訊也可以提供用來進行各種不同類型的分析與研究，提高這些資訊的附加價值，因此，將政府公部門所擁有的各類型資料進行公開並提供大眾使用可以總結出三個重要的目的 [2]，第一項是建立且提供高價值的資訊給社會大眾，第二項是一般大眾可以沒有障礙地取得政府相關資訊，同時透過這些公部門資訊使用者可以更了解政府的

運作機制，而第三項則是促進政府相關的運作能更有效率。

但政府在進行公部門資訊公開的程序中也並非直接將所有的政府資料全部公佈，而是必須遵行幾項基本的原則來進行，這些基本的原則如下 [14]：

- 完整性 (*Complete*) - 所有公眾數據都是可以被提供的，公眾數據資料是沒有受到隱私、安全及特權所限制的。
- 初始資料 (*Primary*) - 收集的資料都是最原始的，沒有受到任何形式的整理或修改。
- 有時限 (*Timely*) - 資料的收集盡可能快速地保存其價值。
- 容易取得 (*Accessible*) - 資料沒有障礙地提供使用者廣泛的用途。
- 能電腦處理 (*Process by Machines*) - 資料是具有合理的架構，能夠提供自動化的處理。
- 無對象差別 (*Non-Discriminatory*) - 資料可以提供給任何人使用，不需要經過註冊便可取得。
- 非專有 (*Non-Proprietary*) - 資料的格式是開放的，不受任何獨立實體的控制。
- 版權自由 (*License Free*) - 資料是不受任何版權、專利、商標或商業秘密的規定，但合理的隱私，安全和權限制則是可以被允許的。
- 持續性 (*Permanence*) - 資料必須存放在網路上固定唯一不變的位置，這些位置能提供公眾在分享這些文件時能夠直接連結到原始的資料。
- 易於分析 (*Promote analysis*) - 政府公開的資料應該要能提供適當的格式或方式進行分析使用，而不是提供政府已經分析好的資料。
- 安全格式 (*Safe file formats*) - 政府公開的資料內容不應包含可執行的內容，這種可執行的內容將可能帶來安全性的風險。
- 可信任及追蹤 (*Provenance and trust*) - 公開的資料應該要有數位簽章或是包含建立與發佈日期的驗證，如此能協助公眾確認資料來源的真實性。

2.2 政府資訊再利用

目前開放式政府在實際的執行面而言，美、英等國目前皆已積極在進行政府公開資訊平台的建立，第三方組織透過此平台可以進行資料的再運用，例如運用 *Linked Open Data* 的語意網技術將政府公部門資訊以更開放、透明的方式進行串聯與增值，目前台灣在政府資訊公開法通過後，對於政府資訊公開的相關法令上可說是漸漸趨於完備，雖然在法律面已建立了法源的依據，但目前實際執行資訊公開的方式仍是較為分散，目前台灣國內的重要課題則是如何在法令規範的範圍內透過資訊科技的協助來落實資訊公開的精神，以下則分別針對不同國家目前資訊公開的現狀做說明。

2.2.1 美國

美國政府在 1966 年便公布了資訊自由法 (*Freedom of Information Act*, *FOIA*)，根據此法任何人或任何組織都能夠向美國聯邦政府申請資訊，其聯邦政府在該法的規範內必須盡可能公布相關資訊。除此之外，美國在 1996 年也通過了電子資訊自由法 (*Electronic Freedom of Information Act*, *EFOIA*)，其更奠定了美國在電子資訊公開的基礎法源依據，美國也因此成為世界各國對於資訊公開立法效法的對象，2009 年美國新政府上台後即以透明、開放作為政府的施政方向，對於政府資訊公開的執行更加積極，限定政府機關必須在一定時間內公布具有高價值的資料提供大眾查詢 [12]，此外在公佈資料所使用格式也必須是可以讓電腦程式讀取運用，在同年的 5 月美國政府便公布了 *www.data.gov* 網站，開始在網站上直接公佈經濟、衛生、環境等相關資訊並提供多元的資料格式給使用者在網路下載使用，*Data.gov* 平台成為政府資訊公開的入口平台，該網站除了有提供眾多資料集的完整描述定義外，也提供相關資訊說明如何使用這些資料，*Data.gov* 所提供的資料內容分為原始資料 (*Raw Data*)、工具程式 (*Tool*) 及地圖資訊 (*Geo Data*) 三種，在原始資料的部份有多種不同格式來提供公眾搜尋下載，工具程式則是可以協助使用者使用這些公開資料，地圖資訊則提供與公開資料有關的地圖服務。除此之外，資訊再利用方面則透過 *Linking Open Government Data* (*LOGD*) 計畫將 *Data.gov* 內擁有的資料進行增值再利用，透過將原始的公開資料轉換為 *Resource Description Framework*(*RDF*) 格式，再利用 *Linked Open Data*

(LOD) 架構結合不同來源的資料進行彙整，最後建立出如政府相關資訊儀表板等應用。

2.2.2 英國

2009 年英國政府財政部向議會提出了一份關於聰明政府 (*Smarter Government*) 的報告 [15]，該報告建議政府以線上主動揭露政府資料的方式作為政府資訊公開的模式。同時也提出了七項承諾作為政府的公開資訊原則：

1. 公開資訊以機器可讀取且可重複利用的方式公開。
2. 公開資訊須提供一個單一的入口，該入口能很容易被使用者找到運用。
3. 公開資訊的格式必須是開放的標準且遵循 W3C 制定的規格。
4. 所有的公開的原始資料 (*Raw Data*) 必須是連結資料 (*Linked Data*) 的格式。
5. 大部分的公開資訊必須以公開授權的方式發佈且能自由的使用，即便是商業運用也是如此。
6. 在政府的其他網站公開的資訊也必須以可以重複利用的資料格式作為公開格式。
7. 個人、族群、商業機密或其他第三方的資料皆必須被保護。

利用開放式且可重複利用的資料格式進行資訊公開的最大目的，便是可以提供國民依其自訂的方式使用與分析資料，不論是一般人民或有影響力的團體皆可使用同樣的資料進行分析，除此之外也可將分析後的結果資訊回饋給政府來作為公共政策制定的輔助參考，目前 *Data.gov.uk* 已陸續提供多種不同類別的原始資料，包含了地方政府、衛生健康、教育、刑事司法及警察等資料，其他種類的資訊如民間服務部門、政府採購等也將會陸續公佈，截至 2010 年的 5 月已提供了超過 3000 個資料集，*Data.gov.uk* 目前已有採用 *Linked Data* 的技術直接將公開資訊以 *RDF* 格式進行發佈，並同時也有提供以 *SPARQL Protocol and RDF Query Language (SPARQL)* 技術設計的查詢服務，使用者可以直接透過該查詢服務取得原始公部門資料。

2.2.3 澳洲

澳洲政府於 2009 年成立專責的小組負責整合新的數位技術及提高政府的透明度，在該小組的報告中建議澳洲政府轉換為開放式政府，並提出 12 項建議說明如何達成 [4]，並提出建立一個資訊環境用來主動揭露政府的相關資訊、文件及相關資料集。其中與資訊系統相關的具體建議如下：

1. 將公部門的資訊公開，能夠被使用及再利用。
2. 明確定義版權使用的相關問題。
3. 建立積極的資訊出版計畫。
4. 建立優質的實作指南確保政府入口網站的資訊安全。
5. 在入口網站的內容中能確保個人隱私及其他機密資料的保護。
6. 在各系統中能更新相關聯邦記錄的定義。
7. 要求各整府相關部門建立公開資訊時能符合 *World wide Web Consortium's Web Content Accessibility Guidelines(WCAG)* 的相關使用規範，確保傷殘人士也能使用。
8. 鼓勵資訊相關的應用，如非營利組織提供相關數位工具，促進政府透明度及資訊的運用。

Data.gov.au 目前已經提供原始資料的下載功能，資料格式則以 *XML* 及電子化表格居多，尚未如英、美等包含 *RDF* 格式，但除此之外在線上也有提供額外的工具程式可以進行資料的再加工與運用。

2.2.4 台灣

目前台灣已完成訂定資訊公開法作為政府資訊公開的法源依據，目前台灣政府的資訊公開方式主要包含了幾個功能性的定位 [1]：

- 政令宣導。

- 滿足人民知的權利。
- 促進民主政治的落實。
- 增加國家競爭力。

雖然目前已經建置了如中華民國統計資訊網 (www.stat.gov.tw) 等政府資訊公開網站，但資料內容主要為主計處所蒐集的資料，其他政府部門的資料部分有涵蓋，若要查詢各級政府、部會所公布的大部分公開資訊則仍須經由各級政府單位的網站來查詢，除資料分散沒有統一的蒐尋管道外，資料格式屬於能夠提供電腦可閱讀 (*Machine Readable*) 的資料也較少，相對於美國、英國等政府所建置的平台，除讓公部門資料可以更方便被大眾取得外，利用電腦可讀取 (*Machine Readable*) 的資料格式，第三方組織也能容易地將這些公開資訊進行再加值使用，這些資訊的提供可提高政府應該具有的責任政治及反應能力外，也可以讓公眾使用者更認識政府的施政及創造額外的經濟價值。

第 3 章 相關研究

3.1 利用語意網與 OLAP 技術整合公部門統計資料

在公部門統計資料整合的研究中有運用語意網技術結合 OLAP 架構進行資料整合的方式 [9]，其中 *Online analytical processing(OLAP)* 是目前在進行數據整合分析時經常會使用到的工具，而在 OLAP 系統內描述資料的方式通常會以 *Multidimensional Model(MDM)* 來表達，一般而言 MDM 模型會用星狀的資料模型來表示，單一數值本身可以用星狀的分支來表達出具有多重維度的意涵，這種模型可以容易地將資料以不同維度的檢視角度進行切割 (*Slice*) 與捲起 (*Roll-Up*) 的操作，因此利用 OLAP 系統來進行數據分析有其設計上的優勢，如可以運用類似傳統的 RDB 的管理工具等，而當若要進行分析處理的資料格式並非傳統的 RDB 而是其他格式時，如目前已經被大量運用電子化表格 (*EXCEL*)、*XML-Baesd* 檔案，或是目前越來越多的資料開始以 *Linked Data* 架構的本體論格式在網路上直接進行發佈等，眾多不同格式的資料也可以透過對應的方式將數據資料轉換為 MDM 架構提供運用 OLAP 系統來處理分析。

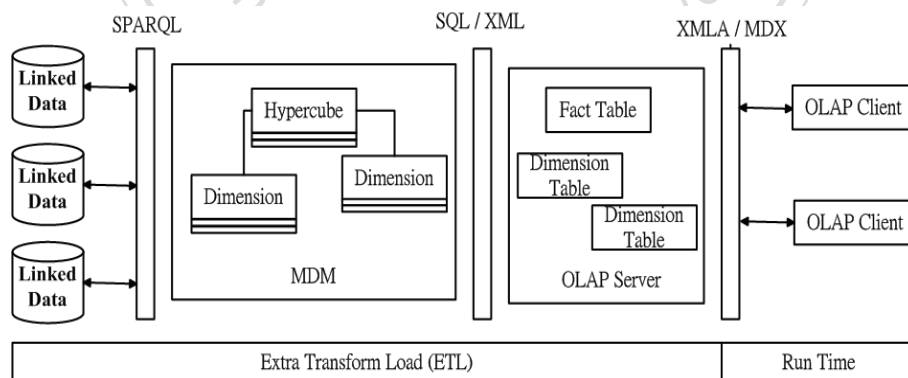


圖 3.1: 利用 OLAP 架構轉換統計數據

當利用語意網技術結合 OLAP 進行資料整合時，待整合的本體論資料對象只要能提供可以執行查詢功能的 *SPARQL Endpoint Service*，這些資料就可以加入 OLAP 可抽取的對象集合中，其中執行資料整合的程序分成 *extract-transform-load(ETL)* 及 *Run Time* 兩個主要階段，*ETL* 階段是整合程序的前置處理作業，

