

異質性資訊源環境中的資料擷取：

以調節器為基礎的方法*

A Mediator-based Approach for Data Access in the Environment of Heterogeneous Information Sources

朱雨其
Yu-Chi Chu

嚴國瑞
Kuo-Ruei Yen

江明洲
Ming-Chou Jiang

楊鍵樵
Chen-Chau Yang

台灣科技大學電子工程系

Dept. of Electronic Engineering

National Taiwan University of Science and Technology

ycchu@sun.epa.gov.tw ccyang@selab3.et.ntust.edu.tw

摘要

本文旨在探討運用調節器的設計原理與查詢處理方式以整合異質性資訊源。我們提出「資訊需求」的概念模型作為資訊源語意整合時選取資訊源的依據，此模型可減少資訊整合時無用資料的產生。此外，我們並運用一個植基於知識系統的兩階層式超資料伺服器作為調節器的輔助工具，以處理異質性資訊源整合與查詢處理的問題。超資料伺服器的領域知識層描述系統應用時所需的相關知識，而資訊源模式層則保留整合資訊源的資料語意，並以此兩階層的對應關係作為系統的共同資料模式。調節器除了處理使用者之語意查詢，並結合包裝器將查詢轉換成對應的資訊源查詢語句。我們提出不受資訊源影響的資訊源選取對應機制以及查詢語意剖析解譯演算法，將查詢語句分解為對應各資訊源的子查詢。

關鍵詞：異質性資訊源，調節器，資訊需求，超資料伺服器，包裝器

ABSTRACT

The design principles and query processing of mediator for integrating heterogeneous information sources are presented. We proposed a conceptual model of "information in demand" to serve as a reference of the selection of information sources when they are integrated. In addition, we adopt a knowledge-based two-level metadata server which plays as a role of assistant in cooperate with the mediator to handle the problems of integration and query processing among the heterogeneous information sources. The domain model in metadata server (level I) keeps the knowledge within application domain, and the

information source model (level II) describes the semantics of information sources. Therefore, a common data model can be built via the mapping relation between both levels. Associating with the correspondent wrapper, mediator not only handles the semantic queries from user, but also transforms the queries for corresponding wrapper. We also present two algorithms for information source selection and query translation. These algorithms can independently reformulate the query into related sub-queries.

Keyword: heterogeneous information sources, mediator, information in demand, metadata server, wrapper.

一、緒論

晚近由於資料庫與網路技術的快速發展，傳統集中式處理、獨立存在的資料庫管理系統，開始面臨到實際應用上的困難。在資料庫系統方面，資料的儲存方式不再單純，而資料的表現形態也更加的多元化。在網路系統方面，由於網際網路(Internet)的蓬勃發展，讓任何人都可以利用網路來尋找想要的資料；是以在網路環境下資料存取的需求變得更為殷切。但是，分散在各處的資料，包括了傳統的文數字資料、或是多媒體的影音檔案，都各自有其儲存的方式與格式。由於這些資料的格式可能互不相容、甚或彼此衝突，於是在網路上構成一種異質性資訊源(Heterogeneous Information Sources)環境。在這種複雜且多變的環境下，如何有效且快速的對使用者提供整合性的服務，遂成為當下極具學理與實用價值的研究課題。

Wiederhold 氏[8]於一九九二年倡議以調節器

* 本研究計畫由國科會補助，計畫編號：NSC86-2213-E011-023

(mediator)作為資訊系統智慧型整合的一種運作模式，其後陸續有學者在此理論架構下研製各類系統及工具軟體。例如，HERMES系統(Heterogeneous Reasoning and Mediator System)[7]是一個運用調節器架構而植基於混合式知識庫(Hybrid Knowledge Base)理論的系統，用以提供一個整合不同資訊源的工作平台。在HERMES系統中用一種宣告性的語言來定義調節器，並製作一個特殊的編譯器來編譯調節器；其主要貢獻在於倡議以知識工程的推理方法來作領域(Domain)整合及語意整合，並定義了一套相當完整的調節器語言。HERMES雖具有相當的擴展性，但每次新增加資訊源時必需重新編譯調節器，使用上實難以稱便。其次，該系統對不同資訊源之間的資料一致性問題並未作探討，所有的資訊擷取都是使用者被動式的查詢。TSIMMIS(The Stanford-IBM Manager of Multiple Information Sources)[3]是由史丹福大學所發展，該系統提供一組工具軟體以整合異質性的資訊源。其主要的系統元件包括了調節器、轉換器(Translator)、分類器/抽取器(Classifier/Extractor)、制約管理器(Constraint Manager)等。其主要作法是藉由轉換器將異質的資訊源轉換成一個共同的模式，稱為物件交換模式(Object Exchange Model, OEM)，而後調節器加以整合。基本上來說，TSIMMIS是一種物件導向技術的衍生與應用，所以它可以處理結構化與非結構化的資料。但整體而言，TSIMMIS的架構略嫌鬆散，各模組之間耦合性的處理尚有待加強。Garlic[2]系統則是由IBM Almaden研究中心所發展，其目的在於研發一套系統及其相關的工具軟體，用以管理大量且異質性的多媒體資訊。Garlic將各種不同性質的資訊源視成一種單一的綱要，而後透過特殊化的SQL來操作及查詢資料；所以其基本的理論背景與TSIMMIS相似，只是其物件導向的模式主要係架構於ODMG-93的標準。Garlic並未使用調節器，而是採用類似中途軟體(Middleware)的概念，唯其對於制約管理及支援智慧型存取技術方面則未作討論。

