

語意式資料倉儲系統之架構研究

黃士銘 周棟祥

中正大學資訊管理研究所

摘要

近來，資料倉儲(Data Warehousing)已經從一種學術性的理論研究，漸次衍生成為一種廣為大家所能接受的應用技術。資料倉儲內部經由整合、運算後的資料以資料晶圓(Data Cube) 的結構資料型態儲存，並提供給使用者一個多維度(Multi-Dimensional)的資料分析及查詢。但是由於目前的資料倉儲系統建構資料量過大，造成資料查詢不易，往往無法即時回應需求；另一方面，每個資料晶圓都是由資料彙整而成的獨立個體，缺乏一個完整的語意來表達資料晶圓彼此間的關係，使得決策者無法做出更深入的判斷與分析。因此我們認為想要解決資料倉儲內資料晶圓的資料重覆性與資訊獨立的問題，應該要對資料倉儲內的資料晶圓建立一個具有語意關係的資料晶圓模型(Semantic Cube Model)，此模式將具有資料語意模式上的繼承、分類、聚合等語意關係，不但可以解決目前各個資料晶圓之間資訊獨立與資料重覆等情況，並且可藉由資料晶圓間彼此關係的建立，減少資料倉儲的資料量，提供更豐富的資料查詢能力，進而提升整體資料倉儲系統的效能。為了實際驗證本研究的假設，在本文的最後，我們以個案方式來說明整個架構與建置方法的可行性，再利用系統雛形與實驗設計法計算資料的縮減量，以驗證其有效性，並藉由實地的應用與訪談來評估其成效。

關鍵字：資料倉儲、資料晶圓、物件導向技術、語意關係模式

The Research of Semantic Data Warehouse System

Shi-Ming Huang Tung-Hsiang Chou
Department of Information Management
National Chung Cheng University

Abstract

Nowadays, data warehouse systems are become a popular technology in the enterprise. Because of network bandwidth and communication is not sophisticated enough for practical usage. Thus, a lot of enterprises have focus on building data mart with materialize view. But most of these data warehouse systems have some limitations in terms of flexibility, efficiency and scalability. In particular, the sizes of these data warehouses are constantly growing and will soon become overloaded with data that led to difficulties in data maintenance and analysis. GartnerGroup estimates that by 2004, enterprises will be faced with 30 times more data than 1999 levels. In this paper, we focus on integrating data information between the data cubes, which can solve the problem of duplication and independent information of data cube. We present a semantic cube model (SCM), which extends

object-oriented technology for data warehouses and allows users to design the abstract semantic relationships between different cubes, thus improving the performance of query integrity and reducing data duplication.

Keywords: Data Warehouse, Data Cube, Object-Oriented Technology, Semantic Cube Model

1. 緒論

近十年間，資料倉儲(Data Warehousing)已經從一種學術性的理論研究，漸次衍生成為一種廣為大家所能接受的應用技術。它主要是針對決策支援系統的需求所發展出來的新一代資料庫技術，此技術具有整合資料庫的特性，能將來自於不同系統來源的大量營運資料作一適當的組合與彙整，以方便使用者以不同的角度，查詢出符合自己所需要的資訊，並協助其作正確的決策。簡單地說，資料倉儲就是將企業內寶貴的營運資料彙總整理，使其成為協助主管做出各種決策的整合性「智庫」。利用這個「智庫」，企業可以更靈活、深入地分析所有需要的資料，進而達到提昇競爭力的效果。

由於目前的網路通訊狀況的限制，無法克服資料查詢時，系統回應時間太長的問題，所以大部分的資料倉儲系統建構方式皆採用實質視界技術，並已有許多相關性研究在探討要如何減少在實質視界技術上的資料重覆性的問題與如何保持和遠端資料一致性的問題(Abadi 1997；Indulska 1999)。另外，為了解決實質視界資料倉儲資料量過大造成查詢資料不易的問題，因而就有所謂的資料市集(Data Mart)的觀念出現，像微軟的 SQL Server、甲骨文的 OLAP Server、Sybase 的 IQ 與 Business Object 等資料倉儲系統，皆是採用此種觀念，將經過整合、運算後的資料以資料晶圓(Data Cube) (Devoney 1997；Gardner 1998) 的結構資料型態儲存。

資料晶圓提供給使用者一個多維度(Multi-Dimensional)的資料分析及查詢的結構；但這種方式將會造成了每個資料晶圓都是由資料彙整而成的獨立個體，缺乏一個完整的語意來表達資料晶圓彼此間的關係，如此更造成了資訊獨立，而使用者也只能抓取部分的知識，從單一的構面去分析資訊，無法作一個全方位的分析，圖 1 顯示目前一般的資料倉儲系統的架構。

一般的資料倉儲架構不僅造成資料獨立問題，也會造成資料晶圓間的資料重覆問題，例如某一公司部門利用資料晶圓 A 分析公司所有的銷售資訊，其包括時間、產品名稱、公司等維度資訊，而後如果其部門經理又只想知道 99 年公司所銷售的某單一產品(如圖 2 中 B 類的 B1 與 B2)之銷售資訊時，此時可發現該產品的部分銷售資料已經重覆出現在資料晶圓 A 中，但以目前的資料晶圓模式

並無法由該資料晶圓 A 將所要的資訊直接萃取出來，而需另行建立一個新的資料晶圓 B 來做出該使用者所要的資訊，因此容易產生資料晶圓間的資料重覆性問題。

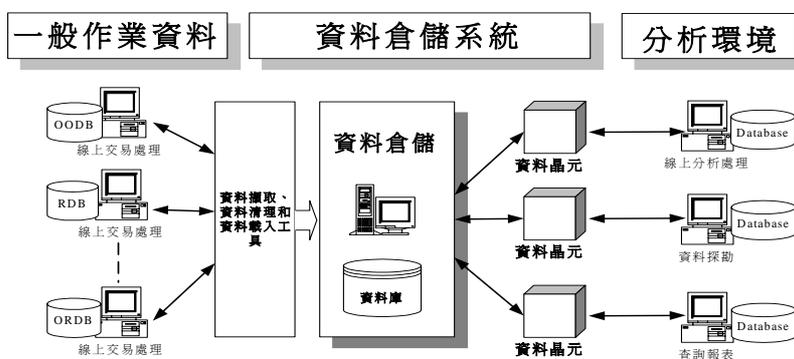


圖 1：一般的資料倉儲架構

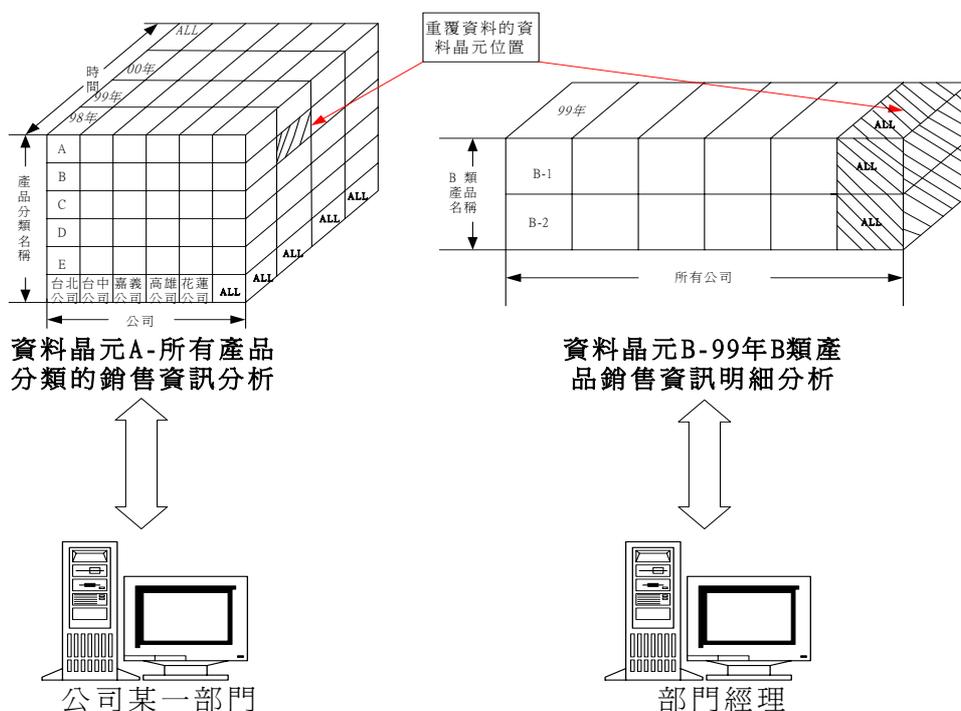


圖 2：資料晶圓間資料重覆的問題

目前已有許多相關之研究認為在資料倉儲系統中，對於資料晶圓的維護動作是相當重要的(Hurtado 1999；Riedewald 2000)，而這些研究的重點皆強調如何維護資料晶圓本身資料一致性與資料量減少的問題，但鮮有針對如何在資料晶圓間建立一個完整語意關係之相關研究。自從關聯式資料模式被提出後，便有許多資料模式研究人員發現了關聯式資料模式的不足，因此就有「語意資料模式」的發展，來解決在關聯式資料上的問題(Fong 1995)。語意資料模式除包含一般資料模式的特點外，另外強調繼承、分類、聚合等自然界中所含有的語意關係，因此