在 *ETL* 程序中必須先建立出可能的 *MDM* 框架，之後再透過利用 *SPARQL* 查詢語法，針對不同的對象抽取對應至 *MDM* 框架的資料，最後建立出以 *MDM* 架構描述的虛擬 *Hyper Cube*，完成 *MDM* 對應程序後再將虛擬的 *Hyper Cube* 導入 *OLAP System* 中產生實體的 *Dimension Table* 及 *Fact Table*，而在 *Run Time* 階段則是透過系統所提供的用戶端工具進行查詢 (圖 3.1)。

雖然透過此一架構可以進行多來源的數據資料整合，但從架構設計上來看還是存有一些的限制，本研究也針對這些限制提出改善的方法如下：

- 整合程序的前置作業必須先行建立 *Multidimensional Model* 後才能進行實際資料的抽取，若資料的內容沒有在 *MDM* 的描述設計內，則該資料便無法被轉換，因此在本研究中則改為利用動態整合查詢的方式來修正這個問題，個別資料源本身仍獨立各自描述，透過分散式查詢的方式從個別來源中抽取出適當資料來完成動態的整合。
- 若有新的資料源加入或修改，則必須重新改寫 *MDM* 的描述並重新執行 *ETL* 的處理流程，這樣的模式在資料源的異動上缺乏適當的彈性，因此本研究是利用資料源註冊的方式提供方便的加入於移除，而整合框架 (*Schema*) 的建立則是在查詢時由使用者利用查詢條件組合出來，這個動態產生整合框架的模式可以增加資料整合的彈性及可能性。

3.2 語意式整合查尋引擎

另一個運用語意網技術整合統計資料的方式是運用動態查詢的方式來完成，語意式查詢引擎 (*Semantic Web Integration Query Engine-SemWIQ*) [10] 的整合方式是運用 *Virtual Graph* 的概念，不同來源的資料源都必須對應至邏輯上 *Graph* 的一部分，而資料整合的對象也可以不須限定在單一格式，資料源只要能夠透過的中介媒體 (*Wrapper*) 輔助提供 *SPARQL Endpoint Service* 的查詢介面就都可以加入整合的範圍，資料源加入的方式是利用動態註冊的方式來達成，所有加入整合查詢的資料描述資訊都會存放在資料源登錄區塊 (*Data Source Registry*) 內，註冊完成後查詢引擎本身會自動利用查詢的方式對資料源內部進行資料的蒐集，運作的方式類似於搜集資料庫統計資訊 (*Database Statistics*) 的方式，針對資料

對象內的所有內容進行查詢掃描，最後將掃描後的資料內容預先儲存 *Meta Data* 區塊中 (圖 3.2)。

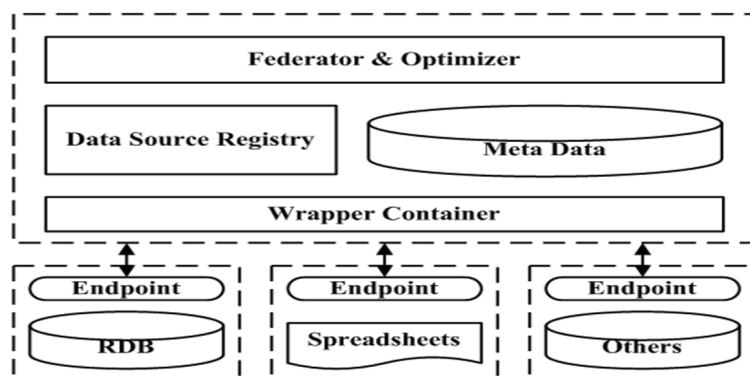


圖 3.2: 語意式整合查尋引擎架構

當要進行整合查詢時，查詢語法的下達是對 *Virtual Graph* 的內容進行查詢，因此 *SemWIQ* 在運作上會先針對查詢語法作分析，比對 *Registry* 及 *Meta Data* 儲存區的資訊，找出對應到查詢語法的實際資料源集合後，針對個別資料源改寫適當的查詢語法進行分散式子查詢，最後將個別子查詢得到的結果合併輸出。

語意式整合查尋引擎 *SemWIQ* 在架構設計上可以進行多資料源的查詢，雖然利用中介媒體的輔助克服了不同資料源描述格式不一致的問題，但資料源仍必須都使用一致的語意描述字彙才能透過查詢完成整合的目的，也就是說資料源本身在語意描述上必須要能事先能對應到 *Virtual Graph* 的一部分，但實際上資料源的描述應該是個別獨立的，在開放式的環境下即便是同一領域的資料源也很難確保使用一致的語意描述，而若是在這個設計上要整合不同領域的資料則更是難以達成，因此在本研究所提出的方法中除了同樣使用中介媒體來克服不同資料源描述格式上的差異，還透過所設計的延伸關聯性知識庫來修正不同資料源間語意描述的不一致，而進行整合時則可以運用這些關聯描述作為整合查詢的條件，透過本研究的設計架構，不同領域且語意描述相異的資料源都可以加入整合的範圍內。

第 4 章 整合方法塑模

4.1 原始資料對應轉換

統計資料的儲存與描述格式非常的多元，如資料庫、電子表格等，因此當要進行整合時，無法避免必須處理不同格式彼此描述資料的差異，因此若可以將不同格式的資料都改用一致的描述架構，則在整合時所遇到的困難便會降低許多，而可以用來描述統計資料結構的語法非常的多元，例如，以 XML 為基礎的 *SDMX* 及以本體論為基礎的 *SCOVO* [7]、*RDF Data Cube Vocabulary* [8] 等 (圖 4.1)。

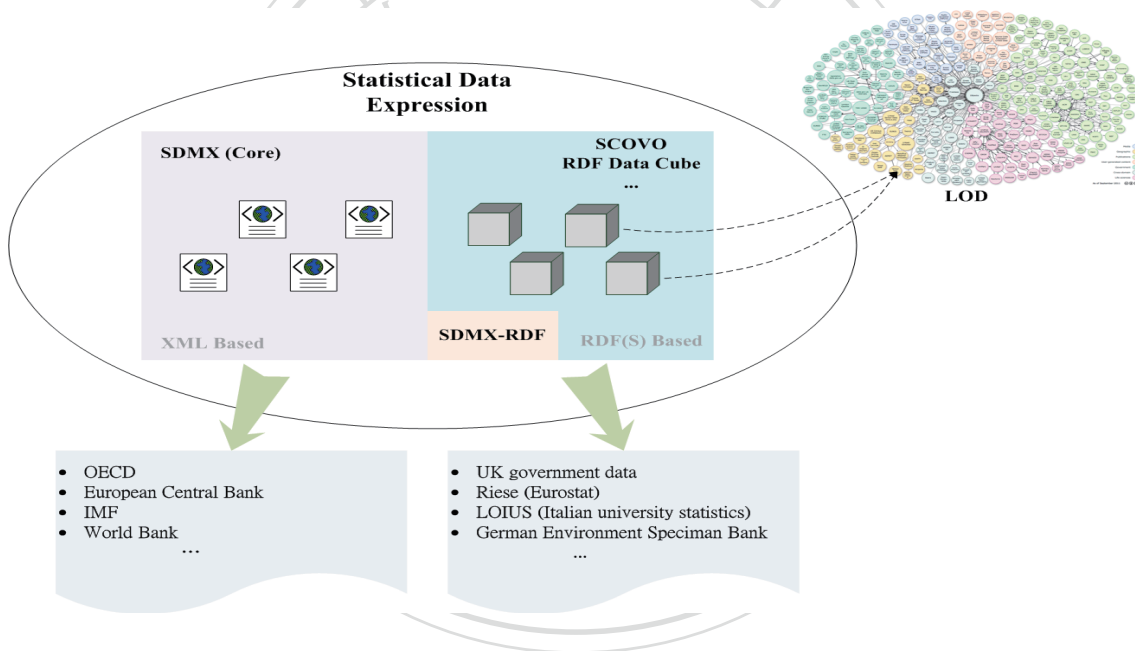


圖 4.1: 統計資料描述類型示意圖

本研究選擇使用本體論作為統一的資料描述格式，用本體論來描述統計資料，比電子表格、資料庫更具有語意上容易擴展的優勢，也有機會可以透過 *Linked Open Data*(LOD) 架構來串連擴充資料的意涵，在本體論的查詢能力上也有完整的查詢語言來支持，如 *SPARQL* [13]，除此之外，目前已經開始有開放式政府 (*Open Government*) 以本體論格式進行統計資料的公開，如 *Data.gov*、*Data.gov.uk* 等。

為了要將不同格式的資料改為利用一致的本體論字彙來進行描述，必須透過框架 (*Schema*)、實例 (*Instances*) 對映的方式轉換為虛擬的本體論，本研究架構

使用 *RDF Data Cube Vocabulary* 作為本體論格式的描述字彙，原始的數值改為利用三個屬性重新定義，*Dimension Property* 描述數值所代表的資料維度，如日期、性別、地區等。*Attribute Property* 則描述數值所使用的計量單位。*Measure Property* 則為統計值本身的數值內容。數值本身所可以被描述的維度並不僅侷限於單一維度，事實上也很少有僅表示單一維度的統計數值，利用 *Data Cube* 的描述方式可以將原本的統計資料轉換為虛擬的多維度 *Data Cube*，其最後整合程序所處裡的對象便是轉換後的本體論結構。而本研究處理不同資料源的整合時，不是直接對個別的資料源作事先的轉換匯出，而是讓原始資料仍存放於其原有的儲存媒體，在進行整合時才會針對可能的對象進行查詢並且萃取出適當的內容進行使用，因此在架構設計上，是在原始的資料源上利用中介軟體 (*Wrapper*) 將資料源進行包裝並提供對外使用，如 *D2RQ* [3]、*XLWrap* [11] 等，進行整合查詢時則是直接對中介軟體下達 *SPARQL* 查詢指令，在接收到查詢指令後轉換為原始資料源內部的查詢語法執行查詢程序，最後抽取出適當的資料轉換為本體論格式輸出。

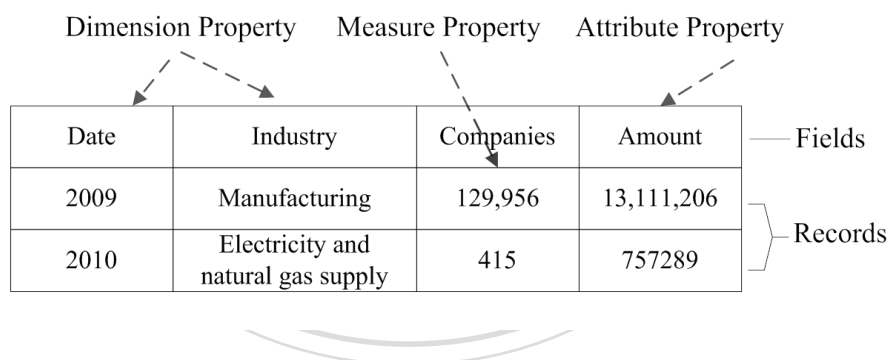


圖 4.2: 資料庫對應至 Data Cube

統計資料若是以資料庫作為為儲存媒體時，因資料庫架構的封閉特性導致原始資料不容易與其他資料源整合，在開放的目的下為了要利用中介軟體來提供本體論格式的查詢，必須先分析資料庫內資料的特徵屬性，區別出適當的 *Data Cube* 屬性 (*Property*) 元素，如圖 4.2 中透過事先的分析，可以區分出 *Date* 及 *Industry* 兩個欄位具有資料維度的特性，而計量單位及資料數值則可以透過 *Company* 及 *Amount* 兩個欄位判斷出來，最後透過中介軟體的功能進行格式對應的設定，完成後便可以在查詢時提供動態的對應轉換，將原始封閉性的紀錄轉換為開放式的多維度本體論，例如透過 *D2RQ* 的使用可以讓資料仍存放於原始