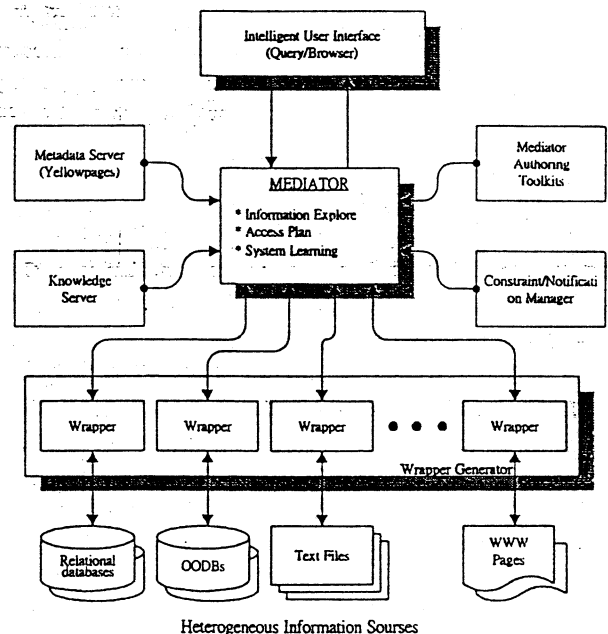
基於上述的背景說明，我們認為在異質性資訊源環境下建構一個主動式且具整合性的系統，是亟具學理與實務價值並符合資訊科技趨勢的一項研究。我們已提出一個稱為InfoHub的模組化整合系統[4]，相較於前述各系統，InfoHub架構的不同之處在於其對於動態的資訊環境的掌控亟有助益，並在異質性資訊源的环境下可提供以下的功能：

1. 主動式：自動作需求判斷提供訊息(智慧型)
2. 整合性：對於異質的、非傳統式資訊源的包容
3. 一致性：系統不需要支援交易監督的制約方式
4. 完整性：能發現未完整定義的資料(隱含性資料)

InfoHub的整體系統架構如圖一，其核心模組包括調節器、包裝器及智慧型的使用者介面。其中調節器是系統運轉的主要引擎；包裝器則是一種轉化器，使得各個資訊源具有共通性的模式(interoperable model)；智慧型使用者介面則是提供使用者擷取資料的瀏覽與查

詢工具，並對使用者的需求，作先期的前處理(Pre-processing)。

本文旨在闡述InfoHub系統中調節器的研製技術與其相關理論的探討。下一節我們說明調節器的系統架構，包括了設計原則及其語意模型。第三節則描述一個二層式超資料伺服器的維護與管理，此超資料伺服器將與調節器作緊密的結合，以協助調節器對查詢的操作，第四節敘述調節器對語意查詢的處理，包含資訊源選取對應機制及語意的剖析解譯，第五節則作結論並探討未來的研究方向。



圖一：InfoHub系統整體架構圖

二、調節器系統架構

2.1 調節器設計原則

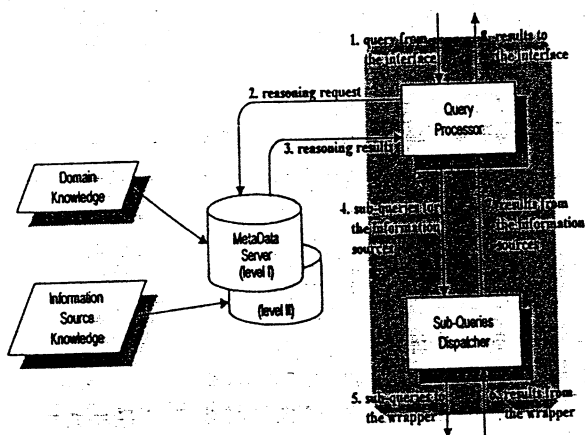
調節器為異質性資訊源系統的核心，功能是整合系統中的領域知識與資訊源系統。調節器保留處理系統查詢時必要的知識，例如系統應用問題的限制與規則、對應資訊源的語意結構等，使系統在處理查詢時能夠找出對應的資訊源，並且分派適當的子查詢給包裝器，以便存取資訊源中的資料，同時根據應用規則對查詢結果進行整合與推論，最後才回應結果給使用者。調節器系統必須要能滿足下列需求：