其能表現出複雜的資料型態與資料間關係。

因此我們認為想要解決資料倉儲內資料晶圓的資料重覆性與資訊獨立的問題，應該要對資料倉儲內的資料晶圓建立一個具有語意關係的資料晶圓模型 (Semantic Cube Model)，此模式將具有資料語意模式上的繼承、分類、聚合等語意關係，不但可以解決目前各個資料晶圓之間資訊獨立與資料重覆等情況，並且可藉由資料晶圓間彼此關係的建立，減少資料倉儲的資料量，提供更豐富的資料查詢能力，進而提升整體資料倉儲系統的效能。

2. 相關研究

2.1. 資料倉儲系統的介紹

資料倉儲可說是一種多元化的資料庫，它可從不同種類的資料庫中讀取並結合資料，進而形成完整的資訊來回應使用者多元化的查詢。根據 *William Inmon* 對資料倉儲下了定義：「資料倉儲是一種用於協助管理上的需要，將資料重新組織，使它具有主題導向的、整合性的、非揮發性的及隨時間變動的資料搜集技術 (William 2000；Inmon 1997)。」由此可知，資料倉儲的目的在於提供管理者作決策支援，而有別於傳統的資料庫僅用於提供組織完整的交易記錄與資訊；另外，資料倉儲也可從數種不同的來源去搜集資料，並將所搜集來的資料儲存在一快速、方便的機制中，讓使用者能夠更容易、更快速、更有彈性地取得資訊。

若把資料倉儲看作是一個企業，那麼資料市集(Data Mart)(William 2000；Inmon 1997)便可以當作是該企業中的一個部門。而不同的資料市集負責提供資料倉儲中不同需求的資訊，使用資料市集來分散資料倉儲的方式，不但可以降低開發資料倉儲的時間和成本，同時也能提高資料倉儲的效能和安全性。現今的資料倉儲系統如 MS-SQL OLAP Server、ORACLE OLAP Server、SYBASE Meta-CUBE 均提供一新的結構稱為資料晶圓(Data Cube)，讓設計者能以此作為建立資料市集的基礎。近來，有很多研究與文獻均在探討如何設計資料倉儲系統 (Gopalkrishnam.1998；Wu 2001)或者是如何加強資料倉儲的效能(Chen 1999；Miller 1998)以及討論資料晶圓的建立方法與查詢效能，但對於討論資料晶圓間的關係研究似乎較少(Hurtado 1999；Riedewald 2000)；這種狀況類似早期的資料庫研究，在經過一段時間後才發現建立資料間彼此的關係是如此的重要(即所謂的 Data Modeling)，因而我們的研究將朝向探討資料晶圓間的關係來發展，並希望藉此達到增加資料倉儲系統的效能。以下我們將說明目前物件導向資料倉儲之各項研究，並將其分門別類：

物件導向資料倉儲之相關文獻研究	
研究文獻類別	文獻特色與重點
架構設計與轉換 (Abbadi 1997 ; Chen 1999 ; Abello 2002 ; Bensberg 2003)	<ul style="list-style-type: none"> ★利用物件導向軟體工程探討資料倉儲之架構 ★定義資料倉儲中物件化的維度資料模型 ★探討如何在使用物件導向架構的資料倉儲系統中存放資料 ★介紹資料倉儲中的中介資料與物件導向資料模式之對應問題 ★適用於物件導向線上分析處理的架構 ★利用物件導向方法建立多維度資料庫概念模型 ★物件關連式資料倉儲概念層設計之研究 ★將資料綱要轉換成爲物件導向資料倉儲之綱要
效能與資料問題 (Miller 1998 ; Riedewald 2000 ; Bellahsene 2002)	<ul style="list-style-type: none"> ★介紹在物件導向資料倉儲的非壓縮資料模型，並提出其三個漸進式維護法 ★利用物件導向設計方式來增加實質視界資料維護的效率 ★從異質分散式資料探討物件導向資料倉儲對資訊擴散的問題 ★討論在物件導向資料倉儲中，資料的查詢效能
資料晶元 (Gray 1997 ; Hurtado 1999 ; Riedewald 2000 ; Papadias 2002)	<ul style="list-style-type: none"> ★探討單一資料晶元內，維度對晶元的影響 ★建立資料晶元的方式與查詢 ★資料晶元的效能問題 ★資料晶元內資料的維護與更新問題 ★在資料晶元上快速回應查詢效能評估 ★建立一個物件化的資料晶元來做資料探勘

2.2. 資料倉儲系統的規劃與建置

在目前資料倉儲系統的建置上，通常是採用星狀綱要與雪片狀綱要兩種資料模式，這些方法均是採用「由上往下」的設計方式，目的在使資訊的呈現能由最上方的綱要資料向下延伸至細部的資訊，以便讓使用者可以容易地轉換多種維度的資料分析方式，現將其架構與意義分述於後：

1. 星狀綱要(Star Schema)

星狀綱要資料結構是目前最常被用來當作設計資料倉儲系統時的架構(William 2000)，它也是一個具有高度最佳化的資料結構，亦可稱爲維度模式架構(Dimensional Model)(Inmon 1997)。基本的星狀綱要都是由一個資料表格爲中心點稱爲事實表格(Fact Table)，例如圖 3 中的銷售量事實表格，以此向外延伸出去；而其外圍是由幾個外來鍵所參照的維度表格(Dimension Table)，例如圖 3 中

產品銷售星狀綱要中有時間維度表格、產品維度表格與地區維度表格等資訊。

2. 雪片狀綱要(Snowflake Schema)

當星狀綱要中的維度表格資料內容過多時，某些維度就需進行正規化的動作，將原本複雜的維度表格，分成兩個以上的維度表格，並以外來鍵的方式串連在一起，而原本的星狀綱要結構也將因此改變，使原先的星狀綱要更向外延伸出去，此時的資料綱要稱為雪片狀綱要(William 2000)。例如圖 4 中的產品銷售雪片狀綱要內，其中的產品維度表格的類別資訊，就向外延伸成爲另一個維度表格叫產品類別維度表格，提供了產品維度更詳細的資訊。

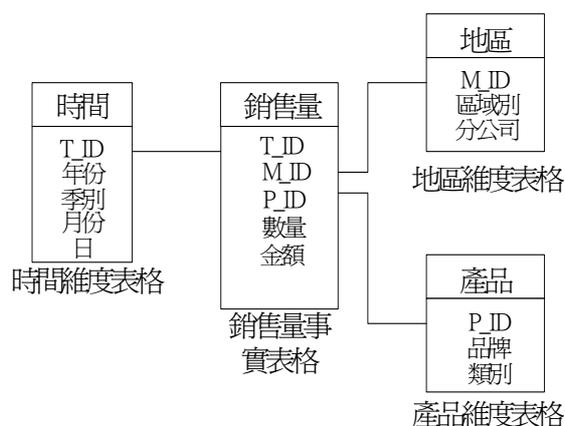


圖 3：產品銷售星狀綱要

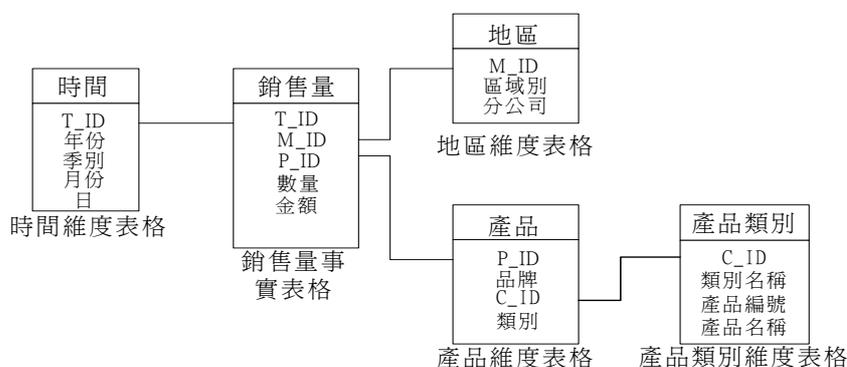


圖 4：產品銷售雪片狀綱要

2.3. 資料晶圓上的資料呈現方式

資料晶圓的產生主要是依據所選擇的維度及測量值的資訊，來產生所需要建立資料晶圓的 SQL 敘述。假設我們選擇了 N 個維度及 M 個測量值，則我們需要產生 $2^N - 1$ 個 Union 指令的 SQL 敘述，再配合 SQL 敘述中所需要選擇(Select)指令的項目，便可以產生建立資料晶圓所需要的 SQL 敘述(Gray 1997；黃士銘