資料庫內，利用 *D2RQ* 內的格式對應檔 (*Mapping File*) 設定，資料庫便可以提供 *SPARQL* 查詢語法的下達及本體論結果的輸出。

除了以資料庫的形式進行儲存與提供查詢使用外，也有為數眾多的資料是以電子表格 (*Spreadsheet*) 作為資料發佈的格式，但是電子表格就如同資料庫一樣具有封閉的特性，同時又因其可以利用文書處理軟體彈性的運用，在資料內容的描述上比起資料庫又更加彈性與複雜，因此這樣結構更不容易與其他資料整合，所以就如同資料庫的對應轉換方式，事先進行資料內容的分析，如圖 4.3 中可以分析出資料維度有 *Date*、*Purpose* 及 *Residence*，而計量單位則是 *Thousand People*，因此在整合架構下電子表格也可以利用如同 *RDF123 [5]*、*XLWrap* 的中介軟體，將資料轉換為多維度 *Data Cube* 描述架構。

Date	Purpose			Residence	
	Business	Tourism	Visit	Japan	USA
	Thousand People	Thousand People	Thousand People	Thousand People	Thousand People
2009	796	2,298	414	1,001	369
2010	938	3,246	497	1,080	396

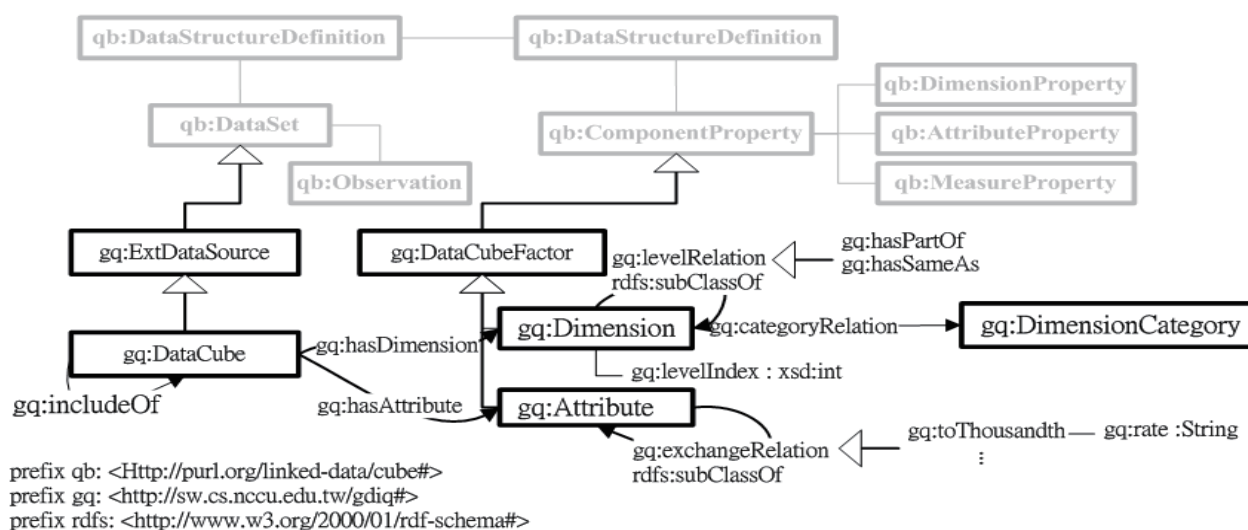
圖 4.3: 電子化表格對應至 *Data Cube*.

雖然資料透過中介軟體的對應程序，最後都轉換為一致的 *Data Cube* 結構，但不同資料源在 *Data Cube* 內特徵元素的使用與描述上仍是自行定義的，不同資料源彼此間仍保持其獨立的特性不需互相參照，資料源可以自行針對個別的資料維度 (*Dimension*)、計量單位 (*Attribute*) 進行各自描述，而 *Cube* 特徵元素彼此間的差異在要進行多資料源的整合查詢時，會利用存放在整合知識庫 (章節 4.2) 內的關聯性描述來作為跨 *Data Cube* 的整合基礎。

4.2 整合知識庫

透過對應轉換的設計，不同來源、格式的資料利用本體論描述字彙重新定義其資料項 (數值)，利用維度 (*Dimension*) 及屬性 (*Attribute*) 等 *Data Cube* 描述字

彙描繪出原本資料數值的意涵，但 *RDF Data Cube Vocabulary* 本身沒有提供處理跨資料來源的描述字彙，因此為了提供跨不同資料來源的整合能力，本研究擴充了原始的描述字彙，設計出整合知識庫 (*Integration Library*) (圖 4.4)，目的是用來儲存與維護不同資料源所擁有的 *Cube* 特徵元素彼此的關聯性定義。



Integration Library

圖 4.4: 整合知識庫描述字彙

在本研究架構下的資料整合方式是利用查詢來完成，對單一 *Data Cube* 內的資料查詢方式是利用本身的 *Cube* 特徵元素作為查詢條件，而當查詢的範圍擴及不同的資料源時，原始的查詢條件很難完整涵蓋這些差異，單一範圍的查詢條件難以完整滿足多重來源的查詢對象，因此進行整合查詢時所面對的第一個困難便是必須修正彼此資料語意不一致的狀況，而我們是利用整合知識庫內的關聯性描述來修正原始查詢語法的不足，這些關聯性描述可以用來擴大可能的查詢對象及找出額外關聯性條件，因此當要進行跨不同來源的資料整合時，整合程序可以利用擴增後的查詢條件從適當的資料對象中切割萃取出符合擴增查詢條件的資料項 (章節 4.3)。

雖然不同格式描述的資料源已經轉換為一致架構並且可以接受查詢語法的執行，利用分散式的查詢方法，可以從不同來源的對象中萃取出適當的資料，但是不同來資料源各自用來描述內容所使用的特徵元素如資料維度、計量單位等字彙是依其需要獨立描述的，因此透過查詢取回的資料項仍是維持原始的資料語意，因此最後還是必須經過適當的重整程序重建出新的 *Hyper Cube* (章節 4.4)。

在整合知識庫的字彙定義中描述多維度 *Data Cube* 的特徵元素主要類別分別為 *gq:ExtDataSource* 及 *gq>DataCubeFactor*，而 *gq>DataCube* 是 *gq:ExtDataSource* 的子類別，用來描述登錄進知識庫內的 *Data Cube* 位址資訊，而 *gq>DataCubeFactor* 是特徵元素的上層類別，在登錄的程序中利用查詢的方式來記錄 *Data Cube* 內所包含的資料維度及計量單位資訊，*Data Cube* 與資料維度的關係鏈則是使用 *gq:hasDimension* 屬性來描述，而資料維度本身則用類別 *gq:Dimension* 描述，另外 *Data Cube* 與計量單位的關係屬性是用 *gq:hasAttribute* 來描述，計量單位本身則是用類別 *gq:Attribute* 描述。

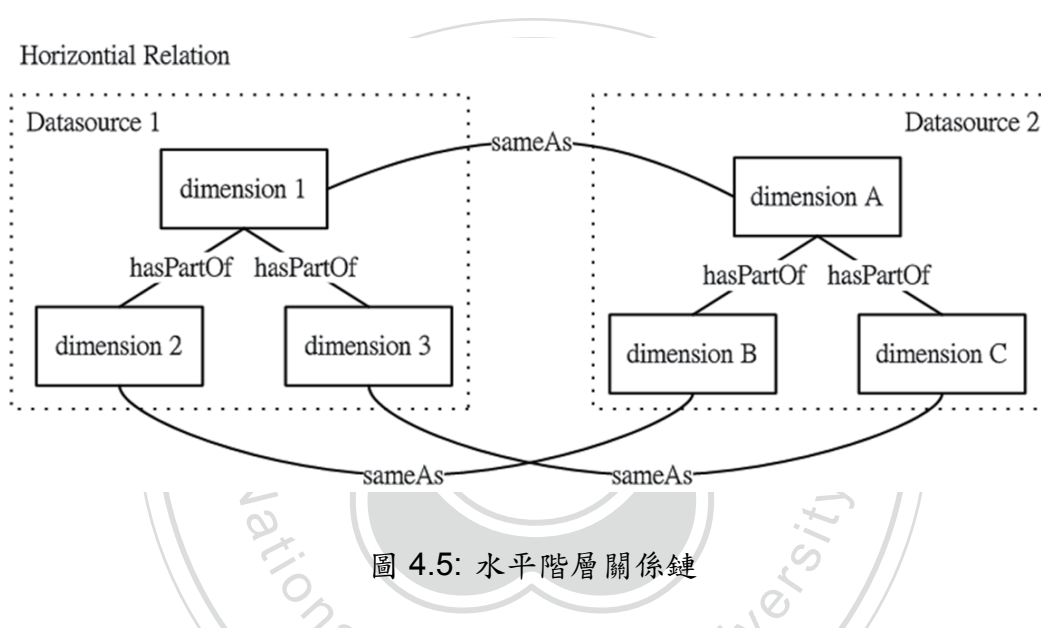


圖 4.5: 水平階層關係鏈

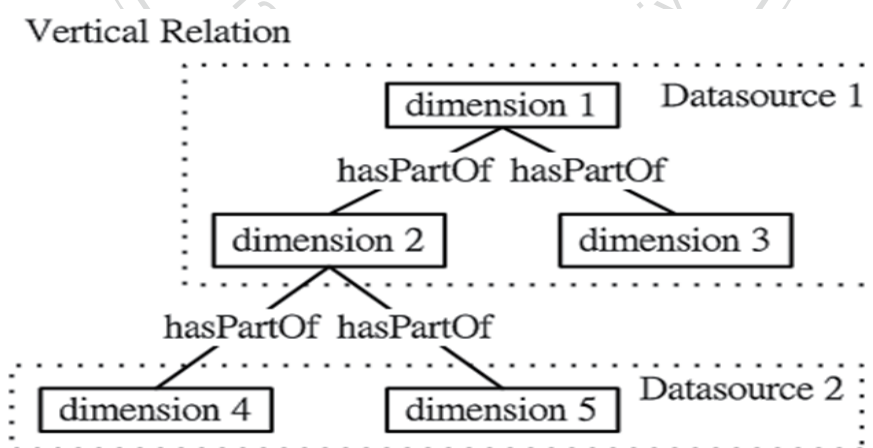


圖 4.6: 垂直階層關係鏈

為了進行跨資料對象整合的需要，整合知識庫內除了描述資料對象與 *Cube* 特徵元素的關聯性外，不同 *Cube* 間的資料維度及計量單位彼此的關聯性也必須完整描述，而這些跨 *Data Cube* 的關連性描述是整合程序中進行查詢語法改寫

與資料重整的基礎資訊，資料維度彼此間可能存在有階層關聯性 (*Level Relation*) 及子類別關聯性 (*SubClass*)，在階層關聯性中可以區分為垂直及水平的關係鏈，其中使用 *gq:hasSameAs* 來描述資料維度間的水平關係鏈，水平關係鏈是指不同資料維度彼此具有相同的意義但在不同的資料對象中使用不同的字彙來描述 (圖 4.5)，另外使用 *gq:hasPartOf* 來描述不同資料維度上下層的垂直階層包含關係 (圖 4.6)，下層資料維度的資料進行適當的數值運算 (如: 加總、平均等) 後即成為上層資料維度的資料，從單一 *Data Cube* 的角度來看，垂直階層的維度關係鏈定義了描述範圍的細緻程度，傳統上透過垂直階層關係可以將單一 *Data Cube* 內的資料進行捲起 (*Roll-Up*) 及切片 (*Slice*) 的操作，而同樣的關聯性描述運用在橫跨不同的 *Data Cube* 時，利用維度階層描述則可以將存放在不同 *Data Cube* 內的同一階層資料切割出來合併成上層資料維度。另外，在子類別的關係鏈的部分利用 *rdfs:subClassOf* 來描述，有別於 *gq:hasPartOf* 的垂直階層包含關係鏈，子類別主要是定義了上層維度類別的細部分類方式，所有子類別維度的集合進行彙總後並不一定能夠取代上層維度，而是要取決於子類別分類的方式是否夠完整，而在整合知識庫中完整的包含關係是使用 *gq:hasPartOf* 來描述而不是使用 *rdfs:subClassOf*，在進行整合查詢時，子類別的關聯定義主要是運用在將不同子類別的內容可以同時在查詢結果中並排顯示，這類型的產出可以做為不同類別的比較參考。