1. 接受並處理來自使用者介面的查詢需求：調節器中需要一個查詢處理的模組，用來讀入、分解與剖析使用者的查詢，再由剖析後的查詢產生對超資料伺服器的推論需求，並送交超資料伺服器處理。超資料伺服器則藉由資訊源對應用領域的語意轉換模式，將參與的異質資訊源概念上組成一個新的資訊系統。
2. 建立存取計畫(access plan)：此程序需由查詢處理模組與超資料伺服器交互作用完成，需要完成查詢的定義替換(substitute-definition)及資訊

源選取(information source selection)兩項工作。透過這兩項工作可以使針對應用領域模式(domain model)的使用者查詢,轉換為資訊源模式(information source model)的查詢,並交予子查詢派送器(sub-query dispatcher),將子查詢送交給適當的包裝器。

3. 派送子查詢的處理機制: 透過查詢處理的轉換機制與超資料庫系統的推論分析,系統可以產生對應到各個資訊源的子查詢,因此調節器需要具有能夠將子查詢派送到包裝器的能力。此外,調節器還要能產生適當的派送順序,以避免因子查詢間有資料相依(data dependence)的關係而發生錯誤,或產生不正確的查詢結果。
4. 接受並處理查詢結果: 當子查詢完成派送後,調節器需要能夠判斷查詢結果是否與特殊的情形、限制或規則有契合(match)的情形,並且依照判斷的結果執行適當回應,才將結論傳回給使用者。

圖二說明了調節器的基本架構,圖中各模組間所傳遞的資訊同時說明了調節器查詢處理的流程,各項資訊前標明的數字說明了查詢處理的順序關係。由於調節器系統擁有處理特定應用問題的知識,因此系統還可以利用調節器中已經定義好的規則,進行主動式的查詢。主動式查詢並非由使用者透過使用者介面發送,而是由系統自行觸發產生,並根據所得的資料與系統預設規則,做出智慧型的判斷,有別於由使用者發送的被動式查詢。



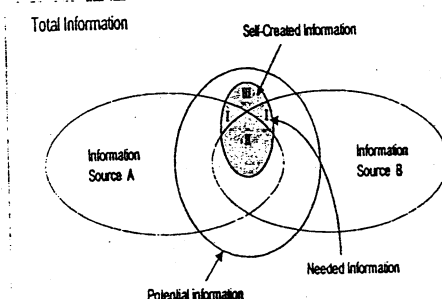
圖二：調節器系統架構圖

2.2 調節器語意模型

2.2.1 資訊需求模型

目前散布在網際網路上的資料極為豐富,資料的存取方式也相當簡易,但是對於資訊應用者來說,由於可供取用或是可能需要使用的資料量太過龐大與分散,所以如何能夠找出符合需要的資訊,並且順利地擷取出實際想要的資料,就成為資訊提供者(information provider)以及資訊接受者(information

receiver)共同關心的課題。現階段普遍解決的方法是利用搜尋引擎(search engine)來幫助使用者找尋資料,但是利用關鍵字(keyword)配合簡單邏輯運算的方式並不容易找出真正符合需要的資料。原因之一是對於關鍵字意義認定的差異問題,另外,由於資料增加的太過迅速,任何以關鍵字詞的搜尋都將會找出過多的資料,然而我們發現大多數的查詢結果都是無用的垃圾資訊。就以目前的網際網路現況為例,查詢單一關鍵字詞後出現上千筆的資料是常有的事,但是滿足需求的卻是屈指可數。



圖三：資訊需求概念模型

我們提出「資訊需求」(Information in Demand)這種折衷於多資料庫系統與分散式系統的概念,以作為異質性資訊源環境知識建立原理的佐證。我們發現,根據某項特別的應用整合數個資訊源時,若每個資訊源僅輸出與應用有關的語意結構,同時系統的應用抽象概念又係根據新的應用所設立,並且透過系統化的建立方法與轉換機制,在資訊源整合的過程中所遇到的阻力會較傳統網要整合要小,同時系統也將更符合使用者的需求。