2002), 其演算法如表 1, 在表 2 是國內某家汽車銷售資料, 表 3 為產生 SQL 指令的程序式, 表 4 即是利用表 3 執行後所得到之結果。

表 1：資料晶圓演算法

```

Select  { S(0) }, { F(M1, M2, ....., MM-1) }
From  Data_Base
Union
Select  { S(1) }, { F(M1, M2, ....., MM-1) }
From  Data_Base
Group By  D1
Union
Select  { S(2) }, { F(M1, M2, ....., MM-1) }
From  Data_Base
Group By  D1, D2
Union
:
:
:
:
Union
Select  { S(2N-1) }, { F(M1, M2, ....., MM-1) }
From  Data_Base
Group By  D1, D2, D3, ....., DN-1
    
```

表 2：汽車銷售資料表

類型	日期		銷售量
	年份	月份	
汽油引擎	2000	1	100
	2000	2	200
柴油引擎	2000	1	10
	2000	2	20

表 3：建立汽車銷售事實表格敘述

```

Select  類別,年份,月份,Sum(銷售量)
From  Car_Sales
Group By  類別,年份,月份
Union
Select  類別,年份,ALL,Sum(銷售量)
From  Car_Sales
Group By  類別,年份
Union
Select  類別,ALL,月份,Sum(銷售量)
From  Car_Sales
Group By  類別,月份
Union
Select  類別,ALL,ALL,Sum(銷售量)
From  Car_Sales
Group By  類別
Union
Select  ALL,年份,月份,Sum(銷售量)
From  Car_Sales
Group By  年份,月份
Union
Select  ALL,年份,ALL,Sum(銷售量)
    
```

```

From Car_Sales
Group By 年份
Union
Select ALL,ALL,月份,Sum(銷售量)
From Car_Sales
Group By 月份
Union
Select ALL,ALL,ALL,Sum(銷售量)
From Car_Sales

```

表 4：汽車銷售事實表格

類型	日期		銷售量
	年份	月份	
汽油引擎	2000	1	100
	2000	2	200
	2000	ALL	300
	ALL	1	100
	ALL	2	200
	ALL	ALL	300
柴油引擎	2000	1	10
	2000	2	20
	2000	ALL	30
	ALL	1	10
	ALL	2	20
	ALL	ALL	30
ALL	2000	1	110
ALL	2000	2	220
ALL	2000	ALL	330
ALL	ALL	1	110
ALL	ALL	2	220
ALL	ALL	ALL	330

2.4. 資料晶圓的運作

資料晶圓允許使用者可根據不同的主題和角度來觀看資料晶圓中的資料，每一個資料晶圓除了擁有各式各樣的維度資訊外，有些維度則又具有層級的關係，而每一個維度代表活動事實的一種視野或觀點。因此每個資料晶圓就擁有自己本身的特性，而我們也可以對這一些資料晶圓作各種角度的查詢與分析，常見的功能有上捲(Roll-Up)、下展(Drill-Down)、切片(Slice)、轉換(Dice)等(黃士銘 2002)。

3. 語意資料晶圓模型

在這個章節本研究將提出一個在資料倉儲系統中的資料語意關係模式，稱為語意資料晶圓模型(Semantic Cube Model)如圖 5。我們的語意晶圓模式擴充原星狀綱要的資料模式，利用物件導向的特性，讓此模式不僅可表示出原星狀綱要的資訊，以產生一般的資料晶圓並且允許使用者在資料倉儲中的不同資料晶圓上

建立其語意關係，其包括繼承、聚合及分類等。利用這個特性來解決目前資料晶圓資訊獨立與資料重覆性的問題。

語意晶圓模型
<pre> Cube cube_name [IS_A (Cube, Cube...)] { Dimension {Attribute SYSTEM_TYPE [A_PART_OF (Cube)] [A_MEMBER_OF (Cube, Cube,...)] } [Level {New_Dimension (Attribute (Attribute...))}] Measurement {(aggregation function, attribute) } [Constraint {Condition}] } </pre>
語意資料晶圓的符號意義註解
<p>a. []：在中括弧內的指令最多只產生一次或者不產生</p> <p>b. {}：在大括弧內的指令最少出現一次以上</p> <p>c. “ ”：表示‘OR’的運算子</p> <p>d. IS_A (Cube)：定義 CUBE 間繼承的關係，多重繼承則用“，”將 CUBE 隔開</p> <p>e. A_PART_OF (Cube)：定義 CUBE 間聚合的關係，表示此 Attribute 是由所指定的 Cube 所聚合起來的</p> <p>f. A_MEMBER_OF (Cube)：定義 CUBE 間分類的關係，表示此 Attribute 的內容是屬於所指定的 Cube 的那一類型</p> <p>g. SYSTEM_TYPE：系統初始設定之資料型態(如 CHAR、NUMBER...)</p> <p>h. Constraint: 限制資料晶圓的條件，並結合 Condition 用來說明資料晶圓的條件關係</p> <p>i. Dimension：定義此資料晶圓的維度屬性，Dimension 的名稱為 Attribute</p> <p>j. Attribute：設定維度屬性名稱與資料類型</p> <p>k. Level：定義 Dimension 內各維度的階層關係，以()表示階層的關係，外部稱為父階層，內部為子階層，階層的最上層允許使用者自行定義名稱，以表示為此階層的名稱，階層名稱亦可被當成為維度。</p> <p>l. Measurement：定義資料倉儲內資料晶圓的使用函數與測量值的屬性。</p>

圖 5：語意資料晶圓模型

語意資料晶圓的建立
<pre> Cube Car { Dimension { kind CHAR year NUMBER month NUMBER } Level {date (year(month))} Measurement{(SUM, sales)} } </pre>

圖 6：語意資料晶圓模式範例

現舉一汽車銷售的例子來說明本研究之語意資料晶圓模型如何建立一般的資料晶圓。假設原資料庫上有類型(KIND)、銷售的年(YEAR)月(MONTH)、與銷

售量等欄位，利用此語意資料晶圓模型，我們可定義出一晶圓(Car)，此晶圓可依類型與日期將銷售資料聚合(如圖 6)。

我們所提出的語意晶圓模型不但可表示出一般晶圓的星狀結構，更可建構晶圓間相互的關係，包括『繼承』、『聚合』、『分類』。

3.1. 資料晶圓的繼承關係

繼承一般稱為 IS-A，其是指從已定義好的類別(可稱為父類別)中，去定義出另一個新的類別(稱為子類別)，而這個新的類別除了擁有新定義的屬性與方法外，還包括所有父類別的資訊。

舉例來說，使用者從一汽車銷售資料庫中，建立一名為「汽車銷售」的資料晶圓，其星狀綱要如圖 7，根據這個傳統星狀綱要可以得到一個事實表格資料，如表 5。但是今天如果有另一個使用者要建立一個與「汽車銷售」有關的「電動車銷售」資料晶圓時，此時就會產生一個資料上重覆的問題。假如使用者想建立一個關於「電動車銷售」的資料晶圓，其傳統星狀綱要如圖 8，根據此星狀綱要亦可以得到一個事實表格，如表 6。

要解決此問題，可利用我們的語意晶圓模式的繼承關係，這種方式不但可以減少資料運算時間，同時也可降低資料的儲存空間；根據繼承定義，本研究使用「繼承」的觀念與語意晶圓模式來實作汽車與電動車(*Electronic_Car*)的銷售資料晶圓，如圖 9。