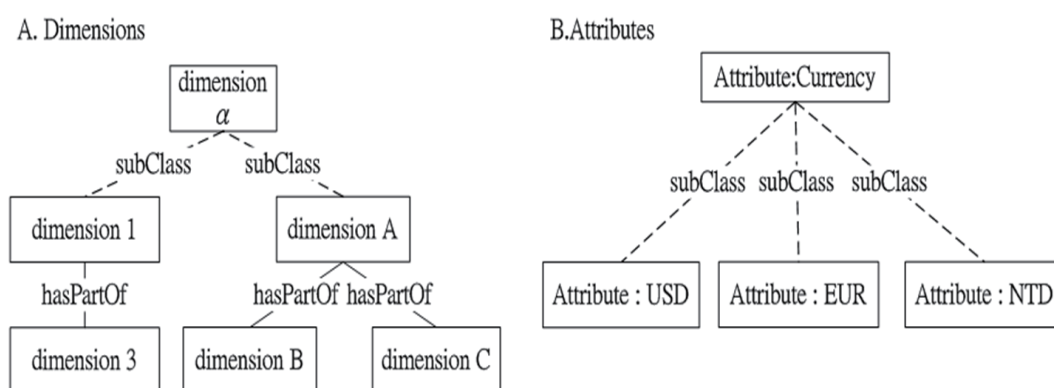


圖 4.7: 子類別關係鏈

除此之外在整合知識庫內額外設計了不屬於描述 *Cube* 元素關係鏈的分類用類別 *gq:DimensionCategory*，主要是用來將登錄進知識庫內的資料維度做適當的分類，並且利用 *gq:levelIndex* 來描述該維度在所屬分類中的階層深度，透過這些額外的資訊在進行整合查詢時，能夠容易地提供分類顯示的功能，便於最後讓

使用者選取適當的資料維度作為整合查詢的條件。除了資料維度外，計量單位 (*Attribute*) 對統計資料也是很重要的特徵元素之一，計量單位在統計資料內定義了數值的表達方式，在整合知識庫內是使用類別 *gq:Attribute* 來描述計量單位，通常單一的 *Data Cube* 內的資料數值會使用一致的計量單位來描述統計數值，但同樣維度的資料在不同 *Cube* 內描述時，則可能會以不同的計量單位來表示，因此跨不同 *Data Cube* 所用來描述數值的計量單位彼此可能存在數值換算的關聯性，因此要整合跨資料源的資料項時，計量單位的換算方式也必須被考量進來，在知識庫內是利用 *gq:exchangeRelation* 來定義計量單位的換算關係鏈，在其下則是依不同計量單位所需要的換算關聯性定義出實際運用的子類別，如千分之一的換算關聯性描述 *gq:toThousandth*(圖 4.8)，在個別換算關聯性中實際的換算比率則使用 *gq:rate* 來定義。

C. Exchange Relation



圖 4.8: 計量單位換算關聯性

從整合知識庫範例 (圖 4.9) 中可以觀察出，其中包含了兩個資料源分別為 *DataCube_1* 及 *DataCube_2*，在 *DataCube_1* 內定義兩個資料維度分別為產業別 (*Manufacturing*) 及年度 (2010)，而使用的計量單位為百分比 (*Percentage*)，*DataCube_2* 的資料維度則定義了產業別、日期 (2010-01、2010-02 等)，計量單位則為新台幣千元 (*Thousand-of-NTD*)，因此可以將個別獨立的資料維度「年度」及「日期」透過關係鏈 *gq:hasPartOf* 串聯起來，而單位「千元」則可以透過 *gq:toThousandth*(千分之一) 建立與「百萬元」的換算關係，利用這些關係描述將有機會可以將原始的資料轉換為另一個面向，如 *DataCube_2* 可以將原本以「日期」表示的資料轉換為以「年度」表示，而數值單位也有機會轉換為以百萬元為計量單位。

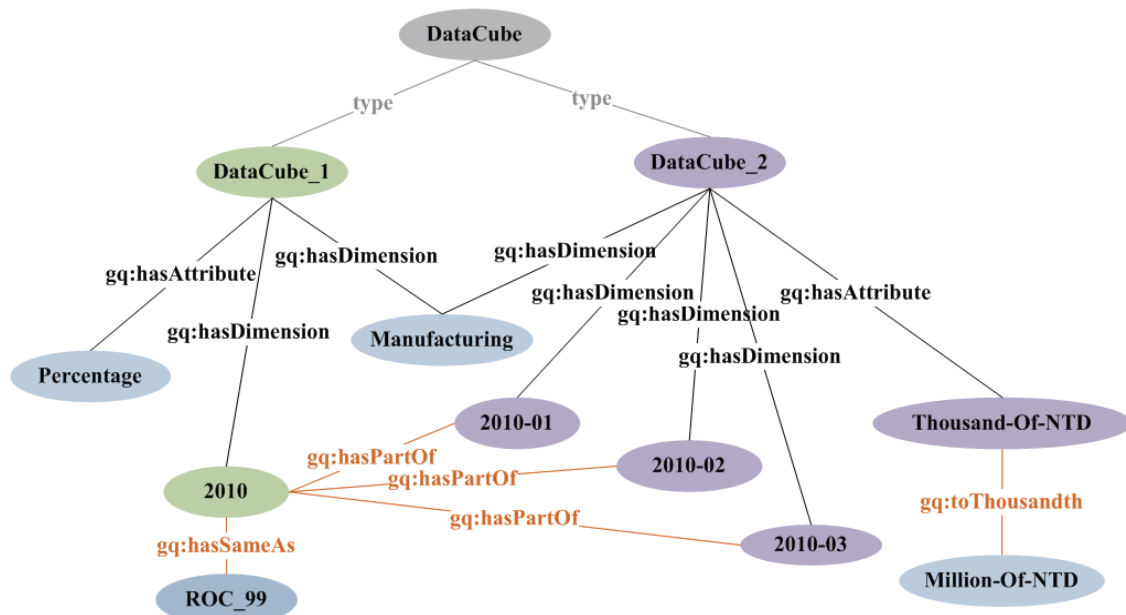
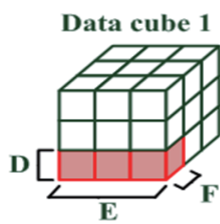


圖 4.9: 整合知識庫範例

4.3 查詢語法改寫與執行

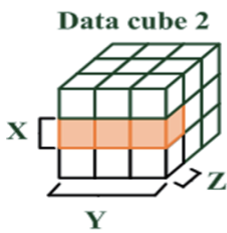
傳統上不同資料來源進行資料整合時會先執行資料框架 (Schema) 的整合，將資料源內的資料進行對映 (Mapping) 與校準 (Alignment) 的程序後才會進行資料實例 (Instances) 的整合，但這種整合方式運用在統計資料上不一定適合，因不同資料源間彼此描述的領域範圍的差異性可能很大，不一定能夠事先彙整為單一的資料源，另外統計資料運用的方式常是將不同類型的資料依照特定的查詢目進行個別的萃取，最後將萃取出來的結果進行比對、參考及再利用，因此有別於傳統的整合方式，在本研究所設計的方式是透過分散式查詢個別已註冊的不同 *Data Cube* 對象中萃取出適當的資料進行整合，其中使用的分散式查詢方式是利用本體論查詢語言 SPARQL 針對不同的對象下達適當的語法來完成 (圖 4.11)。

在多維度架構下，進行整合的方式是依據不同的查詢需求，從已經註冊的資料源集合中，選擇出特定的 *Cube* 並從中切割出適當的資料項進行整合，而在進行整合查詢時，查詢條件在資料對象很多的情況下是難以含蓋到所有可能的資料對象，而若是整合的條件是上層的彙總維度則條件的選擇更是困難，例如圖 4.11 的 *Data Cube 1* 萃取出資料所用的查詢條件是 $\{D,E,F\}$ ，而要從 *Data Cube 2* 萃取出資料所用的查詢條件是 $\{X,Y,Z\}$ ，因此不同的資料對象有不同的描述



Query Syntax

```
SELECT ?item ?value
WHERE { ?item qb:measure ?value.
        ?item qb:dimension D .
        ?item qb:dimension E .
        ?item qb:dimension F .
}
```



Query Syntax

```
SELECT ?item ?value
WHERE { ?item qb:measure ?value.
        ?item qb:dimension X .
        ?item qb:dimension Y .
        ?item qb:dimension Z .
}
```

圖 4.10: Data Cube 查詢語法

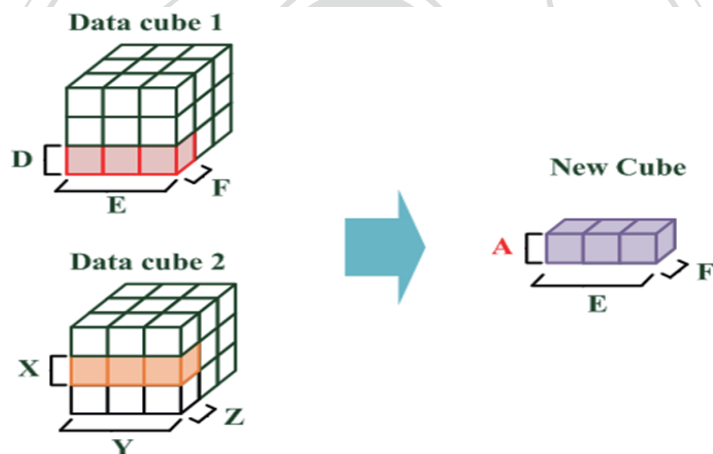


圖 4.11: Data Cube 整合示意圖

語意，當要進行資料整合時，要透過單一查詢語法的下達來完成整合查詢是有其困難，再者若透過整合知識庫內的關聯性描述(圖 4.12)可以看出維度 D 與維度 X 可以彙整成維度 A ，但若是改以維度 A 為條件直接進行個別查詢則會造成無法取得資料，原因是個別資料源實際上不存在有維度 A 的描述，因此進行整合式查詢時，為了要修正個別資料源語意上的差異，在整合平台的設計上是透過整合知識庫找出原始條件的延伸關聯性條件，透過這些延伸的條件擴大可能的查詢範圍，在找出所有可能的查詢條件後再針對各別不同的查詢對象改寫出適合各別對象的查詢語法進行查詢。

在圖 4.13中，原始查詢語法 (*Original Syntax*) 透過知識庫內的關聯性描述來擴充相關條件產生改寫後的查詢語法 (*New Syntax*)，但擴充後的查詢語法不一定

Integration Library Description Example

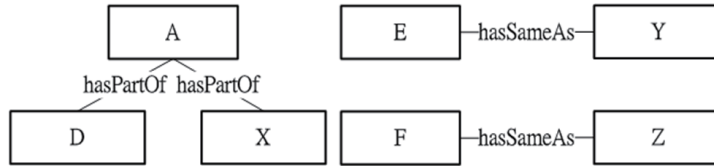


圖 4.12: 查詢語法改寫用關聯性範例

適合個別的查詢對象，原因是查詢語法的條件整合了各種不同的可能性，但資料對象進行查詢時只需要使用到部分的條件即可，而在實際進行查詢之前我們可以透過知識庫內的資料源註冊資訊得到資料源個別擁有的 *Cube* 元素，因此可已針對不同的對象篩選出適合的條件進行分散式查詢 (圖 4.13內的 *For Data Cube 1* 及 *For Data Cube 2*)