圖三所描述的是本文倡議的資訊需求概念模型,外框的正方形表示整個系統中所能包含的全部資訊,兩大橢圓形表示資訊源 A、B 本身所包含資訊的集合,中心的圓形表示系統中可能潛在的資訊,有陰影的小橢圓形表示系統應用的範圍中需要處理的資訊。此外,第 I 號區域表示可由整合的資訊源系統單獨提供,第 II 號區域則應由系統從整合的資訊中推論或定義而得,第 III 號區域的資訊係為整合後的系統所設計,而所整合的資訊源中可能並不包含可用的資訊,因此需要由系統建立者自行建立其專業資訊。由圖三我們發現有陰影的小橢圓形部分是使用者的需求(in demand)部分,其它部分的資訊則是多餘的。在需求的資料中,僅有第 II 區域需要由資訊源整合得到,因此在系統整合資訊源的過程中,若能排除不必要的資訊,則整合工作就可以變得較為簡單。

因此,調節器的設計不應以資訊來源的多寡作為系統關切的焦點,取而代之的是以系統提供的資訊內容作為依據。由系統所要解決的問題來設定系統知識的領域,在領域內定義需要處理的問題、判斷時需要的資料、以及可能查詢的項目等問題。再由系統建立

者制定合理的系統知識，並且確立所要選用的資訊源。一般而言，在資訊系統發展生命週期 (system development life cycle) 中[5]，不論先決定範圍再找尋資訊源，或由已知資訊源產生新的設定，只要最終所有領域知識需要的資料的範圍都有適當的資訊源對應，就可以運用調節器的系統架構來運作。

2.2.2 領域模式與資訊源模式

領域模式是用以描述應用問題領域中資訊的類別，以及知識類別間的關係。在進行描述工作時，不必考慮所描述的類別與關係是否都有相對應的資訊源系統資料，而是以使用者處理真實世界工作 (real-world task) 時所需資訊的觀點作為描述的中心。領域模式的建立步驟為：1). 定義領域中的類別、2). 加入類別間的關係、3). 定義類別的關係鍵 (key role)、4). 定義關係以及 5). 加入關係中的其它定義。

在領域模式中可以用以建立應用需要的相關知識，例如「年輕的經理人」(年齡小於四十歲的經理人) 的概念可能對應用十分重要，我們可以定義 Young_Manager 概念如下：

```
(defconcept Young_Manager
  is (and MANAGER
    (< EMPLOYEE_age 40)))
```

領域模式中的應用知識並不需要有實際的資訊源對應，屬於系統應用層次的資訊可以定義在領域模式層中，只要該知識的定義合理，並且能符合系統應用的需要。至於未與實際資訊源直接提供完整定義的知識，可以由系統建立者依據可取得的間接資料，以關聯或推論限制的方式定義。在前述「資訊需求」的概念中，使用者自建知識，如 Young_Manager 概念、資訊源潛在的知識等都是在領域模式層中定義。

我們定義了 CONCEPT_LINK 與 RELATION_LINK 兩個關係，分別表示領域概念與資訊源概念、領域關連與資訊源關連的對應關係，以供系統處理查詢需求時能正確地對應到適當的資訊源。其中 CONCEPT_LINK 關係的定義為：(資訊源概念, 範圍層概念, 包裝器名稱, 欄位指標, 包裝器指標)，而 RELATION_LINK 關係則定義為：(資訊源關係, 範圍層關係, 資訊源實體名稱, 資訊源實體項目, 實體類型)。

三、超資料伺服器

超資料伺服器(metadata server)是一個描述資訊源結構、資料形態與資料關係的知識系統。在異質性資訊源環境中，藉由資訊源對應用領域的語意轉換模式，將參與的異質資訊源概念上組成一個新的資訊系統，超資料伺服器就是描述資訊源中資料、資料關係以及與應用知識對應關聯的知識系統。

3.1 超資料伺服器維護與管理

就已定義的超資料伺服器來說，並不需要特別的維護工作，但是當知識發生變化(新增或刪除知識)、資訊源變化(資訊源的新增或移除)以及合併超資料伺服器時，就需要調整超資料伺服器的知識語意，以符合系統應用的需要。