而根據圖 9 的電動車資料晶圓語意模型，可以發現電動車資料晶圓透過型態的關係(即 `Car.kind='Electronic Engine'`)，來繼承汽車資料晶圓的語意模型，而這樣的繼承關係可以圖 10 來表示，如此一來將使得電動車銷售資料晶圓去繼承部分在汽車銷售資料晶圓的事實表格資料(如表 5 與表 6 的灰色部分)。

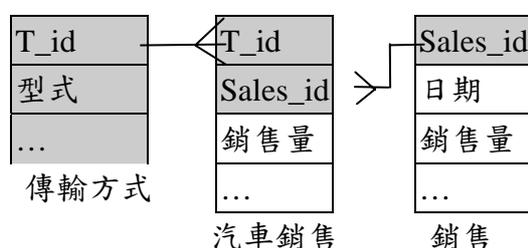


圖 7：汽車銷售的星狀綱要

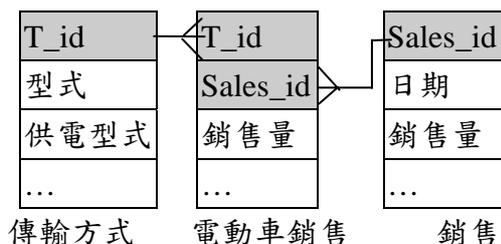


圖 8：電動車銷售的星狀綱要

表 5：汽車銷售資料晶圓的事實表格

Kind	Date		Sales
	Year	Month	
Gas Engine	2000	1	100
	2000	2	200
	2000	ALL	300
	ALL	1	100
	ALL	2	200
	ALL	ALL	300
Electronic Engine	2000	1	10
	2000	2	20
	2000	ALL	30
	ALL	1	10
	ALL	2	20
	ALL	ALL	30
ALL	2000	1	110
ALL	2000	2	220
ALL	2000	ALL	330
ALL	ALL	1	110
ALL	ALL	2	220
ALL	ALL	ALL	330

表 6：電動車銷售資料晶圓的事實表格

kind	Power	Date		Sales	
		Year	Month		
Electronic Engine	Electro-Magnetic	2000	1	10	
		2000	ALL	10	
		ALL	1	10	
		ALL	ALL	10	
		Electronic	2000	2	20
			2000	ALL	20
	ALL		2	20	
	ALL		ALL	20	
	ALL		2000	1	10
	ALL		2000	2	20
	ALL	2000	ALL	30	
	ALL	ALL	1	10	
ALL	ALL	2	20		
ALL	ALL	ALL	30		

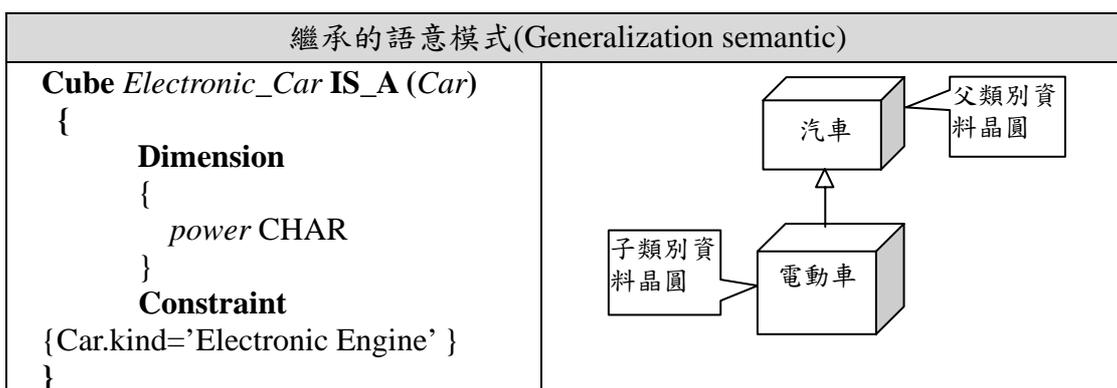


圖 9：電動車資料晶圓語意模型

圖 10：汽車與電動車資料晶圓之繼承關係

繼承不但可加強晶圓間彼此的關係，而且可以省掉不必要的重複資料(如表 5 與表 6 的灰色部份)。從另一方面來看，有了繼承的關係後，將可以大大減少資

料晶圓的資料量，例如上述的汽車銷售例子，如果一開始就將 Power 屬性放入汽車銷售資料晶圓的維度中，則資料量將會大幅增加，但使用本研究所發展出來的語意資料晶圓模型後，則資料量將可大為減少一個維度的資料記錄來達到原來所要的功能。

3.2. 資料晶圓的聚合關係

聚合 (Aggregation) 的語意乃指可將幾個物件整合成單一物件，即 A-PART-OF 的關係。例如「汽車」這個物件可以是由『引擎』與『車身』等物件類別所構成，像這樣的關係即是所謂的聚合(如圖 11)。

現舉一例子來說明晶圓間聚合的關係。若我們想要分析汽車的主要的零件引擎與車身的維護費，則我們可分別建置此二主要零件的晶圓(即 Engine 晶圓與 Body 晶圓)；若我們要進一步分析汽車的維護費則可建置一汽車的晶圓(即 Car 晶圓)，此晶圓可將引擎(Engine)與車身(Body)等晶圓聚合起來的。圖 12 顯示在本研究中此語意所表示的語法。

一部汽車可能會有很多種不同類的引擎與車身，而每個引擎維護的費用不同，而車身烤漆費用也不相同，如表 7 及表 8 為 7、8 月份某 N 牌修車廠的引擎 (Engine) 晶圓與車身(Body)晶圓的資料。而表 9 則為汽車(Car)晶圓的資料，透過聚合的關係，其資料由表 7 及表 8 資料晶圓聚合而成，因此在 Engine_maintenance 與 Body_maintenance 欄位中存放具其所連結的資料的物件識別碼 (Object Identifier)，以方便適時的參照其他資料晶圓的資訊。如此將使一個資料晶圓能囊括其它不同資料晶圓的資料，並將其組合起來，使整個資料倉儲系統中資料晶圓間的語意更為強烈，資料晶圓間亦不再只是獨立的資訊，而且可減少大量的重覆性資料。

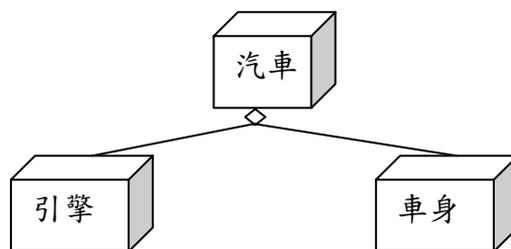


圖 11：資料晶圓語意模型的聚合關係

聚合的語意模式(Aggregation semantic)
Cube Engine { Dimension { <i>Engine_type</i> CHAR <i>year</i> NUMBER <i>month</i> NUMBER <i>Fees</i> NUMBER} Level { <i>date</i> (year(<i>month</i>))} Measurement {(SUM, <i>Fees</i>)} }
Cube Body { Dimension { <i>Body_type</i> CHAR <i>year</i> NUMBER <i>month</i> NUMBER <i>Fees</i> NUMBER} Level { <i>date</i> (year(<i>month</i>))} Measurement {(SUM, <i>Fees</i>)} }
Cube Car { Dimension { <i>Transmission</i> CHAR <i>year</i> NUMBER <i>month</i> NUMBER <i>Engine_maintenance</i> A_PART_OF (Engine) <i>Body_maintenance</i> A_PART_OF (Body) } Level { <i>date</i> (year(<i>month</i>))} Measurement {(SUM, <i>sales</i>)} }

圖 12：資料晶圓語意模型的聚合關係

表 7：引擎維護費的事實表格

Engine_type	Date		Fees
	Year	Month	
Turbo	2000	7	30000
VTEC	2000	8	5000
Turbo	2000	ALL	30000
Turbo	ALL	7	30000
Turbo	ALL	ALL	30000
VTEC	2000	ALL	5000
VTEC	ALL	8	5000
VTEC	ALL	ALL	5000
ALL	2000	7	30000
ALL	2000	8	5000
ALL	2000	ALL	35000
ALL	ALL	7	30000
ALL	ALL	8	5000
ALL	ALL	ALL	35000