Query Syntax Rewriting & Execute

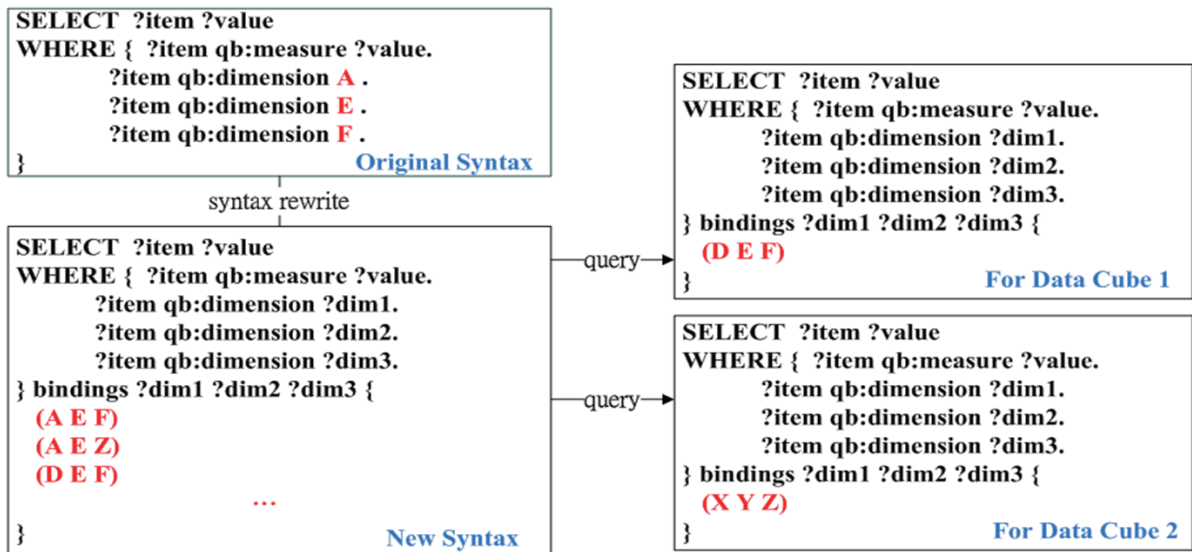


圖 4.13: 原始查詢語法改寫為子查詢語法

4.4 查詢結果重整

在查詢語法改寫的過程中，透過分析原始查詢條件並利用整合知識庫內的關聯性描述，從特定的資料對象中切割萃取出符合條件的資料，但這些透過擴充條

件萃取出來的資料項仍保留其原始的語意結構各自獨立，與原始查詢目的期望得到的整合性資料還有一段差距，因此必須透過適當的重整程序修正結果集中語意描述的差異，最後重建成為完整的 *Hyper Cube*，因此實際在進行結果重建的過程中就必須處理幾項重要的問題：

- 不同的資料源內可能存在相同的資料，進行彙整程序後即會產生重複性資料集合，而在彙整結果重建的過程中也可能產生新的重複性資料，因此在重建程序中必須將這些具有相同的資料維度、計量單位及統計數值等特徵元素完全相同的重複性資料進行移除 (圖 4.14)。

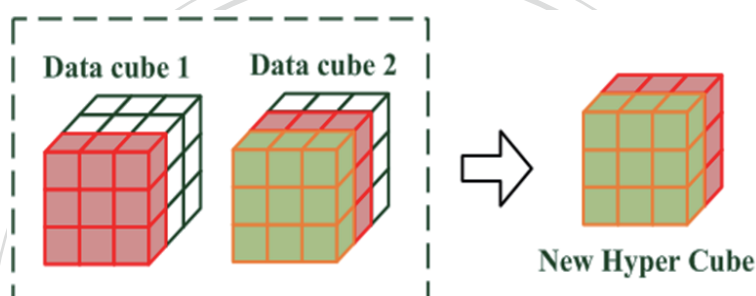


圖 4.14: 重複性資料移除

- 若不同來源的統計數值彼此所使用的計量單位具有一定的換算率，則可以進行計量單位與統計數值的轉換計算 (圖 4.15)。

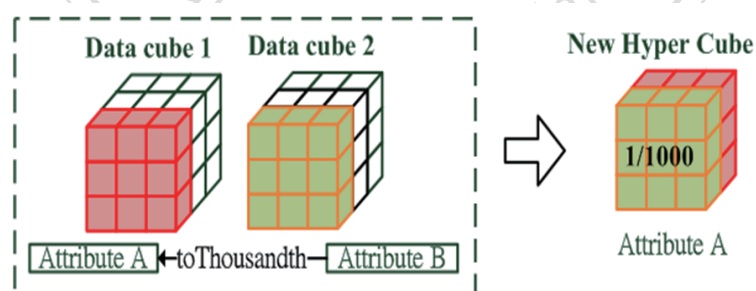


圖 4.15: 計量單位換算

- 不同來源的資料項其資料維度彼此具有水平的 *hasSameAs* 關係鏈則進行水平維度轉換，將資料項原始維度轉換至查詢條件指定的維度 (圖 4.16)。
- 查詢的結果資料集合中，資料項與查詢條件具有垂直的階層關係，則可以將資料維度捲起 (*Roll-Up*) 至查詢條件指定的維度並將資料數值進行合併運算 (圖 4.17)。

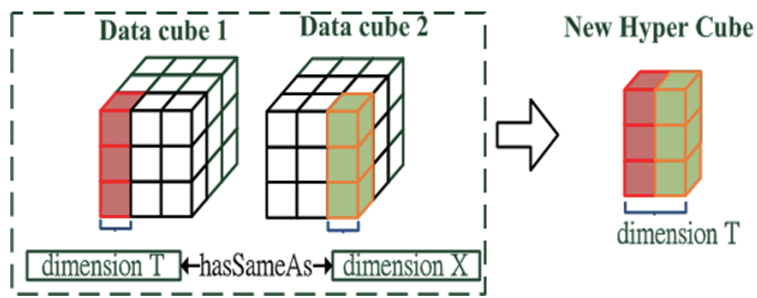


圖 4.16: 水平維度轉換

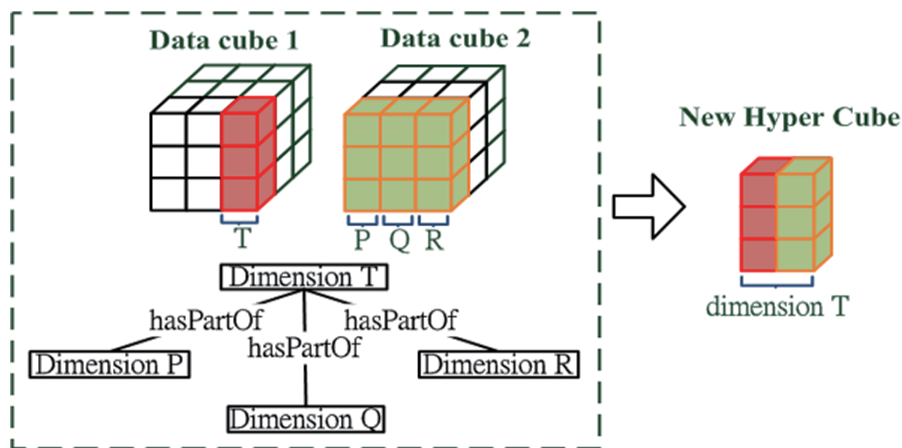


圖 4.17: 垂直維度合併

在完成一連串的资料重整程序後，查詢結果已彙整成精簡的资料集合，雖然是如此但仍可能有無法滿足原始查詢條件期望得到的結果，無法滿足的情況有兩種，一種是來源資料的不足，原因是在開放式架構下，整合架構所登錄的资料源不一定能完整蒐集，因此查詢彙整時有可能無法取得資料並彙整至原始查詢條件，另外一種情形則是資料描述顆粒的細緻程度不同產生的差異，顆粒細緻度指的就是描述單一資料項的維度數量，單一資料項所具有的維度數量越多則代表其資料描述的細緻程度越高，因此當查詢條件的維度數量與實際資料項的維度數量不同時，便會產生細緻度上的差異，而資料項在整合的過程中會進行資料數值的運算與維度描述的階層提升，但不會改變原始維度的數量，這兩種類型的查詢結果雖然可能造成與期望整合的結果有所差異，但整合平台仍會將這類型的資料輸出，用來提供內容上的比較判讀參考，使用者最後則可以透過修改條件與反覆查詢來修正至最佳的整合結果。

第 5 章 整合程序與實驗平台設計

透過前一個章節所設計的整合方法，本研究實作了一個以本體論為描述基礎的整合程序與查詢平台，整合程序共區分為 3 個主程序，其中有資料源建立程序 (*Create Cube Process*)、查詢程序 (*Query Process*)、建立查詢結果程序 (*Build Result Process*)，個別的主程序則還包含了不等的子程序 (圖 5.2)，查詢平台的結構則設計了幾個不同的模組區塊，其中包括查詢模組 (*Query Module*)、不同資料源個別的中介轉換模組 (*wrapper*)、整合知識庫 (*Integration Library*)、資料重整模組 (*Reorganize Module*) (圖 5.1)，資料整合的核心概念是當描述資料的格式非常的多元，如資料庫、電子化表格等，而若不同的資料格式都能夠轉換為單一格式，則較為容易達成整合的目的，以下則分別說明各子程序所運用的概念，而各階段的詳細設計方法可參考前一章節的完整說明。

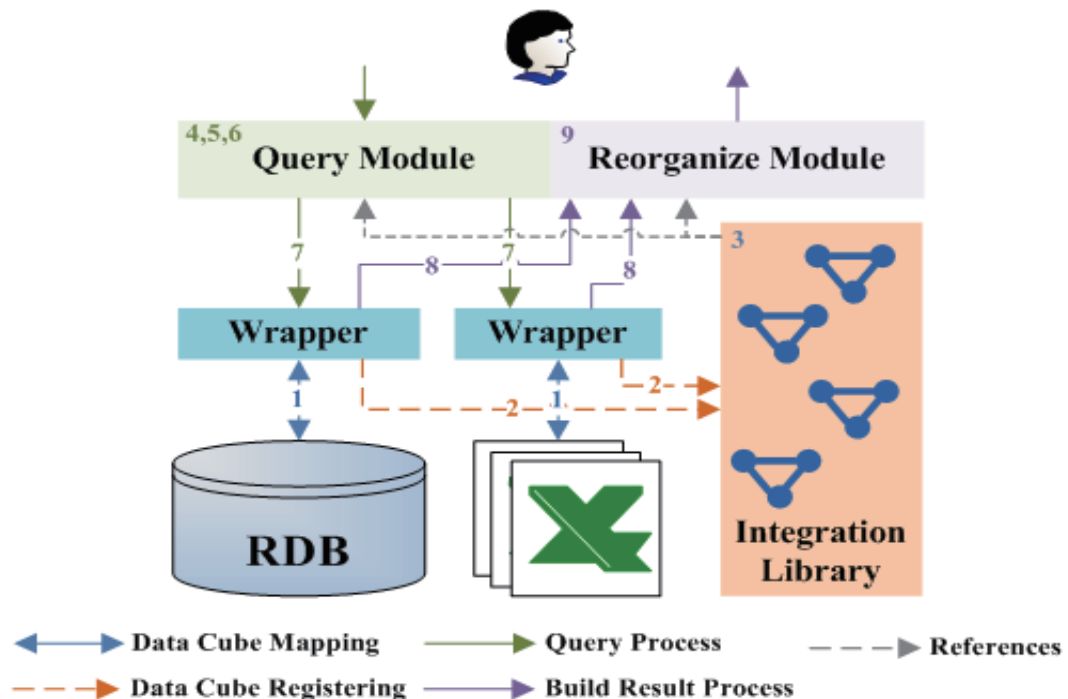


圖 5.1: 多維度整合平台架構與整合流程

- 資料格式對應 (*Data Cube Mapping*) - 不同來源的資料對象都可以運用 *Wrapper* 來提供查詢的機制，資料源不需要事先轉換為本體論結構，而是利用查詢機制動態將查詢結果轉換為本體論結構即可，在本體論的轉換對應上