1. 知識的變化：當發生知識需要新增的情形時，則

與系統建立知識的過程相似，只需要將新增的知識加入已存在的知識庫中，但是必須要避免發生語意衝突。當某些知識需要被刪除時，問題會較新增知識困難許多，通常必須由系統建立者以人工的方式完成：首先必須先由範圍層中刪除對應的知識結構，而所有與其相關的概念關連也都要一併刪除，接下來必須刪除與其對應的資訊源對應關係，但是資訊源階層的知識可以不用刪除。當刪除工作完成後，運用 Loom 知識系統中的 (list-undefined-concepts) 述詞來檢視知識庫中是否存在定義不恰當的知識。由於 Loom 中的知識建立規則要符合分類原則[6]，所有的知識概念與關係都必須有適當的領域分類定義，因此若因為刪除某些知識描述而造成部分知識定義不完全的情形時，可以由知識庫描述工具檢測出錯誤。

2. 資訊源的變化：資訊源的變化有下列兩種情形：

(1) 提供資料的資訊源發生內部與結構變化：當資訊源的結構發生變化時，若領域知識並沒有相關的變化，只需要改變領域與資訊源的對應關係即可。若領域知識有所改變，則必需要刪除原先的對應關係以及相關連的其它知識，以重新加入資訊源的方式處理。

(2) 新增或移除資訊源：當資訊源的新增純粹是伴隨新的應用知識時，則新的應用知識與資訊源語意並不影響舊應用知識與資訊源對應關係，因此不會有太大的影響。但是當新增資訊源要取代舊資訊源時，由於資訊源與領域知識的對應關係必需由資訊源概念與資訊源關係共同組合來描述，所以新資訊源必需要完全替換舊資訊源的對應關係，否則會發生資訊源無法對應的衝突。當資訊源需要移除時，必須將資訊源所對應的領域知識一併移除，同時與被刪除領域知識有關的其它領域知識的關連也要刪除。

3. 超資料伺服器的合併：當兩個已存在的超資料伺服器需要進行整合時，可以使用高階應用的整合方式進行合併。將超資料伺服器所代表的系統視為新的資訊源系統，因此只需要檢視舊系統的知識語意是否滿足新系統的應用，同時既有系統並不受到新系統應用的影響。

3.2 超資料伺服器知識語意的衝突

由於異質性資訊源系統係根據應用問題所設計，同時只整合資訊源中需要的部分資訊，因此相較於多資料庫系統的衝突情形，異質性資訊源系統所面臨的衝突問題較少，但是語意的衝突仍是無法避免的問題。本節討論超資料伺服器知識建立過程中可能面臨的語意衝突問題，並提供原則性的解決方式。

3.2.1 不確定性衝突：

1. 同義異名：不會造成知識描述的困難，相同語意的知識物件可以連接至一個具有共同定義的關係或概念。在超資料伺服器中，知識物件必需滿足分類原則，因此分散在不同概念的關係在應用上蘊含不同的意義，透過共同的概念，可以清楚的描述同義異名知識物件的性質。

- 異義同名：在單一的超資料伺服器中並不會發生，因為系統中並不允許同名物件定義於相同概念結構中，但是可能發生於多知識庫的萃取整合過程中，同時需要在建立整合應用前解決。當有衝突發生時，解決的方式有：整合者在建立時避免定義可能衝突的問題，但此法並不理想。

3.2.2 知識意義的衝突：

- 與時間、資料更新有關的衝突：此部分係知識庫使用後所產生的衝突，系統建立者可以針對可能的衝突設計偵測或校正機制，以自動化的方式來檢測知識結構與資訊源的衝突。
- 知識本身的矛盾衝突：需要由專家認知，並由系統建立者的測試才能解決。

四、包裝器設計

包裝器之功能主要係將資訊源邏輯性地轉換成共同模式[10]。在執行的過程中包裝器要先將查詢需求轉換成資訊源可執行的陳述，而後再將執行結果轉成共同模式；我們研擬的包裝器系統架構如圖四。以下逐一說明系統中各元件的功能及採行的技術與理念。

● 查詢處理模組(Query Processor)：

包裝器的主要工作就是將調節器傳來的 Loom 查詢需求轉換成資訊源特定的查詢介面或語言，而資訊源查詢介面則由系統定義的資訊源描述語言來描述說明，使查詢處理模組能做正確的轉換。此外，因為我們系統是以 Loom 為內部查詢語言，所以模組內包含了針對 Loom 查詢語言的剖析器(Parser)，而剖析器產生一資料結構儲存查詢語意，並交由各資訊源查詢轉換器(Query Converter)轉換至目的語言或介面，轉換後之查詢字串再交由連接管理模組傳送至目的資訊源執行之。