表 8：車身烤漆的事實表格

Body_type	Date		Fees
	Year	Month	
Sporty	2000	7	30000
Wagon	2000	8	50000
Sporty	2000	ALL	30000
Sporty	ALL	7	30000
Sporty	ALL	ALL	30000
Wagon	2000	ALL	50000
Wagon	ALL	8	50000
Wagon	ALL	ALL	50000
ALL	2000	7	30000
ALL	2000	8	50000
ALL	2000	ALL	80000
ALL	ALL	7	30000
ALL	ALL	8	50000
ALL	ALL	ALL	80000

表 9：汽車銷售資料晶圓的事實表格

Transmission	Engine_maintenance	Body_maintenance	Date		Sales
			Year	Month	
Automatic	(Engine)	(Body)	2000	7	10000
Stick	(Engine)	(Body)	2000	8	5000
Automatic	(Engine)	(Body)	2000	ALL	10000
Automatic	(Engine)	(Body)	ALL	7	10000
Automatic	(Engine)	(Body)	ALL	ALL	10000
Stick	(Engine)	(Body)	2000	ALL	5000
Stick	(Engine)	(Body)	ALL	8	5000
Stick	(Engine)	(Body)	ALL	ALL	5000
ALL	(Engine)	(Body)	2000	7	10000
ALL	(Engine)	(Body)	2000	8	5000
ALL	(Engine)	(Body)	2000	ALL	15000
ALL	(Engine)	(Body)	ALL	7	10000
ALL	(Engine)	(Body)	ALL	8	5000
ALL	(Engine)	(Body)	ALL	ALL	15000

3.3. 資料晶圓的分類關係

分類是另一個常見的語意關係，其包含兩種基本的意義，一種為多重繼承觀念，另一種特性是以成員的觀念即 A-Member-Of 為主，本研究採用第二種定義。在此觀念下，所定義出來的類別均有一種對應關聯的特性，如同一救難團隊其包括了醫生、護士與消防人員等成員。

現利用一公司內部的案例來說明此關係在資料晶圓的應用，假設此公司要進行績效評估，因而建立了有銷售人員的績效評估晶圓(Sales)與技術人員

(Technique)績效評估晶圓，其資料分別如表 10 與表 11 所示。若此公司另外要分析專案的績效，而專案的成員為技術人員或業務人員，則可建置專案的績效語意晶圓，其資料表示如表 12。圖 13 顯示出本例中的語意晶元模型。

Categorization Operation	
CUBE Project {	
DIMENSIONS {	
<i>project_name</i> CHAR	
member	
A_MEMBER_OF (<i>Technique, Sales</i>)}	
MEASUREMENTS {(<i>Profit</i>)}	
}	

圖 13：資料晶圓語意模型的分類語意關係

表 10：銷售人員績效晶圓資料

Id	Name	Profit
01	Jack	50000
01	ALL	50000
ALL	Jack	50000
18	Sam	45000
18	ALL	45000
ALL	Sam	45000
ALL	ALL	95000

表 11：技術人員績效晶圓資料

Id	Name	Profit
08	Sandy	100000
08	ALL	100000
ALL	Sandy	100000
15	Carole	20000
15	ALL	20000
ALL	Carole	20000
ALL	ALL	120000

表 12：專案績效晶圓資料

project_name	member	Profit
Internet DW	(Sales,Technique)	100
OODW	(Sales,Tecnique,Sales)	70

Project_name	member		Profit
	Two	Three	
Internet DW	ALL	**	100
OODW	*	ALL	70
Internet DW	ALL	ALL	100
ALL	ALL	**	100
ALL	*	ALL	100
ALL	ALL	ALL	170

* = ((Sales),(Technique))

**=((Sales),(Technique),(Sales))

3.4. 語意資料晶圓的中介資料

在建置一個資料倉儲系統前，首先，應先設計一個可儲存資料倉儲資料結構的資料庫，以方便儲存我們所建立的資料晶圓相關結構與資料，而這樣的資料庫又稱爲之中介資料。本系統之中介資料架構(如圖 14)，總共包含了六個資料表，在圖中每一資料表有其對應的相關屬性值與主鍵，而每個資料表也可以透過這種關係相互連結，如此即形成了一個以星狀綱要爲結構的中介資料系統。表 13 說明各資料表的內容。

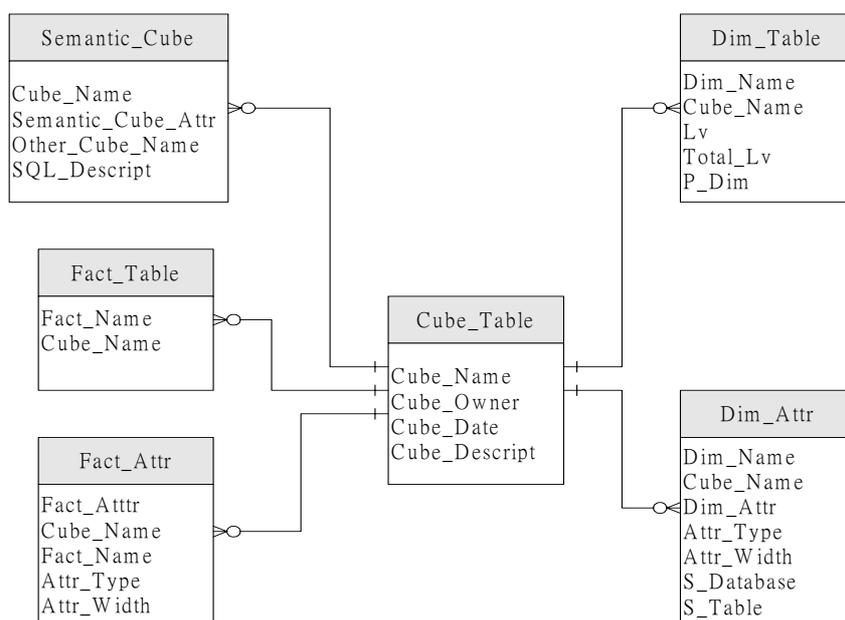


圖 14：語意中介資料

表 13：中介資料表格說明

中介資料表格	內容
Cube_Table	本研究中 Cube_Table 即是星狀綱要中的事實表格，它除了擁有事實表格的特性之外，並詳細記錄在資料倉儲系統中每一個資料晶圓的相關訊息，諸如資料晶圓名稱、擁有者、資料晶圓建立日期與其描述等資訊，並且以此為基礎向外延伸至相關的維度表格。
Semantic_Cube	Semantic_Cube 表格主要描述相對應的資料晶圓的關係與其對應的屬性關係，透過此表格的描述可以更清楚的知道資料晶圓間所建立的語意關係模型，諸如繼承、聚合、分類等，並透過結構化查詢語言(SQL)來實現相對應的資料。
Fact_Table	有了存放資料晶圓的事實表格後，皆下來即是存放有關於此維度資料晶圓之「事實表格」，其包含了資料晶圓名稱與事實表格名稱

	等
Fact_Attr	有了敘述事實維度的事實表格後，其相關事實表格的內容屬性值也應被建立在「事實屬性表格」中。
Dim_Table	定義出每個資料晶圓的各個維度屬性，其包括了資料晶圓名稱、維度名稱、階層等相關資訊，這些資訊存放在這個「維度資料表」。
Dim_Attr	這個「維度屬性資料表」用來存放相對應於維度資料表的各種屬性，其包括了，資料晶圓名稱、維度名稱、屬性資料等；如此的一個動作形成了整個中介資料結構。

3.5. 語意資料倉儲系統的查詢功能

本研究所提出的語意晶圓因符合星狀綱要結構，因此可搭配本研究之中介資料，來進行典型的資料晶圓基本查詢動作如上捲(Roll-Up)、下展(Drill-Down)、切片(Slice)、轉換(Dice)外，另外亦可進行特殊的語意查詢功能，其包括：

1. 放大(Zoom In)