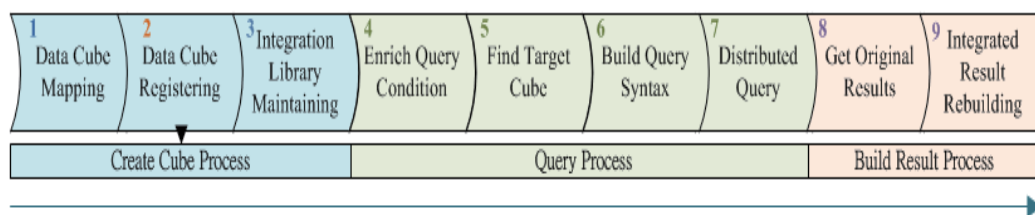


圖 5.2: 多維度整合程序

是使用 *RDF Data Cube Vocabulary* 作為描述的字彙，不同格式的統計資料都可以利用 *Cube* 描述字彙轉換為多維度的 *Data Cube* 結構，雖然不同來源的資料都利用 *Wrapper* 來轉換為相同的 *Data Cube* 描述結構，但不同資料源仍可以保有其資料的獨立描述特性，也就是特徵元素 (如資料維度、計量單位) 的描述上可自行定義不須與其他資料源交互參照 (章節 4.1)。

- 資料源註冊 (*Data Cube Registering*) - 在整合方法的塑模設計中，不同的資料來源透過資料對應及中介軟體 (*Wrapper*) 的轉換後都能夠提供 *SPARQL* 的查詢服務，而為了要能夠利用單一平台來完成整合，必須先讓平台蒐集及記錄可查詢對象的資訊，因此要進行資料源的註冊程序，在開放式的環境下資料來源可能非常多元且不定，設計註冊程序的優點是可以容易地進行多重資料源的加入與移除，註冊程序分成兩個階段，第一階段先將資料源位址 *URL* 存放至整合知識庫內，在整合知識庫內是利用類別 *gq:DataCube* 來描述資料源，註冊程序的第二階段是利用 *SPARQL* 自動查詢的方式對已註冊的資料源進行蒐集 *Cube* 特徵資訊，如資料維度、計量單位等，整個註冊程序完成後整合知識庫內便存放了資料源的位址及其完整的 *Cube* 特徵資訊，這些資訊便可以在後續關聯性維護及整合查詢時取出運用，
- 維護整合知識庫 (*Integration Library Maintaining*) - 在本研究所設計的整合架構下，可整合的對象都透過對應轉換的方式可以提供本體論的查詢結果，同時資料源也利用註冊程序位址及元素特徵存放在整合知識庫內，因此為了進行整合的需要，知識庫內也必須建立及維護這些特徵元素彼此間的關係鏈描述，例如不同資料維度間的水平轉換與垂直合併關聯性、不同計量單位間的換算關聯性等 (章節 4.2)。
- 條件擴充、對象確認、語法重建及分散式查詢 (*Enrich Query Condition, Find Target Cube, Build Query Syntax, Distributed Query*) - 本研究不是事先將個

別資料源整合彙整為單一資料集，而是透過動態查詢的方式來完成整合的目的，當透過查詢來進行資料項 (資料數值) 萃取時，因個別資料集是各自描述其資料語意並且彼此可能並不相容，所以單一的查詢語法很難包含不同的資料對象，因此我們利用整合知識庫內的關連性描述來進行查詢語法的改寫 (*Query Syntax Rewriting*)，透過分析原始查詢條件的方式將原始查詢條件內所未涵蓋到而又與原始條件具有一定關聯性的條件也包含進來，利用改寫後的查詢語法再從不同的資料源中萃取切割出適當的資料項 (章節 4.3)。

- 取回原始查詢結果及重建整合資料 (*Get Original Results, Integrated Result Rebuilding*) - 透過改寫後的查詢語法可以將資料從不同的資料源中萃取出來，但這些萃取出來的資料仍是維持其原始描述內容，因此若是要得到符合原始查詢條件所期望得到的整合結果必須再進行資料的重整，除了重複性資料的刪除及計量單位的換算外，還必須利用整合知識庫內的關聯性資訊進行結果資料項水平的轉換與垂直的合併，最後重建出符合原始查詢期望的新資料集 (章節 4.4)。

關於查詢語法改寫的程序實際執行運用的演算法說明如下，在演算法 1 中程序 *getExtQD* 及 *getExtQA* 利用原始的查詢條件從整合知識庫中分別找出延伸的資料維度集合及可進行數值單位換算的計量單位集合，程序 *getByDAndA* 則利用擴增的條件集合從知識庫的資料對象註冊區中找出包含這些條件的對象集合，程序 *simplifyCubes* 則是分析這些可查詢的對象集合，若資料對象彼此間具有包含於 (*includeOf*) 的關聯性則必須再進一步篩選查詢對象，由查詢者從彼此有包含關係的對象集合中擇一使用，避免將具有包含關係的資料同時被查詢出來造成具有相同意涵的資料被彙總，程序 *BindF* 則是針對不同的查詢對象改寫出適合的查詢語法進行查詢，語法改寫的方式是透過利用 *SPARQL1.1* [6] 定義的條件 *Binding* 語法來完成，將擴增後的查詢條件與查詢變數進行 *Binding*，利用迴圈的方式針對各別資料對象改寫產生各自適用的查詢語法進行查詢，最後將取得查詢結果合併輸出。

Algorithm 1 查詢語法改寫與執行程序

$Cubes$ (查詢對象集合) ($[B_1 \ \dots \ B_r]$) $\in QB$

Require:

$sparql_o$ (原始查詢語法)

$QDimensions$ (查詢條件的維度集合) ($[D_1 \ \dots \ D_r]$) $\in QDimensions$

$QAttributes$ (查詢條件的單位集合) ($[A_1 \ \dots \ A_r]$) $\in QAttributes$

$ExtQD \leftarrow getExtQD(QDimensions)$ \triangleright 從整合知識庫內取得延伸維度集合

$ExtQA \leftarrow getExtQA(QAttributes)$ \triangleright 從整合知識庫內取得延伸單位集合

$Cubes \leftarrow getByDAndA(ExtQD, ExtQA)$ \triangleright 透過延伸維度及延伸單位集合找出可查詢資料對象

$Cubes \leftarrow simplifyCubes(Cubes)$ \triangleright 簡化查詢對象集合

for each cube in $Cubes$ **do**

$sparql_b \leftarrow BindF(sparql_o, cube, ExtQD, ExtQA)$ \triangleright 依個別查詢對象產生

SPARQL 查詢語法

$RESULT \leftarrow ExecuteQuery(sparql_b, cube)$ \triangleright 取回查詢結果

$RESULT_o \leftarrow RESULT_o + RESULT$ \triangleright 將個別查詢結果加入結果集合中

end for

return $RESULT_o$ \triangleright 回傳查詢結果集合

重整程序的演算法說明如下，在演算法 2 中先將在查詢階段得到的資料 QR 利用程序 *AttributeTrans* 進行計量單位的轉換，*AttributeTrans* 轉換的方式是利用整合知識庫內的關聯性描述 $gq:ExchangeRelation$ 將使用擴充條件的計量單位的資料項轉換至與原始查詢條件，程序 *SameAsTrans* 則是利用水平關聯性描述將運用擴充維度條件所描述的資料項轉換至同一垂直階層，完成以上兩個步驟後再利用程序 *RemoveDuple* 將初始整合產生的重複性資料剔除，最後在重整的初始階段資料項集合僅剩下垂直階層的關聯性尚未處理，接下來程序 *SplitByAttAndDimCount* 則是將資料以計量單位及資料維度數量的組合進行子集合的分群，各別子集合內的資料項都是同一計量單位且同一維度數量，利用區分子集合的方式在重整結果最後可以得到將不同計量基礎的資料放在一起比較的顯示效果，而不同維度數量的資料則有可能是因為不同子集合所描述資料的顆粒

細緻度有差異，而較粗顆粒 (即維度數量較少) 的資料不一定能完整取代較細緻顆粒 (即維度數量較多) 的資料群，因此不同細緻度的資料項會分開進行彙整程序，最後都會一併顯示在輸出的整合結果中。

完成子集合分群後，依序對各別的子集合進行合併運算的程序，首先利用程序 *getNQD* 找出子集合內非原始查詢條件的維度集合 *NQD*，在資料合併運算的程序中會針對原始查詢條件 (*QC*) 及非原始查詢條件 (*NQD*) 使用不同的合併策略，非原始條件 (*NQD*) 所運用的策略是盡可能整合至最上層維度，利用程序 *getButtonUpTree* 來取得以 *NQD* 為葉節點的維度樹集合，運用這個維度樹集合盡可能將非查詢維度的部分彙總至該維度樹的最上層，而對於原始查詢條件所運用的策略則是盡可能將資料整合至條件所指定的階層即可，因此利用程序 *getTopDownTree* 取得以查詢條件 *QC* 為根節點的維度樹集合並盡可能將資料彙總合併至根結點。

Algorithm 2 查詢結果重整程序

D(資料維度), *Dimension A*(計量單位), *Attribute M*(資料數值), *Measure*

OB(Cube 資料項), $([D_1 \dots D_n], A, M) \in OB$

NQD(非查詢條件的維度集合) $([D_1 \dots D_r]) \in NQD$

Require:

QC(查詢條件), $([D_1 \dots D_i], [A_1 \dots A_j]) \in QC$

QR(待重整資料項集合), $([OB_1 \dots OB_n]) \in QR$

QR \leftarrow *AttributeTrans*(*QR*) ▷ 計量單位換算

QR \leftarrow *SameAsTrans*(*QR*) ▷ 水平維度轉換

QR \leftarrow *RemoveDuple*(*QR*) ▷ 重複性資料刪除

CC \leftarrow *SplitByAttAndDimCount*(*QR*) ▷ 區分可重整子集合

for each *CB* **in** *CC* **do**

NQD \leftarrow *getNQD*(*QC*, *CB*) ▷ 取出非原始查詢維度的延伸維度集合

TS_a \leftarrow *getButtonUpTrees*(*NQD*) ▷ 取得延伸維度的各別維度樹集合

TS_b \leftarrow *getTopDownTrees*(*QC*) ▷ 取得原始查詢維度的各別維度樹集合

CB \leftarrow *Merge*(*CB*, *TS_a* + *TS_b*, *false*) ▷ 進行子集合內資料彙整運算

QR \leftarrow *QR* + *CB* ▷ 將重整後子集合加入結果集合

end for

return *QR* ▷ 回傳重整結果集合

演算法 3 是被演算法 2(查詢結果重整程序) 所呼叫，主要目的是實際執行資料項的合併判斷程序，在輸入的維度樹集合條件中，依序掃描個別維度樹內的各階層，利用程序 *setMergeMark* 以維度階層為單位，依序判斷資料項集合是否有包含完整的階層資料項，若有則設定該資料維度為可合併維度，同一階層判斷完成後將可合併資料進行彙整合併計算，最後重複執行彙整運算至所有的維度樹階層都檢查判斷完成為止。

Algorithm 3 子集合內資料項彙整運算程序

Require:

CB (待合併資料項集合), $([OB_1 \dots OB_n]) \in CB$

TS_t (維度樹集合) $([T_1 \dots T_k]) \in TS_t$

isFinish 是否還有待合併資料項

if *isFinish* = *true* **then**

 Break

else

for each T_i in TS_t **do**

for each part in T_i **do**

$CB \leftarrow setMergeMark(CB, part)$ ▷ 以單一階層為條件依序判斷待整

合對象是否有具有階層內的所有維度，若有則設定可合併註記

end for

end for

$CB \leftarrow RemoveDuple(CB)$

▷ 移除重複性資料

if *CanMerge*(CB) **then**

$CB \leftarrow MergeCube(CB)$

▷ 將可合併資料 Roll-Up 至上層階層

else

isFinish = *true*

end if

Merge($CB, TS_t, isFinish$);

▷ 重複執行合併程序

end if

第 6 章 平台實作與驗證

6.1 建立多維度整合平台

為了驗證本研究關於多維度統計資料整合論述的架構及整合程序 (圖 5.2)，建立了一個 Web 架構的多維度整合平台，其中包含了兩個重要的頁籤 *Integrate Query* 及 *Configuration*，其中 *Configuration* 頁籤主要是提供進行整合程序中的第二步 *Data Cube Registering* 程序 (圖 5.2)，運用註冊的登錄的方式可以很彈性地讓能提供 *SPARQL Endpoint Service* 查詢機制的資料源動態加入多維度整合平台內 (圖 6.1)。