● 連接管理模組(Connection Manager)：

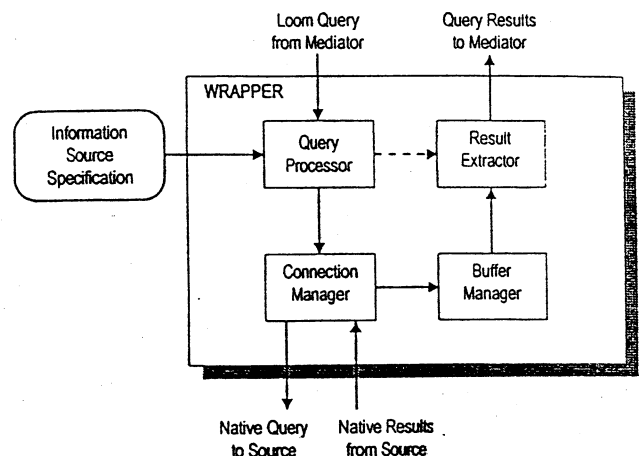
接收由查詢處理模組轉換後的資訊源查詢字串，並透過 Internet 或 Intranet 傳送至遠端之資訊源，且接收資訊源處理後之查詢結果。除此之外，此模組還須處理與資訊源連結及錯誤偵測之問題。在實作方面，目前市面上有許多資料庫之連結工具可供搭配選用，如：ODBC(Open DataBase Connectivity)、JDBC(Java DataBase Connectivity)提供與關聯式資料庫之連結、具網路功能的程式語言如 Java--能與 WWW 伺服器連接之程式庫，以及各資料庫廠商所提供的前端連接資料庫之 API 等等。因此實作上可以採用這些工具，以迅速地達成連接的目的。

● 緩衝管理模組(Buffer Manager)：

接收由連接管理模組傳回之資訊源查詢結果，並作暫時之儲存。另一方面，最後的查詢結果可能需要將前幾次的查詢結果作合併的動作，因此也必須管理先前的查詢結果。

● 結果萃取模組(Result Extractor)：

因查詢結果之表示法及格式可能是相當複雜的結構，所以須先將查詢結果萃取出有用的資訊，再進一步濾出調節器想要的屬性與相關訊息。此外，因為萃取後的結果可能為一複雜物件，所以我們的結果表示式將採用類似 Lisp 語言的串列表示，以表現階層式的物件結構。



圖四：包裝器系統架構

五、語意查詢處理

我們採用 Loom [6] 的 Retrieve 巨集指令查詢語言標準，並且以此作為系統中不同子系統之共同查詢表示模式。在系統中 Retrieve 將會以兩種格式出現，分別是調節器系統(包括查詢介面轉換)使用的 Retrieve 敘述，以及在資訊源包裝器中的 Retrieve 敘述。由於異質性多資訊源環境中包含了應用問題的知識，因此 InfoHub 系統中的調節並不是一個單純的資訊查詢系統，它不但能夠擷取被整合資訊源中的資料，同時也能夠處理與應用領域有關的應用問題。在實際運作上，使用者介面將使用者的查詢需求轉換為調節器內部的 Loom 查詢形式，同時這些查詢子句的條條描述對象是超資料伺服器領域階層中的知識。因此，調節器內部的推理器 (Loom-Query Reasoner) 將會對 Loom 查詢執行剖析與解譯的工作，並將 Loom 查詢轉換為對等的資訊源子查詢，並傳給查詢分解器 (Query Decomposer)。查詢分解器將分解包含資訊源子查詢的整合查詢，並且將配組後的子查詢傳遞給派送器 (Dispatcher)，由派送器將子查詢送交至包裝器。由包裝器所傳回的查詢結果會由結果組合器 (Result Assembler)，由結果組合器產生結果或結論。

5.1 查詢類型

在 InfoHub 系統中我們分析異質性資訊源環境下可能出現三種查詢類型[9]，並各有其處理方式：

- 自動執行系統已定義的查詢：這類的查詢主要是由系統建立者所定義的查詢，並且可以設定查詢的執行時間，使其自動地執行。除此之外，這些已經定義的查詢還可以配合知識規則的比對，檢查是否符合特定的條件，做更進一步的利用，特別適合如核能控制、環境監測或金融

決策系統等應用。

2. 由使用者執行已定義的查詢：系統建立者可為使用者建立應用問題的查詢，透過使用者介面填入適當的資料，使用者就可以得到查詢的結果或結論。這類查詢與前述的查詢類似，惟查詢的執行時間並非由系統定義，而且查詢條件可由使用者依據不同需求而設定不同的查詢條件，與自動執行的固定查詢條件不同。
3. 由使用者執行未定義的查詢：此類的查詢是系統建立者所未定義的查詢，可能的輸入方式是由使用者自由輸入，或者是由智慧型的使用介面根據需求所產生。由於系統中並未定義，而此類查詢很可能是表達系統中潛在的語意，因此必須先檢視查詢的條件是否滿足知識庫中的知識語意，並且要能記錄該條件在知識結構中的相關位置。