當繼承或分類關係的語意功能被加入到資料晶圓時，即代表此資料晶圓具有放大或縮小資料的能力，也就是說透過語意的關係可以使資料晶圓去找出其相關的細部資訊，如圖 15 之演算法，假設目前的資料晶圓是顯示有關 1995 年銷售量的資料，但如果透過本研究的繼承關係，建立一關於 1995 年襯衫銷售量的資料晶圓時，即可透過『放大』的關係來查詢相關的資料晶圓，如圖 17，其放大之查詢語法如圖 16 所示。

```

FUNCTION ZOOMIN (Current_Measurement)
RETURN CUBE IS
    Inner_Cube CUBE
BEGIN
    SQL_DESC = "SELECT Cube_Name
                FROM
                Metadata.Semantic_Cube
                WHERE Related_Cube
                = Current_Cube_Name"
    Inner_Cube=EXECUTE (SQL_DESC)
    DATA = "SELECT * FROM Inner_Cube "
    Dimension_Data = EXEC(DATA)
    RETURN Inner_Cube
END
    
```

圖 15：放大動作演算法

```

SELECT ZOOMIN (產品銷售量) ON X-AXIS
FROM CUBES
    
```

圖 16：放大動作之語法

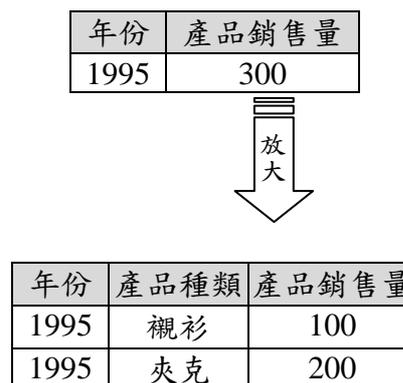


圖 17：放大動作

2. 縮小(Zoom Out)

這個動作與放大相反，它主要是將目前的資訊，縮小至原先的資料晶圓中，讓使用者方便查詢與分析較簡潔的資訊，如圖 18 之演算方式，譬如目前顯示 1995 年襯衫等產品銷售資訊，當執行一次縮小的動作時，即可見 1995 年的產品銷售狀況，如圖 20，而縮小之查詢動作語法如圖 19。

```

FUNCTION ZOOMOUT (Current_Cube)
RETURN CUBE IS
    Outer_Cube CUBE
BEGIN
    SQL_DESC = "SELECT Related_Cube
                FROM
                Metadata.Semantic_Cube
                WHERE Cube_Name =
                Current_Cube"
    Outer_CUBE = EXECUTE
                (SQL_DESC)
    DATA = "SELECT * FROM
            Outer_Cube "
    Dimension_Data = EXEC(DATA)
    RETURN Outer_Cube
END
  
```

圖 18：縮小動作演算法

```

SELECT ZOOMOUT (SALES) ON X-AXIS
FROM CUBES
  
```

圖 19：縮小動作之語法

年份	產品種類	產品銷售量
1995	襯衫	100
1995	夾克	200



年份	產品銷售量
1995	300

圖 20：縮小動作

3. 擴張(Expand)

而當聚合關係的語意功能被加入到資料晶圓時，即代表此資料晶圓具有擴張或簡化資料晶圓的能力，也就是說透過語意的關係可以使資料晶圓去找出其相關的資料晶圓，如圖 21 之演算方式，假設目前的資料晶圓是顯示有關於自排車銷售量的資料，但其引擎維護費用卻是由另一個引擎資料晶圓所提供，因此透過本研究的聚合關係，便可建立兩資料晶圓的關係，使其可透過『擴張』動作來查詢相關的資訊，如圖 23，而擴張之查詢動作語法如圖 22。

```

FUNCTION EXPAND (Cube)
RETURN CUBE IS
    Sub_Cube CUBE,Message
    VARCHAR
BEGIN
    IF Cube.Next != NULL
        SQL_DESC = " SELECT * FROM
                    Metadata Where
                    Cube_Name=Cu
                    be"
        Sub_Cube = EXEC(SQL_DESC)
        DATA = "SELECT * FROM
                Sub_Cube "
        Dimension_Data = EXEC(DATA)
        RETURN Sub_Cube
    ELSE
        Message = "This is Bottom of
data!"
        RETURN Message
    END IF
END
    
```

圖 21：擴張動作演算法

```

SELECT EXPAND (引擎) ON X-AXIS
FROM 引擎
    
```

圖 22：擴張動作之語法

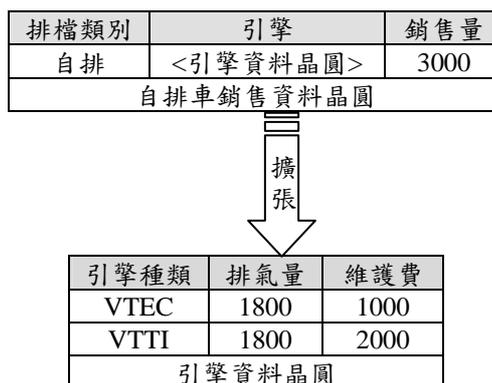


圖 23：擴張動作

4. 簡化(Narrow Down)

這個動作與擴張相反，它主要的動作是將目前所瀏覽的資料晶圓資訊，簡化至原先的資料晶圓中，提供使用者較簡潔的資訊以方便查詢與分析比較，如圖 24 之演算過程，譬如目前顯示引擎資料晶圓等產品的銷售資訊，當執行一次簡化的動作時，即可見到原先的產品銷售狀況，如圖 26，圖 25 為簡化動作之查詢語法。

5. 聚焦(Focus On)

這個功能主要是提供給上層的資料晶圓來使用，就像樹型的資料結構一般，使用者可以透過搜尋的方式將某一查詢規則套用在某一個資料節點上，應用這樣的關係即可產生相關的『聚焦動作』，其演算法如圖 27，例如一間公司擁有許多的部門，但如果上司只想看有關業務部這個資料晶圓的相關資訊時，即可透過此動作來查詢分析，如圖 29 所示，而聚焦之查詢動作語法如圖 28。

```

FUNCTION NarrowDown (Cube)
RETURN CUBE IS
    Parent_Cube CUBE,Message VARCHAR
BEGIN
    IF Cube.Previous != NULL
        SQL_DESC = "SELECT * FROM
                    Metadata Where
                    Cube_Name=Cube"
    Parent_Cube = EXEC(SQL_DESC)
    DATA = "SELECT * FROM
            Parent_Cube"
    Dimension_Data = EXEC(DATA)
    RETURN Parent_Cube
    ELSE
        Message = "This is Top of data!"
    RETURN Message
    END IF
END
        
```

圖 24：簡化動作演算法

SELECT NARROWDOWN (自排車銷售量) ON X-AXIS
FROM 自排車銷售量

圖 25：簡化動作之語法

引擎種類	排氣量	維護費
VTEC	1800	1000
VTTI	1800	2000

引擎資料晶圓

排檔類別	引擎	銷售量
自排	<引擎資料晶圓>	3000

自排車銷售資料晶圓

圖 26：簡化動作

```

FUNCTION FOCUSON (Cube)
RETURN CUBE IS
    New_Cube CUBE
Begin
    SQL_DESC = "SELECT * FROM
                Semantic_Cube of
                metadata Where
                Cube_Name=Cube"
    New_Cube = EXEC(SQL_DESC)
    DATA = "SELECT * FROM
            New_Cube"
    Dimension_Data = EXEC(DATA)
    RETURN New_Cube
End
        
```

圖 27：聚焦動作演算法

SELECT FOCUSON (公司) ON X-AXIS
FROM 公司
WHERE 公司.對應部門='業務'

圖 28：聚焦動作之語法

資料晶圓	對應部門	年份
<Sales>	業務	2000
...		...

公司資料晶圓

部門	項目	金額
業務	交際費	3000
...