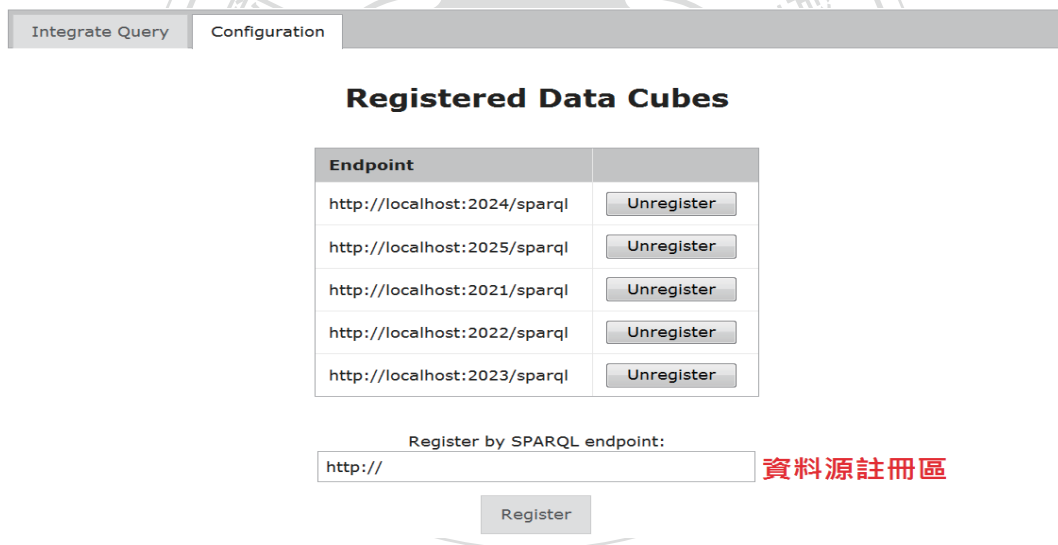


圖 6.1: 資料源註冊功能

Integrate Query 頁籤則是執行整合查詢的主要入口畫面，其中包含了有資料維度選擇區、計量單位選擇區、資料集選擇區、*SPARQL* 查詢語法輸入區、查詢結果區及資料重整區等六個主要區塊 (圖 6.2)，執行整合程序時先在資料集選擇區、資料維度選擇區及計量單位選擇區設定查詢條件，若資料集選擇區沒有設定則整合架構會將所有登錄資料集加入至可能的查詢對象集合中，最後在 *SPARQL* 查詢語法輸入區完成完成查詢語法的編輯，執行送出後便開始執行查詢程序，平台會自動利用資料維度選擇區及計量單位選擇區所選擇的條件集合為基礎，再透過整合知識庫內所維護的關聯性描述執行查詢語法改寫程序 (章節 4.3)，最後再從

查詢對象集合中找出可以整合的對象進行查詢，完成後再將各別查詢的結果於結果輸出區內整批顯示。

The screenshot shows a web interface titled "Integrate Query" with a "Configuration" tab. It features several input fields and buttons:

- Select Dimension 資料維度選擇區**: A dropdown menu with "Var" and "URI" options.
- Select Attribute 計量單位選擇區**: A dropdown menu with "Var" and "URI" options.
- Select Datasource 查詢對象選擇區**: A dropdown menu with "URI" selected.
- SPARQL查詢語法輸入區**: A text input field containing the query: `select * where {?s ?p ?o} limit 10`.
- 查詢結果輸出區**: Buttons for "Go!" and "Reset".
- 資料重整輸出區**: Radio buttons for "SUM" and "AVG", and a "Reorganize" button.

Below the "Reorganize" button, there are labels for "Before Reorganize:" and "After Reorganize:".

圖 6.2: 整合查詢功能

完成整合查詢後，在查詢結果區內的資料仍呈現分散獨立的描述架構，因此必須再透過執行資料重整程序 (章節 4.4) 將結果資料集合重新整理，最後將重整的結果顯示於資料重整結果區，作後利用虛擬化圖表工具來顯示結果以供資料判讀及利用。

6.2 測試範例說明

整合測試平台以目前台灣電子化政府所公開的資訊作為測試參考對象，並將參考資料作適當的簡化與修改以便能夠在資料複雜度較低的情況下完整測試整合程序的每個步驟，測試的對象包含了資料庫與電子化表格兩個不同格式的儲存方式，其中資料庫的部分選擇使用中華民國統計資訊網 (<http://www.stat.gov.tw/>) 的統計資料庫內的「來台旅客人數統計」作為模擬的資料庫測試標的，而電子化表格

的部分則選擇交通部觀光局 (<http://admin.taiwan.net.tw/>) 所公布的「觀光類重要參考指標」及中華民國行政院主計處 (<http://www.dgbas.gov.tw/>) 所公佈的「營利事業家數及銷售額」、「各行業勞動生產力指數年增率」以及經濟部統計處的「製造業受雇員工與工資」等公開統計資料做為模擬標的，將資料對應轉換至 *Data Cube* 結構後登錄至整合知識庫內，登錄完成後在知識庫內建立了五個資料源、九個不同的資料維度及四種計量單位。

6.2.1 範例一：2010 年 1 月亞洲國家與歐洲國家入境台灣的觀光人數

本範例選用「觀光類重要參考指標」(表 6.1) 及「來台旅客人數統計」(表 6.2) 兩個資料集，以下分別以整合步驟的重點程序說明整合的過程。

本體論對應

表 6.1: 觀光類重要參考指標

	日本	美國	中國
99 年 1 月	86	29	87
99 年 2 月	74	28	108

資料來源 1 對應到 *Data Cube* 本體論：

prefix ec: <<http://sw.cs.nccu.edu/datasource-1#>> .

prefix qb: <<http://purl.org/linked-data/cube#>> .

ec: *obs1* *a* *qb*: *Observation*;

qb: *dataSet* *ec*: *dataset1*;

ec: *dimension* – *ROC_date* "*ROC – 99 – 01*";

ec: *dimension* – *country* "*Japan*";

ec: *attribute* – *people_amount* "*ThousandPeople*";

ec: *measure* – *quantity* 86;

.

表 6.2: 來台旅客人數統計

Date	Region	Value
2010M1	America	34362
2010M1	Europe	13854
2010M2	America	34384
2010M2	Europe	13726

資料來源 2 對應到 *Data Cube* 本體論：

prefix ec: <<http://sw.cs.nccu.edu/datasource-2#>> .

prefix qb: <<http://purl.org/linked-data/cube#>> .

```
ec : obs1 a qb : Observation;
qb : dataSet ec : dataset1;
ec : dimension - West_date "2010 - 01";
ec : dimension - Continent "America";
ec : attribute - people_count "People";
ec : measure - amount 34362;
.
```

查詢語法改寫與執行

進行整合查詢時在 *SPARQL* 查詢語法輸入區輸入查詢條件如下，維度變數 *v1* 的條件是 "2010-01"，維度變數 *v2* 的條件是 "Asia" 及 "Europe"，計量單位變數 *attv* 的條件是 "ThousandPeople"，數值計算方式選擇「SUM」。

查詢條件：

PREFIX rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>>

PREFIX dq: <<http://purl.org/linked-data/cube#>>

SELECT ?v1 ?v2 ?val ?attv

WHERE {

?bnk1 dq:dimension ?dim1 . ?bnk2 dq:dimension ?dim2 .

?bnk3 dq:measure ?mea . ?bnk4 dq:attribute ?att .

?ob rdf:type dq:Observation .

```

?ob ?dim1 ?v1 .
?ob ?dim2 ?v2 .
?ob ?mea ?val .
?ob ?att ?attv .
}

```

透過從維度選擇區及計量單位選擇區取得得原始條件，整合平台再透過整合知識庫進行擴充條件的查詢，查詢語法範例如下：

- 資料維度

- 水平維度: ROC-99-01

```

SELECT ?o ?label
WHERE
{
  <http://sw.cs.nccu.edu.tw/mdip.rdf#West_2010_01>
    <http://sw.cs.nccu.edu.tw/mdip.rdf#hasSameAs> ?o .
  ?o <http://www.w3.org/2000/01/rdf-schema#label> ?label }

```

- 垂直維度: Japan 、 China

```

SELECT ?o ?label
WHERE
{
  <http://sw.cs.nccu.edu.tw/mdip.rdf#Asia>
    <http://sw.cs.nccu.edu.tw/mdip.rdf#hasPartOf> ?o .
  ?o <http://www.w3.org/2000/01/rdf-schema#label> ?label }

```

- 計量單位: People

```

select ?att ?label
where
{
  ?ext <http://www.w3.org/2000/01/rdf-schema#subPropertyOf>
    <http://sw.cs.nccu.edu.tw/mdip.rdf#exchangeRelation> .
  <http://sw.cs.nccu.edu.tw/mdip.rdf#ThousandPeople> ?ext ?att .
  ?att <http://www.w3.org/2000/01/rdf-schema#label> ?label . }

```

原始的查詢條件加上擴增後的條件 *Binding* 至原始的查詢語法並針對不同的查詢對象各別建立出新的查詢語法，範例如下：

```
SELECT ?v1 ?v2 ?val ?attv
WHERE
{
  ?bnk1 dq:dimension ?dim1 . ?bnk2 dq:dimension ?dim2 .
  ?bnk3 dq:measure ?mea . ?bnk4 dq:attribute ?att .
  ?ob rdf:type dq:Observation .
  ?ob ?dim1 ?v1 .
  ?ob ?dim2 ?v2 .
  ?ob ?mea ?val .
  ?ob ?att ?attv }
BINDINGS ?v1 ?v2 ?attv
{
  ("Japan" "ROC-99-01" "People")("China" "ROC-99-01" "People")
}
```

完成查詢語法的改寫與執行後產生初始的查詢結果，結果中各別的资料項仍維持原始的資料描述，如維度上有西元年與民國年的差異，計量單位則有千人與人的差異，除此之外查詢結果資料還沒彙整為查詢條件所設定的“Asia”條件。

Result:

v1	val	v2	attv
Japan	86	ROC-99-01	Thousand_People
China	87	ROC-99-01	Thousand_People
Europe	13854	2010-01	People

圖 6.3: 範例 1-資料查詢結果

查詢結果重整

最後再透過資料重整的程序將資料維度及計量單位進行轉換，彙整的部分則將“Japan”及“China”合併建立出“Asia”資料並產生比較圖表。

After Reorgnize:

v1	val	v2	attv
Asia	173	ROC-99-01	Thousand_People
Europe	13.854	ROC-99-01	Thousand_People

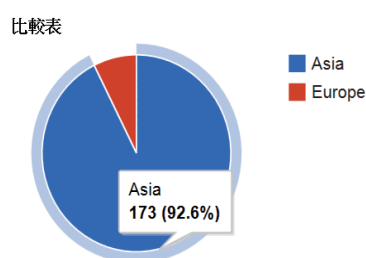


圖 6.4: 範例 1-查詢結果重整

6.2.2 範例二:2009 年與 2010 年製造業家數與生產力指數年增率比較

本範例選用「營利事業家數及銷售額」(表 6.3) 及「各行業勞動生產力指數年增率」(表 6.4) 兩個資料集，以下分別以整合步驟的重點區塊說明整合的過程。

本體論對應

表 6.3: 營利事業家數及銷售額

年度	家數	產業別
2009	393	電力及然氣供應業
2009	128458	製造業
2009	12191	農林漁牧業
2010	415	電力及然氣供應業
2010	130210	製造業
2010	12240	農林漁牧業

資料來源 3 對應到 Data Cube 本體論：

prefix ec: <<http://sw.cs.nccu.edu/datasource-3#>> .

prefix qb: <<http://purl.org/linked-data/cube#>> .

ec: obs1 a qb: Observation;
qb: dataSet ec: dataset3;
ec: dimension – West_year "2009";
ec: dimension – Industry "Electricity_Gas_Supply";
ec: attribute – company_count "company_count";
ec: measure – amount 393;