5.2 資訊源選取對應機制

在資訊源語意知識建立的過程中，系統建立者必須描述資訊源所對應的領域知識關係，其中 Concept_Link 是描述領域類別與包裝器的對應關係，Relation_Link 則是描述領域關係與資訊源中資料欄位的對應關係。

調節器所接受處理的是針對領域知識的查詢，因此不論是語意定義替換、語意一般化或語意特殊化的語意轉換機制，都是針對領域知識進行轉換與合併，因為查詢的敘述係針對領域知識而設定，所以關於查詢的語意轉換也必須在領域模式的定義內處理才合理。此處所討論的資訊源選取對應機制與應用領域範圍的定義無關，只是將語意轉換後的查詢條件轉換為對資訊源的高階查詢，因此對應轉換並不會發生語意轉換漏失的問題。

第一種情形是資訊源本身即為實體資料來源，因此領域知識所對應的資訊源知識即足以提供包裝器進行查詢，只要對應到該包裝器並賦予查詢條件即可。第二種情形是資訊源本身並不是實體資料來源，資訊源只是某些實體資料源的代表。因此，不論是領域知識或資訊源知識，都必須增加與應用知識無關的關係來描述。兩種資訊源層次結構除了在領域與資訊源知識描述不同外，在定義領域/資訊源對應關係時也不相同，因此，我們設計了資訊源選取對應演算法，為兩種轉換情形分別制定選取對應策略，演算法的內容如下：

【演算法 4.1】資訊源選取對應演算法：

【輸入】 domain_query (語意轉換後之領域查詢)

【輸出】 information_source_query (對應之資訊源查詢)

【條件及方法】

1. 設限制子句的排列在每一組 concept 間只對應一個 wrapper。
2. 子句的形式設定為

```
<concept-exp> ::= (<concept-name> <variable>)
<relation-exp> ::= (<relation-name> <bound-variable> <term>)
<assignment-exp> ::= (= <unbound-variable> <arith-
```

```
exp>)
```

```
<comparision-exp> ::= <member-comparison> |
<arithmetic-comparison>
```

3. Concept_link GetCONCEPT_LINK(Concept-name);
/* GetCONCEPT_LINK 將由超資料伺服器中找出 Concept-name 的對應定義，並傳回 CONCEPT_LINK 的串列 */

4. GetRELATION_LINK(Relation-name);
/* GetRELATION_LINK 將由超資料伺服器中找出 Relation-name 的對應定義，並傳回 RELATION_LINK 的串列 */

5.3 查詢語意剖析與解釋

根據 Loom 知識結構的特性，調節器中使用語意定義替換 (substitute-definition)、語意一般化 (generalization) 與語意特殊化 (specialization) 三項語意轉換機制[1]，作為調節器知識語意轉換的基礎操作模式，以保持 Loom 領域查詢與資訊源查詢的語意正確性。

我們提出一個查詢語意剖析與解釋的演算法，將使用者介面交付給調節器的查詢，依照語意處理的各項轉換機制，轉換為對應於資訊源包裝器的資訊源查詢。由 Loom 的 Query 語法，我們得知查詢的語意集中查詢的限制子句，因此查詢的語意剖析與解釋主要是處理查詢中的限制子句，

【演算法 4.2】查詢語意剖析解釋演算法：

【輸入】 domain_query (語意轉換前的領域模式查詢)

【輸出】 information_source_query (轉換後的查詢)

【方法】

1. Decompose domain_query into tokens.
2. Set retrieving variables as query variables.
3. Execute information source selection process.
while (not all clauses map to information source)
{Execute substitute-definition reformulation.
Execute generalize and specialize reformulation.
Move parsing cursor to head of clause-list.
Execute information source selection process.}
while (not all clauses done)
if(<concept-
create new sub_query variable table
until (next <concept-exp>){
put unbound variable into variable table;
next clause;}
next clause;
4. execute sub_query depart process
5. execute sub_query dependency process
6. execute sub_query dispatch

在前述的語意轉換過程中，語意轉換機制會將四種子句依照語意而改變子句中的符號名稱(演算法步驟3中的第一個迴圈)，因此演算法步驟3的第二個迴圈是處理各資訊源的限制子句，主要目的是處理原始領域查詢的相關變數，以供子查詢分割與派送時參考使用。演算法步驟3的第一個迴圈會使全部的限制子句都完成資訊源選取，第二迴圈為子查詢處理前的準