業務部資料晶圓

圖 29：聚焦動作

4. 系統架構與建置方法

爲了驗證本研究的語意資料晶圓模型之可行性，我們利用系統雛形實作方式來驗證。本系統採用三層式架構的設計方式，其架構如圖 30 所示。

本系統的開發分成前端介面與後端伺服器兩方面，均利用 JAVA 語言來設計。前端介面主要功能爲允許使用者利用網頁來瀏覽語意資料晶圓，並可執行資料晶圓的查詢功能；後端則將語意晶圓建置於 MS-SQL Server 2000 資料庫上，資料晶圓設計者利用網頁上的語意晶圓建置程式將語意晶圓存入資料庫中；前端與後端的資料連結透過 JDBC 來進行。

4.1. 系統介面

圖 31 爲我們的後端語意資料晶圓建置系統，要建置一語意資料晶圓可分爲五大步驟(Step1~Step5)，而步驟六(Step6) 爲系統預覽的功能。和一般利用星狀網要建置資料晶圓系統不同之處在於，於晶圓語意處多一建置 IS-A 關係的功能，另外於 Step2 處可建置 A-Part-Of 與 A-Member-Of 語意關係。

圖 32 則顯示前端系統瀏覽語意晶圓的畫面，當使用者對某一資料有興趣，則可利用 Zoom-In 功能連結至另一晶圓進行更細部的資料瀏覽，以增加分析的能力。

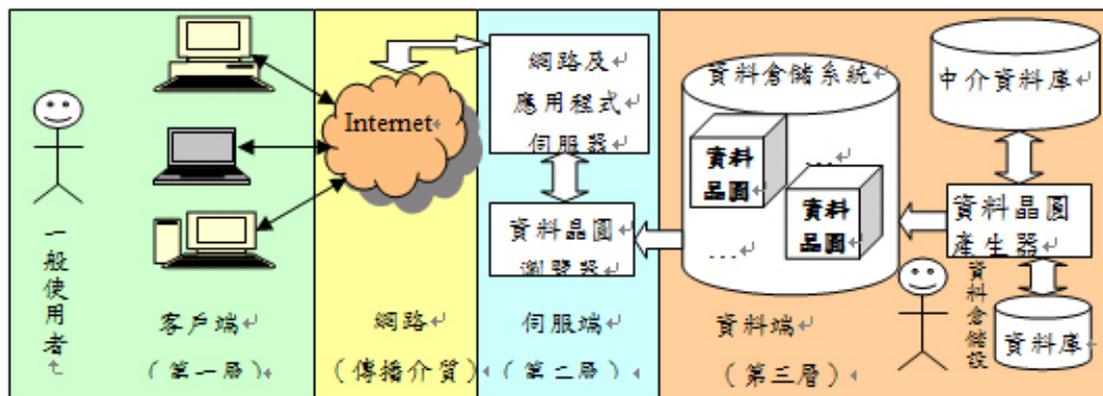


圖 30：網際網路資料倉儲架構

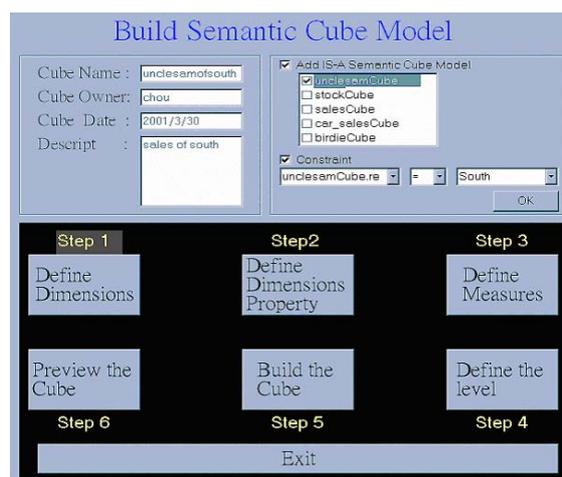


圖 31：語意資料晶圓建置系統

年份	區域	款式	分區	種類
UNIT: 金額				
區域	區域			
年份	年份			
款式	款式			
款式	414000		11751700	
1區	1250000		3842900	
總計	2044000		7600000	

圖 32：語意資料晶圓前端系統

5. 系統評估與個案探討

為了實際驗證本研究的假設，在本論文的最後以個案研究的方式，將本系統實際應用到兩個案中，並藉由實地的應用與訪談來評估其成效。

個案一：服裝公司

此個案係為國內一家知名的服飾廠商，該公司在全省各大百貨公司擁有數十家自營店與專櫃，目前這些店家分佈於全省各地，包括台中、嘉義、台南、高雄等地。

在本個案中的企業由於遭受到許多國內外廠商的競爭壓力，因此先前已建置一網際網路資料倉儲系統來提供給該公司的店面銷售人員、公司內部人員、以及有策略聯盟關係的供應商，可由網際網路來瀏覽分析資料，以便可提高整體的經營品質。圖 33 為該公司的之前的網際網路資料倉儲系統畫面雛形(黃士銘, 2002)。



圖 33：個案公司網際網路資料倉儲系統前端瀏覽介面-銷售金額分析

在網際網路資料倉儲系統建置後半年後，該公司發現其所建置的資料晶圓數變多且日漸複雜，另外資料量亦變大，而且使用者提出需求希望能建立資料晶圓與資料晶圓間的關係，以利他們進行更彈性化的查詢。因此我們利用本研究所提出的技術來解決其問題。

在我們的實驗過程中，原本此企業已建立一個銷售量分析的資料晶圓叫 Sales，此晶圓提供使用者可多維查詢整個公司的銷售資料，假設該公司的經理想要細部分析南部的銷售情況時，其資料倉儲管理員即可利用本研究所發展之語意資料晶圓模型介面，建立一 South_Region 的晶圓(如圖 34)，此晶圓提供更詳細的南部地區銷售分析資料，而且透過 IS-A 關係的建立和 Sales 晶圓建立起繼承的語意關係。如此使用者可透過 ZOOM-IN 與 ZOOM-OUT 功能在兩晶圓間進行交叉的流覽，如此增加資料晶圓間的查詢功能，使用者亦反應使用上較為方便，增加查詢的能力。

```

Semantic Cube Model
Cube South_Region IS_A (Sales) {
  Dimension
  {
    region CHAR
    year CHAR
    month CHAR
  }
  Meadurement
  {
    amount int
    quantity int
  }
  Constraint
  {
    unclesam.region='South'
  }
}
    
```

圖 34：語意資料模型

為驗證資料晶圓的資料是否有減少，我們選取個案公司的部分資料來進行實驗，其實驗的原始資料總數為 3456 筆資料記錄(即 2 個年份*12 個月*2 個區域*4 家分店*3 種款式*6 種產品)，將其製成 Sales 資料晶圓後產生的資料總數為 16380 筆資料記錄(此為依據 Gray J.1997 的演算法)，如果想要增加一個有關於南區銷售的資料晶元 South_Region 而不新增加欄位，則利用傳統的方法將產生 5460 筆資料記錄(i.e. $3*13*5*4*7$)，但透過本研究所提出的語意資料晶元模型，則只需要一些紀錄中介資料的資訊，不需要有這些的資料記錄，因為這幾筆資料記錄已經存在於原先的資料晶元中，不需額外產生，這是本研究與一般的資料倉儲系統不同之處，現將其資料產生數量與市面上所販售的資料倉儲系統(以微軟的 OLAP SERVER 為例)，作一分析與比較(如表 14)。

表 14：資料量比較分析

原始資料筆數	3456 筆資料記錄 = 2 個年份 * 12 個月 * 2 個區域 * 4 家分店 * 3 種款式 * 6 種產品	
Sales 資料晶圓	16380 筆資料記錄 = $3 * 13 * 3 * 5 * 4 * 7$	
	本研究之語意資料晶元模型	MS-OLAP Server
建置一個新的 South_Regio 資料晶圓的資料筆數	不需額外建立，只需透過本語意資料晶元模型即可	需額外建立 5460 筆資料記錄 ($3*13*5*4*7$)

個案二：某家電信公司

除了上述個案資料之外，本研究另將此項研究成果套用在國內某家電信公司之 ADSL 業務申租分析上，該公司之 ADSL 業務依計費方式分為計時制 (Timing) 與固定制 (Fixed) 兩種，而每種計費方式皆提供 Base、Enhance、Advance 三種速率，分別為 512K/64K、768K/128K、1.5M/384K。

該公司的一個營業據點平均每月 ADSL 受理量約為兩萬多件，而全省共計有十多個據點。在過去，該業者為了要分析某一種計費方式常常要把許多資料一同倒入資料倉儲中，不僅資料移轉費時，同時也浪費許多儲存空間，因此期望能利用本研究來解決上述問題。

本研究計畫擬將其受理情形建置成三個名稱為 Base、Enhance 與 Advance 的資料晶圓，其建置方式如圖 35：

Semantic Cube Model
Cube Base { Dimension { <i>count_rate</i> CHAR <i>year</i> NUMBER <i>month</i> NUMBER <i>quantity</i> NUMBER} Level { <i>date</i> (<i>year</i> (<i>month</i>))} Measurement {(SUM, <i>quantity</i>)} }
Cube Enhance { Dimension { <i>count_rate</i> CHAR <i>year</i> NUMBER <i>month</i> NUMBER <i>quantity</i> NUMBER} Level { <i>date</i> (<i>year</i> (<i>month</i>))} Measurement {(SUM, <i>quantity</i>)} }
Cube Advance { Dimension { <i>count_rate</i> CHAR <i>year</i> NUMBER <i>month</i> NUMBER <i>quantity</i> NUMBER} Level { <i>date</i> (<i>year</i> (<i>month</i>))} Measurement {(SUM, <i>quantity</i>)} }

