

# 以物件導向方法設計資料倉儲架構之研究 -以業務銷售作業模型為例

## Object-Oriented Design for Data Warehouse Data Schema -An Data Model on Sales Operation

孫明照

蘇偉仁

蕭榮興

財團法人資訊工業策進會  
電子商務技術實驗室

財團法人資訊工業策進會  
電子商務技術實驗室

財團法人資訊工業策進會  
電子商務技術實驗室

Ming Chao Sun

Wei Jen Su

Jung Hsing Hsiao

[vincents@iii.org.tw](mailto:vincents@iii.org.tw)

[Suweijen@iii.org.tw](mailto:Suweijen@iii.org.tw)

[scottie@iii.org.tw](mailto:scottie@iii.org.tw)

### 摘要

資訊對於企業而言是項極重要的資產,若能充分利用現有資訊於分析決策上,將可為企業帶來更大的競爭優勢。而資料倉儲技術便是從多個異質資料來源,收集並整合成有效之資訊儲存於資料倉儲中,如此才能快速地提供企業決策用分析性資訊。在資料倉儲的建立方法上,傳統都是建構在關連式資料庫之上,以 Table Join 跟事先計算好 Aggregation 資料的方式來達到快速提供企業分析性資訊的特點。

然而建構在傳統關連式資料庫的資料倉儲架構,只能適用於所處環境變動幅度較小的企業,當企業必須面對的是國際性的競爭壓力與快速變動的環境,傳統的關連式資料倉儲缺乏調整彈性的缺點便浮現出來。本篇論文將利用物件導向方法來設計資料倉儲架構,並以業務銷售作業為例,提出符合 Domain 需求的物件導向資料倉儲架構。並針對企業常常面臨到解決問題時,決策資訊無法對應至作業流程的困難點,本論文在以物件導向方法建置資料倉儲之前,先對企業做流程分析,再將流程分析的結果作為物件導向方法需求分析的輸入資訊,讓本研究的結果物件導向資料倉儲架構能與企業流程緊密結合。

註：本研究由經濟部委託財團法人資訊工業策進會網路前瞻應用技術發展四年計畫辦理

關鍵詞：物件導向資料倉儲、流程分析方法、統一化程序(Rational Unified Process)。

### 一、前言

隨著產業環境的快速變遷及資訊科技的進步,企業為了因應外在不斷改變的環境,在不同的時期產生了不同的企業資訊系統,使得企業間的資訊散落在各種異質平台與資料庫上,導致資料整合不易與無法有效率被利用的缺點。而現今屬於知識經濟的時代,企業面臨激烈的國際化競爭,如何運用資訊科技將大量分散的資料轉變成有用的資訊,並經過儲存及分析後輔助企業做出正確的決策,有效掌握客戶需求,這樣的議題是目前企業所面臨最重要的挑戰。

資料倉儲 (Data Warehouse)的技術,便是在蒐集分散各個不同平台、不同儲存媒體的資料,將其整合成一個大型的資料庫,使企業決策者能快速的掌握決策資訊,進而做出正確且及時的決策。

傳統設計資料倉儲的方式,大多採用關連式資料模式(Relational Data Model),但隨著資料的呈現不再只侷限於結構化的文字模式,非結構化的資料模式(如影像、聲音、圖片等等)更是隱藏了大量的

企業知識 為了解決傳統資料倉儲缺乏應變彈性與處理非結構資料的缺點,本研究提出利用物件導向資料模式(Object-Oriented Data Model)來設計資料倉儲,增加企業對於外部環境的應變能力,與軟體重複使用(Software Reuse)【Jacobson, I., 1998】的機制。

企業在透過資料倉儲架構獲取資訊的方法上,常常面臨到的一個問題,就是發現資訊不正確時,系統往往無法立即反應是哪個作業出了問題,造成危機處理的延遲 為了讓主管決策所需要的資訊,與作業流程能存在緊密的關係,以協助企業應變與預防錯誤的發生,本研究在建立資料倉儲階段前,先對企業做流程分析,再將流程分析結果,作為物件導向方法的需求分析來源,擷取出企業的主要流程與關鍵資訊,讓本研究的結果物件導向資料倉儲架構能與企業流程緊密結合。