備工作。步驟 4-6 是子查詢的處理程序，針對剖析與解釋處理後的查詢，進行子查詢分割、子查詢相依與子查詢派送的處理，以送交資訊源包裝器處理。

六、結論與未來研究方向

本文提出調節器的資訊源整合原理及查詢處理機制，以使異質性多資訊源系統能整合應用知識、資訊源資訊語意，並以整合知識處理使用者的查詢需求。我們以兩階層式超資料伺服器維護異質資訊源的整合語意，並以邏輯式知識描述語言作為知識語意的描述工具，因此調節器可因邏輯描述豐富的語意變化使系統更具應用彈性，也可以語意描述的方式整合非結構化資訊源，並利用邏輯陳述的事實對應用問題進行推論，產生比資料庫系統更符合使用者需求的資訊。應用領域階層知識可使系統建立者專注於系統應用問題的需求，同時也方便了系統使用者對資訊取用的了解。採用資訊源語意對應領域知識的做法，可使資訊源整合的過程更加方便，當資訊源有所更動時也不需要改變領域知識。

我們倡議運用資訊需求概念以縮小知識整合的範圍，相較於多資料庫系統完整網要整合的做法，以調節器為核心的作法其整合的過程較為容易，且更具實用價值。調節器配合使用者自建知識，可以方便地利用整合資訊源的資料，產生更適合使用者需要的資訊。我們認為未來可繼續發展研究的項目有：

1. 調節器知識建構工具 (Mediator Authoring Tool Kits)：未來可研製調節器知識建構工具，以幫助系統建立者迅速地建立應用系統的知識語意，並以系統化的方式整合相關資訊源系統。
2. 實體化 (Materialization) 調節器的查詢結果，以建構具有整合性效果的資料倉儲 (Data warehouse)。藉由適當的資料品管過程，我們可以得到針對特定領域的整合性查詢結果，再加以轉換格式後，即可匯成有資訊化 (informative) 效果的資料倉儲。其次，可以再結合資料採掘 (Data mining) 的技術，嘗試在此資料倉儲中發掘更深層的知識，如此則可構成一個更完整的整體應用環境。
3. 模糊語意及加權推論：應用知識中存有許多不確定性、或語意模糊的問題，因此未來可研究在調節器中加入模糊語意的處理，並可為知識加入應用權數，依照資料的變化做智慧型的處理。
4. 語意查詢最佳化：本文所述的查詢處理機制可以正確的處理查詢，但是不論因為知識語意結構或資料擷取的成本問題，最後產生的子查詢並不一定是「最佳的」，因此未來可研究查詢最佳化的問題，使調節器能更有效率的處理查詢需求。

參考文獻

- [1]. Arens, Y., Y. C. Chin, C. A. Hsu, and C. A. Knoblock, "Retrieving and Integrating Data from Multiple Information Sources," *International Journal on Intelligent and Cooperative Information*

Systems, Vol. 2, No. 2, pp. 127-158, 1993.

- [2]. Calvance, D. M. et al, "Towards heterogeneous multimedia information system : The Garlic approach," Research Report, IBM Almaden Research Center, Nov. 1994.
- [3]. Chawathe, S. et al, "The TSIMMIS project : integration of heterogeneous information sources," <http://www-db.stanford.edu/tsimmis/>, 1995.
- [4]. Chu, Y. C., C. C. Lien, and C. C. Yang, "InfoHub: A Flexible System for Retrieving and Integrating Heterogeneous Information Sources," *Proceedings of 1997 Workshop on Distributed System Technologies and Applications*, pp. 597-602, 1997.
- [5]. Kendall, P. A., "Introduction to System Analysis and Design: A Structured Approach," 2nd Edition Wm. C. Brown Publishes 1992.
- [6]. MacGregor, R. M. "Representing Reified Relations in Loom," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 5, pp. 179-183, 1993.
- [7]. Subrahmanian, V. S. et al, "HERMES : A heterogeneous reasoning and mediator system," <http://www.cs.umd.edu/projects/hermes/>, 1995.
- [8]. Wiederhold, G. "Mediators in the architecture of future information systems," *IEEE Computer*, pp. 38-49, March, 1992.
- [9]. 嚴國瑞, "異質性資訊源環境中調節器之研製," 國立台灣工業技術學院碩士論文, 86 年 6 月.
- [10]. 江明洲, "異質性資訊源環境中包裝器之研製," 國立台灣工業技術學院碩士論文, 86 年 6 月.