表 6.4: 各行業勞動生產力指數年增率

	98 年	99 年
電力及然氣供應業	-5.15	3.68
製造業	0.57	17.24
礦業及土石採取業	1.46	16.77

資料來源 4 對應到 *Data Cube* 本體論：

prefix ec: <<http://sw.cs.nccu.edu/datasource-4#>> .

prefix qb: <<http://purl.org/linked-data/cube#>> .

ec: obs1 a qb: Observation;
qb: dataSet ec: dataset4;
ec: dimension – ROC_year "ROC – 98";
ec: dimension – Industry "Electricity_Gas_Supply";
ec: attribute – percentage "percentage";
ec: measure – ratio –5.15;

查詢語法改寫與執行

進行整合查詢時在 *SPARQL* 查詢語法輸入區中第一維度條件選擇 "2009"、"2010"，第二維度條件選擇 "Manufacturing"，計量單位則選擇 "company_count" 及 "percentage"，之後透過整合知識庫內的關聯性描述將查詢條件擴增及 *Binding*

至原始條件並產生新的查詢語法進行查詢，範例如下：

```
SELECT ?v1 ?v2 ?val ?attv
```

```
WHERE
```

```
{  
  ?bnk1 dq:dimension ?dim1 . ?bnk2 dq:dimension ?dim2 .  
  ?bnk3 dq:measure ?mea . ?bnk4 dq:attribute ?att .  
  ?ob rdf:type dq:Observation .  
  ?ob ?dim1 ?v1 .  
  ?ob ?dim2 ?v2 .  
  ?ob ?mea ?val .  
  ?ob ?att ?attv }  
}
```

```
BINDINGS ?v1 ?v2 ?attv
```

```
{  
  ( "2009" "Manufacturing" "company_count")( "2010" "Manufacturing" "com-  
  pany_count")  
}
```

初始查詢結果如下：

Result:

v1	val	v2	attv
2009	128458	Manufacturing	Company_Count
2010	130210	Manufacturing	Company_Count
ROC-98	-0.57	Manufacturing	percentage
ROC-99	17.24	Manufacturing	percentage

圖 6.5: 範例 2-資料查詢結果

查詢結果重整

初始結果資料集在經過彙整重建程序後產生比較資料及圖表。

After Reorgnize:

v1	val	v2	attv
2009	128458	Manufacturing	Company_Count
2010	130210	Manufacturing	Company_Count
2009	-0.57	Manufacturing	percentage
2010	17.24	Manufacturing	percentage

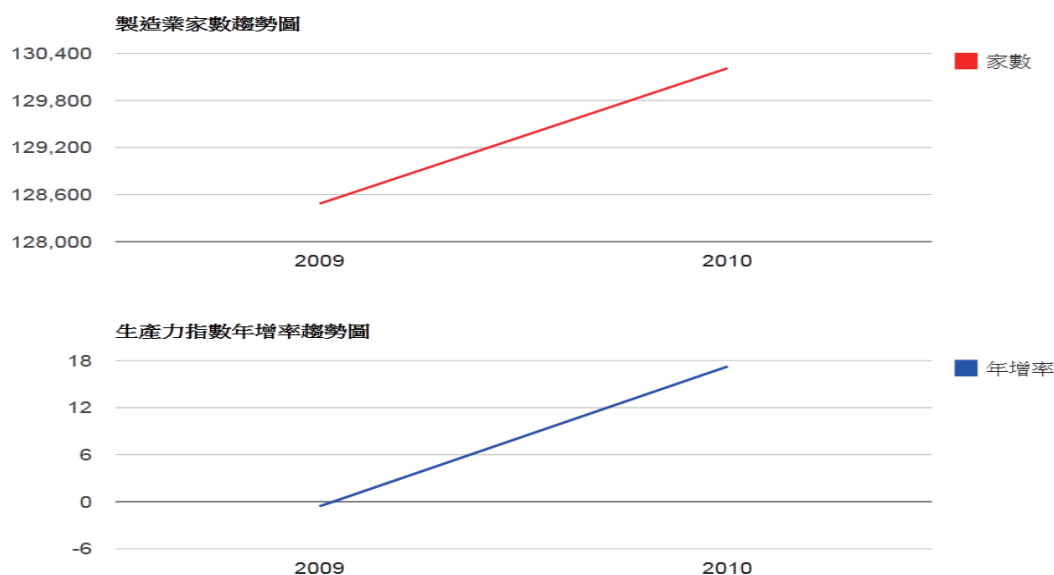


圖 6.6: 範例 2-查詢結果重整

6.2.3 範例三:2011 年與 2012 年生產力指數年增率與受雇員工工資比較

本範例選用「各行業勞動生產力指數年增率」(表 6.4)及「製造業受雇員工與工資」(表 6.5)兩個資料集，數值計算方式選擇「AVG」，以下分別以整合步驟的重點區塊說明整合的過程。

本體論對應

資料來源 5 對應到 *Data Cube* 本體論：

prefix ec: <<http://sw.cs.nccu.edu/datasource-5#>> .

prefix qb: <<http://purl.org/linked-data/cube#>> .

ec: *obs1* *a* *qb*: *Observation*;

qb: *dataSet* *ec*: *dataset5*;

表 6.5: 製造業受雇員工與工資

年月	受雇員工人數	平均薪資
10101	2632760	83789
10102	2636747	40010
10103	2639299	38447

```

ec : dimension – ROC_date    "ROC – 101 – 01";
ec : dimension – Industry    "Manufacturing";
ec : attribute – people     "People";
ec : measure – count       2632760;
.

```

查詢語法改寫與執行

進行整合查詢時在 SPARQL 查詢語法輸入區中第一維度條件選擇"2011"、"2012"，第二維度條件選擇"Manufacturing"，計量單位則選擇"People"及"percentage"，之後透過 *Integration Library* 將查詢條件擴增及 *Binding* 至原始條件並產生新的查詢語法進行查詢，範例如下：

```

SELECT ?v1 ?v2 ?val ?attv
WHERE
{
  ?bnk1 dq:dimension ?dim1 . ?bnk2 dq:dimension ?dim2 .
  ?bnk3 dq:measure ?mea . ?bnk4 dq:attribute ?att .
  ?ob rdf:type dq:Observation .
  ?ob ?dim1 ?v1 .
  ?ob ?dim2 ?v2 .
  ?ob ?mea ?val .
  ?ob ?att ?attv }
BINDINGS ?v1 ?v2 ?attv
{
  ("ROC-101-01" "Manufacturing" "People")
  ("ROC-101-02" "Manufacturing" "People")

```

初始查詢結果如下：

After Reorgnize:

v1	val	v2	attv
ROC-100-10	2640591	Manufacturing	People
ROC-100-11	2640374	Manufacturing	People
ROC-100-12	2639061	Manufacturing	People
ROC-101-01	2632760	Manufacturing	People
ROC-101-02	2636747	Manufacturing	People
ROC-101-03	2639299	Manufacturing	People
ROC-100	3.42	Manufacturing	percentage

圖 6.7: 範例 3-資料查詢結果

查詢結果重整

進行資料合併時，「各行業勞動生產力指數年增率」無 2012 年的資料，而「製造業受雇員工與工資」的資料內容中 2012 的資料只有 1 至 3 月份，無法透過合併建立起完全代表 2012 年的資料，因此這部分的資料會被放置於未整合 (*Invalid*) 區域，最後由使用者判讀是否需增加資料來源或修改查詢條件重新查詢。

Result:

v1	val	v2	attv
2011	2618469	Manufacturing	People
2011	3.42	Manufacturing	percentage

Invalid

v1	val	v2	attv
ROC-101-01	2632760	Manufacturing	People
ROC-101-02	2636747	Manufacturing	People
ROC-101-03	2639299	Manufacturing	People

圖 6.8: 範例 3-查詢結果重整

第 7 章 結論與未來展望

在透明化與資訊公開的環境日益成熟下，大量的政府統計資料被產製與發佈，但是在資料格式與語意描述不一致的情況下加深了資料運用的困難度，因此本研究利用語意網技術來進行政府公開資訊中統計資料的整合，透過本研究所設計的整合程序及整合平台來解決資料格式多元且語意不一致的問題。

本研究運用語意網技術將不同格式的資料轉換為多維度本體論描述結構，透過資料的對應轉換以及配合本研究設計的整合查詢介面，能夠讓原本存放在封閉格式內的資料不再以「檔案」為發佈單位，而是以數值本身為最小查詢單位來提供開放式整合運用，透過關聯性描述知識庫的建立，不同資料來源不需要事先進行全資料的合併，而是可以依照不同整合的需要，選擇適當的資料維度與計量單位作為查詢條件進行整合，在查詢程序中利用自動化的方式找出適當的資料對象，並透過分散式查詢的方式從適當的資料對象中抽取出所需的資料數值進行整合，查詢的結果最後再透過重整程序修正不同資料對象彼此語意描述的差異，除此之外利用圖形化工具的輔助讓整合結果呈現出更豐富的顯示模式。

雖然在開放式的環境下，資料的整合可能會因資料來源的不足或是語意描述細緻度上的差異，造成無法整合至符合原始查詢條件的設定，但即便無法完整彙整成期望的結果集合，利用未整合的資料集合仍可以提供使用者作為修改查詢條件與蒐集新資料對象的參考資訊。

關於未來可能的研究展望可以分為幾個不同的方向，在公開資訊方面可以建議政府在發佈資料的同時一併發佈資料的描述框架，如此可以便於利用資訊技術進行再運用及加值，在多維度整合方面則是將更多不同類型的資料源，如半結構化的網頁等也加入整合範圍，而在資料內涵的解讀能力方面則可以加入領域專家的註解能力來提高整合結果的可用性，亦或是透過多樣的圖形化工具提供整合結果更豐富的呈現。

參 考 文 獻

- [1] 葉俊榮、許宗力. 政府資訊公開制度之研究. 台北：行政院研考會 (1996).
- [2] Berners-Lee, T. *Putting government data online*. <http://www.w3.org/DesignIssues/GovData.html>, 2009.
- [3] Bizer, C. *D2rq - treating non-rdf databases as virtual rdf graphs*. In ISWC (2004).
- [4] Davies, A., and Lithwick, D. *Government 2.0 and access to information: 2. recent developments in proactive disclosure and open data in the united states and other countries*. Ottawa, Canada: Library of Parliament.
- [5] Han, L., Finin, T., Parr, C., Sachs, J., and Joshi, A. *RDF123: a mechanism to transform spreadsheets to RDF*. Tech. rep., University of Maryland, Baltimore County, August 2007. Technical Report.
- [6] Harris, S., and Seaborne, A. *SPARQL 1.1 Query Language*. <http://www.w3.org/TR/sparql11-query/>, 2010.
- [7] Hausenblas, M., Halb, W., Raimond, Y., Feigenbaum, L., and Ayers, D. *Scovo: Using statistics on the web of data*. In ESWC (2009).
- [8] Jeni Tension, T. *The rdf data cube vocabulary*. W3C Working Draft, 2012.
- [9] Kampgen, B., and Harth, A. *Transforming statistical linked data for use in olap systems*. In I-SEMANTICS (2011).
- [10] Langegger, A., and Wöß, W. *SemWIQ - semantic web integrator and query engine*. In GI Jahrestagung (2) (2008).
- [11] Langegger, A., and Wöß, W. *XLWrap - querying and integrating arbitrary spreadsheets with SPARQL*. In ISWC (2009).
- [12] Orszag, P. R. *Open government directive*. Executive Office of the President, Office of Management and Budget, Memorandum for the Heads of Executive Departments and Agencies, Washington, DC, December 8.

[13] Prud'hommeaux, E., and Seaborne, A. *SPARQL query language for rdf*. W3C Recommendation 4 (2008), 1--106.

[14] Tauberer, J. *Open data is civic capital: Best practices for 'open government data'*. Version 1.1, 20 July, 2009.

[15] Treasury, H. M. *Putting the frontline first: smarter government*. Cm 7753.