## 二、文獻探討

### 2.1 流程分析方法

流程細部展開 ( Decomposition ) 方法與技術簡化了流程管理,並使作業透明化、權責清楚劃分、介面連結關係明朗。要如何展開便需要各種方法,而流程分析的方法眾多,有 DFD ( Data Flow Diagram )、 IDEF ( Integration DEfinition for Function modeling )【Mayer, 1998】、 ARIS (Architecture of Integrated Information System)整合性資訊系統架構【Scheer, 2000】 Petri Nets 斐氏網系統模式法【Murata, 1989】 CIM-OSA(CIM Open System Architecture)【Joryze, 1990】 GRAI Method【Chaudhuri, 1996】、 OOMIS (Object-Oriented modeling methodology for Manufacturing Information System)製造資訊系統物件導向塑模技術等。

流程分析方法中由於 IDEF0 方法涵蓋了功能面、行為面、組織面、資訊面四種構面的特性,是一種可以充分表達作業流程特性的模式方法 近年

來在學術研究或產業界, IDEF0 經常被使用去規劃複雜的製造系統之運作功能,以及企業的作業流程的一個模式化工具。它是現況分析( AS-IS Model )的一項工具,具有圖形表示法、從上而下的分解、易了解、易溝通、功能模式及主題明確的特質。其主要目的在了解問題本身,而非解決問題的方法,也就是說決定解決方法之前,須先針對問題本身提出清晰完整的分析模式,因此本研究採用 IDEF0 方法為流程分析之方法。

### 2.2 物件導向方法

物件導向方法是一種反覆的程序,主要包括需求分析、系統分析與設計、實施與測試等階段。Grady Booch 於 1986 年率先發表物件導向的系統開發方法,開啟了物件導向技術在軟體工程上應用的開始。Rumbaugh 於 1995 年提出了物件模式技術 (Object Model Technique,OMT)【James E., 1995】,包含了物件模式(Object Model)、動態模式(Dynamic Model)與功能模式(Functional Model)等三種模型。Jacobson 提出使用個案 USE CASE【Jacobson, 1995】,用來描述使用者與系統如何互動也就是說將每一個使用者視為一個行動者 (Actor),當行動者引發系統執行某項任務時,會發生一連串與系統有關的處理,將這些處理紀錄下來即成為「使用個案」。Coad&Yourdon 認為物件導向分析技術主要包括五個主要的分析步驟,分別是定義主題、確認物件、確認結構、定義屬性、定義服務等【Coad, 1989】。直到 1998 年,Booch、Rumbaugh、Jacobson 等三位物件導向大師,整合了各自特色,提出統一化程序(Rational Unified Process, RUP)【Rational Corp., 1998】,讓物件導向軟體工程方法進入了整合的新世紀。

#### 2.2.1 統一化程序(Rational Unified Process)

統一化程序是一個軟體發展的方法,而一個軟體發展的方法就是將使用者的需求轉換成軟體系統的一系列活動 統一化程序是一個元件導向的方

法，利用軟體元件來建立系統，透過良好定義的介面來互相溝通【Booch, Jacobson, Rumbaugh, 1998】，使用元件基礎的系統架構提供了一些軟體開發時會面臨到發展問題的解決方案【Kruchten, 1999】：

- 元件的採用增加了系統架構的彈性。
- 模組化將系統分成一些定義清楚的元素，並且容易維護。
- 重複使用的特性可藉由一些標準元件架構的採用而獲得更大的效益。
- 元件提供了一個簡單的組構管理環境。
- 視覺化的塑模工具提供了自動化的元件發展。

統一化程序是以使用者案例為驅動因子（Use-Case Driven），並以軟體架構為中心（Architecture-Centric），同時也是循環式及漸進式（Iterative and Incremental）的軟體發展程序【Kruchten, 1999】。它的發展流程分成兩個維度，一個是時間，另一個是核心工作流程元件，如圖 2-1 所示。在時間方面，分成了四個生命週期（lifecycle），Inception、Elaboration、Construction、Transition，每個週期結束都會有一個里程碑（milestone），用來評估每個週期結果的正確性。一個生命週期包含了六個流程元件，Business Modeling、Re-requirements、Analysis and Design、Implementation、Test、Deployment。四個生命週期便是四個循環（cycle），每個循環中都必須經歷六種工作流程，以這種循環且漸進式的發展來組合起兩個維度的相互關係。

四個生命週期中，Inception 稱為初始階段，這個階段的主要目的必須建立系統的企業案例（Business case），跟確定系統範圍。企業案例包含了成功準則、危機評估、資源需求的評估、和階段計劃完成時間。另外，初始使用者案例的建立，用來記錄使用者需求和系統功能與外部環境之交