圖 35：語意資料模型

這些晶圓除了提供使用者多維查詢功能外，亦可整合本研究所發展之語意資料晶圓模型介面，假設該公司的主管想要分析固定制的銷售情況時，其資料倉儲管理員即可利用本研究，建立一 Fixed 的晶圓(如圖 36)，此晶圓將提供更詳細的分析資料，它主要透過 A_Member_OF 關係和圖 35 之晶圓建立語意關係。如此使用者可透過 FOCUSON 功能在晶圓間進行聚焦的流覽，如此增加資料晶圓間的查詢功能，增加資料晶圓附加的查詢能力。

Semantic Cube Model
<pre> Cube Fixed { Dimension { year NUMBER month NUMBER count_rate A_Member_OF (Base) count_rate A_Member_OF (Enhance) count_rate A_Member_OF (Advance) } Level { date (year(month))} Measurement {(SUM, quantuty)} } </pre>

圖 36：語意資料模型

在過去，該公司爲了做同樣的動作，常常需要將三個資料晶圓的資料建置在同一個資料晶圓中，然後再利用 Drill-Down 或 Roll-Up 動作，來協助篩選符合條件的資料，此舉不僅需要儲存雙倍的空間外，同時也降低查詢時的速度(因龐大的資料量不利於搜尋動作)；但將資料晶圓加入語意之後，只需要紀錄一些中介資料的資訊，即可達到該企業之要求，因爲這幾筆資料記錄已經存在於原先的資料晶元中，不需額外產生，不僅減少儲存的空間，也使查詢的功能多樣化。現將其資料晶圓建置方式與市面上所販售的資料倉儲系統(以微軟的 OLAP SERVER 爲例)，作一分析與比較(如表 15)

表 15：資料晶圓建置說明比較

	本研究之語意資料晶元模型	MS-OLAP Server
建置一個新的 Fixed 資料晶圓所需的動作	不需額外建立，只需透過本語意資料晶元模型即可	需額外建立三個資料晶圓(每個資料晶圓皆有上萬筆資料)

在經過數次的實驗後，我們發現資料量有明顯減少的顯著現象，在初步的對使用者訪談，使用者對查詢的滿意度亦提高。爲了整體的評估本研究之系統效能，特別將 Gray J.1997 等學者所提之演算法、微軟的 OLAP SERVER 與本研究列表如下，並做一簡單效能評估：

		本研究	微軟 SQL Server 2000 Analysis Services	Gray J. 1997
查詢功能	上捲	√	√	√
	下展	√	√	√
	切片	√	√	√
	轉換	√	√	√
	放大	√	√	
	縮小	√	√	
	擴張	√		
	簡化	√		
	聚焦	√		
加入語意功能		√	√	
使用資料量[註]		少	較多	多

[註]資料量的多寡依所選擇維度資料而定，如果以表 5 來利用語意資料晶圓模式建立電動車的資料晶圓，將可減少一個維度的資料量(如表 6 中灰色部分)

6. 結論與未來發展

本研究主要是探討一資料倉儲系統內資料晶圓間的互動關係，透過所設計的語意資料晶圓模型來強化晶圓間的相互關聯，並開發一前端資料晶圓瀏覽介面，方便使用者查詢與分析。而本研究主要的貢獻有：

1. 在資料晶圓中加入語意功能，加強每個獨立資料晶圓與其他資料晶圓的互動
2. 減少資料晶圓中不必要的維度資訊，進而減少資料量
3. 增加查詢的多樣性與豐富維度相關資訊
4. 提昇整個資料倉儲系統的效能

除此之外，本研究亦提供一完整的系統架構來驗證整個語意資料晶圓模型，而在這架構中亦採用自行設計的中介資料模式來存放每一個資料晶圓的細部資訊與相關的建立方式，以方便系統使用者維護資料晶圓，進而提供前端使用者更豐富的查詢功能與分析，而未來本研究將朝向與正規化語言結合，並將此作為本研究的下一階段重要項目。

要實作建立一套成功的資料倉儲系統，需要完善的規劃與設計方法，而這也是我們所努力的目標，在完成本研究後，可以得知資料倉儲系統的設計者不但可以充分利用已經存在的資料晶圓外，同時亦可套用已存在之資料晶圓建構新的資料晶圓，並加強其語意關係，除此之外，減少儲存空間，亦是本研究之重要貢

獻之一。

誌謝

本研究承蒙國家科學委員會補助研究經費(計劃編號： NSC89-2213-E-194-041)，特此致謝。另外，感謝薛祥元、蘇俊豪、吳宗瑾、楊君偉等人協助開發雛形系統與個案導入，在此一併致謝。

參考文獻

- [1] 黃士銘(2002)，”建置一網際網路資料倉儲系統”，《資訊管理學報》，9(1)，135-152。
- [2] Abbadi, D.A., Singh, A., and Yurek, T.(1997), “Efficient View Maintenance at Data Warehouses”, *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 26(2), 417-427.
- [3] Abello, A. and Samos, J.(2002), “YAM / sup2 / (yet another multidimensional model): an extension of UML”, *Database Engineering and Applications Symposium, 2002. Proceedings. International*, 172-181.
- [4] Bellahsene, Z.(2002), “Schema Evolution in Data Warehouses,” *Knowledge and Information Systems*, 283-304.
- [5] Bensberg, F.(2003), “Controlling the data warehouse - a balanced scorecard approach”, *Proceedings of the 25th International Conference on Information Technology Interfaces*, 127-135.
- [6] Chen, W.C. and Hong, T.P.(1999), “A Composite Data Model in Object-Oriented Data Warehousing”, *Proceeding of IEEE Conference on Technology of Object-Oriented Languages and Systems*, 400-405.
- [7] Devoney, C. (1997), “In Search of 56K,” *Computerworld*, 77-80.
- [8] Fong, J.(1995), “Mapping Extended Entity Relationship Model to object Modeling Techniques”, *ACM SIGMOD Record*, 24(3), 18-22.
- [9] Gardner and S.R.(1998), “Building the Data Warehouse”, *Communications of ACM*, 41(9) 52-58.
- [10] Gopalkrishnam, V., Qing, L., and Karlapalem, K.(1998), ” Issues of object-relational view design in data warehousing environment,” *Proceeding of IEEE International Conference on Systems, Man, and Cybernetics*, 3, 2732-2737.
- [11] Gray, J. et. al.(1997), “Data Cube: A Relational Aggregation Operator Generalizing Group-By, Gross-Tab, and Sub-Totals”, *Data Mining and Knowledge Discovery*, 29-53
- [12] Hurtado, C.A.; Mendelzon, A.O. and Vaisman, A.A.(1999), “Maintaining data

- cubes under dimension updates,” *Data and Engineering*, 346-355.
- [13] Indulska, M.(1999), “Shared Result Identification for Materialized View Selection”, *Proceeding of IEEE Database Conference*, 49-56.
- [14] Inmon, W.H., Welch, J.D., Glassey, K.L.(1997), *Managing the Data Warehouse*, Wiley Computer Publishing.
- [15] Miller, L.L., Honavar, V., Wong, J. and Nilakanta, S.(1998), “Object-Oriented Data Warehouse for Information Fusion from Heterogeneous Distributed Data and Knowledge Sources”, *Proceeding of IEEE Information Technology Conference*, 27-30.
- [16] Papadias, D., Yufei Tao, Kanis, P., Jun Zhang(2002), “Indexing spatio-temporal data warehouses”, *Proceedings of the 18th International Conference on Data Engineering*, 166-175.
- [17] Riedewald, M.; Agrawal, D. and El abbadi, A.(2000), “pCube:Update-Efficient Online Aggregation with Progressive Feedback and Error Bounds,” *Proceeding of IEEE 12th International Conference on Scientific and Statistical Database Management*, 95-108.
- [18] William A.(2000), *Object-Oriented Data Warehouse Design*, PHPTR.
- [19] Wu, L., Miller, L. and Nilakanta, S.(2001), “Design of data warehouses using metadata,” *Information and Software Technology*, 43(2), 109-119.