互作用，而系統範圍的定義也經由使用者案例圖的建立而明朗化。在這個階段的產物有 Use-Case Model（完成整個系統使用者案例的 10%~20%）、企業案例、危機評估、和專案的規劃。

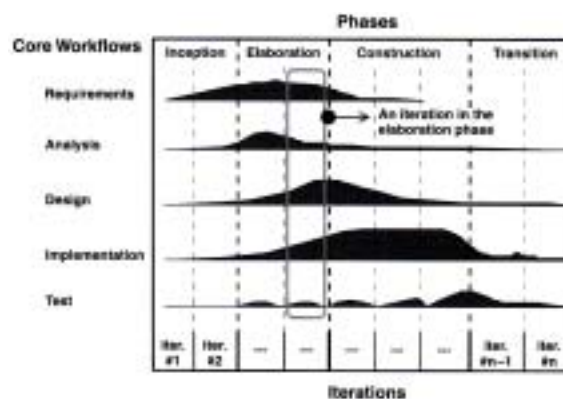


圖 2-1 統一化程序架構圖

資料來源：【Booch, Jacobson, Rumbaugh, 1998】

### 2.3 資料倉儲

資料倉儲的定義是由資料倉儲之父 Inmon, W.H. 所提出【Inmon, W.H., 1992】，在此節中將詳細說明其理論重點。此節共分為兩小節，在第 2.3.1 小節中說明資料倉儲的定義，在第 2.3.2 小節中說明多維度資料模型。

#### 2.3.1 資料倉儲的定義

資料倉儲的建立的主要目的在於資料除垢（Data Cleaning）與資料整合（Data Integration），此正為線上分析處理、資料分析與資料挖掘的前置工作，若不經由資料倉儲，則會造成不同時間點分析結果不相同的窘境。何謂資料倉儲呢？資料倉儲之父 Inmon, W.H. 對於建構資料倉儲所下的四個定義【Inmon, W.H., 1992】：資料倉儲具有主題性（Subject-oriented）、時變性（Time-variant）、整合性（Integrated）與永存性（Non-volatile）四大特色。

資料倉儲的出現並非取代傳統關聯式資料庫的地位，而是在其基礎上進行資料的加值處理，其運作方式為，先將儲存於企業內、外部的異質資料庫進行粹取、轉換、壓縮及載入，並放置在資料倉儲中，並透過線上分析處理機制，能夠快速地對資料進行分析、查詢、報表，也可在資料倉儲上進行

資料挖掘等決策分析功能,因此資料倉儲與資料庫系統是相輔相成的,並非背道而馳。

### 2.3.2 多維度資料模型

多維度資料模型 (Multidimensional Data Model), 主要是由多個維度及衡量值(measures)所構成,其物理意義為分析人員利用哪幾個面向來觀察此衡量值。就維度而言,各維度由多個資料庫中多個欄位屬性所組成,各維度皆有層級關係,如時間維度其層級為年>季>月>日;就衡量值而言,視研究主題而定,可為銷售量、交易量等。若以多維度資料庫(Multidimensional Database)而言,則可直接實作,但是,若以關聯式資料庫(Relational Database)而言,要如何達到多維度的效果呢?其設計方式是以先決定分析的主題,依據此主題來設計事物表中的衡量值,並依據決策人員欲採用的分析面向,如時間、產品及地點等,根據此來設計維度資料表(Dimension Table),最後以事物表為中心,以各維度資料表中的主鍵(Primary Key)與事物表建立關聯。類似此種多維度模型分為星狀綱要(Star Schema)、雪花綱要(Snowflake Schema)對星狀綱要而言,其中間是事物表(Fact 資料表),而周圍則稱為維度表(Dimension 資料表),此種表達法具有三階正規化的精神,而雪花綱要則是把星狀綱要再做進一步的三階正規化。

## 三、物件導向資料倉儲設計方法

企業的運作瞬息萬變,主管如何掌握企業最新的狀況,來支援正確決策的製定,已是主管所面臨的一大挑戰。如今,隨著商業智慧軟體的成熟發展,許多企業可以直接購買現成的商業智慧套裝軟體,來滿足主管決策時的報表需求;不過即使如此,商業智慧軟體功能的需求,已不再是主管重要的訴求了,可以快速且正確的製作出需要的報表,才是左右主管決策的關鍵因素。如何確保資料倉儲的 Data Schema 已經涵蓋了所有主管的需求,避免最後連 Cube 都設計好了後,卻接二連三的發生主

管抱怨資訊不足的狀況。

本研究提出了物件導向資料倉儲架構,透過對企業的流程分析,尋找出企業營運的關鍵流程與資訊,再以這些需求為基礎,利用物件導向方法 RUP(Rational Unified Process)【Booch, Jacobson, and Rumbaugh, 1998】進行資料倉儲架構的設計,如圖 3-1 所示,歸納流程分析所產出的結果,將關鍵流程與資訊作為物件導向方法的需求分析輸入資訊,並於 Use-Case Model 中建立 Use-Case Diagram。接著將歸納每個 Use Case 中有哪些類別,於 Design Model 中建立 Class Diagram,表示這些類別的靜態關係,與每個類別所擁有的資訊,而這些資訊來源也將參考流程分析過程中,所產出的關鍵資訊,才能完整呈現使用者的資訊需求。最後,在 Implementation Model 中會將所有類別依照功能性與相關性,封裝成不同的元件,並於 Component Diagram 中表示物件導向資料倉儲架構結果。



圖 3-1 研究架構

## 四、物件導向資料倉儲模型過程

依照 RUP 所規範的標準程序,在不同的發展階段中將陸續建立六種不同的 Model,而這六種 Model 對應至 UML 表示時,本研究將採用三個觀點的呈現分析的結果,即為 Use-Case View、Logical View 與 Component View,這三種角度包含了 RUP 六種 Model 的呈現。

### 4.1 Use-Case View

Use-Case View 所要呈現的是業務銷售作業的

Use Case Diagram，用來描述企業於業務銷售作業的相關作業與參與作業的人員，本研究 Use Case Diagram 需求的來源，將於流程分析結果中，歸納出業務銷售作業從接單到出貨的相關作業流程與資訊，最後呈現於 Use-Case View，以作為物件導向資料倉儲架構的建立依據。

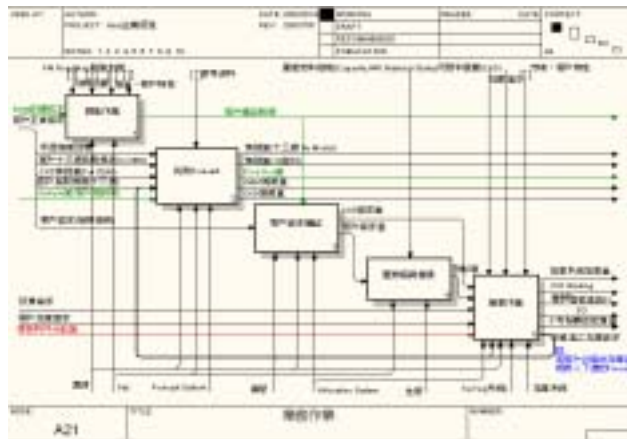


圖 4-1 業務作業 IDEF0 流程分析圖  
以業務銷售作業為例，將歸納流程分析結果，如圖 4-1 所示，提供本研究完成銷售作業所需要的主要流程與資訊，以列表方式歸納出來，如表 4-1 所示，列出所有業務銷售作業的主要流程與資訊，作為 Use Case 的推導參考，與類別中 Attribute 建立的依據。

表 4-1 業務銷售作業關鍵流程與資訊

作業流程	資訊
銷售作業	產品 Bottom Price BOM 總成本 客戶報價
預測作業	長期預測量 短期預測量 年度銷售目標
客戶需求確認	客戶短期需求量
產銷協調會議	分配量 生產產能 製造產能 原料料況 半成品數量
接單作業	維修料件分配量 加單需求量 砍單需求量 銷售訂單數量

作業流程	資訊
	採購訂單數量 採購發票數量
產能規劃作業	預測量 銷售訂單數量 砍單數量 工單數量 產能物料限制量 外包廠延伸產能 測試製具產能
物料需求規劃	每週預測量 採購需求量 採購訂單 原料料況 原料採購週期 庫存量 銷售訂單數量 工單數量
工廠作業	成品數量 半成品數量 砍單後產品數量
出貨作業	客戶付款紀錄 出貨單 出貨數量 採購發票 提貨單

Use Case Diagram 呈現了業務銷售作業的使用案例，如圖 4-1 所示，其中包含了客戶、業務、產品經理、生管與物控等角色；銷售作業、預測作業、客戶需求確認、接單作業與產銷協調會議等使用案例。這些作業活動在流程分析中是屬於業務銷售作業主要的流程作業，每個作業下都還有許多子作業來描述作業的流程，因此，圖 4-2 是以銷售作業為例，呈現了的主要的使用案例，每個 Use Case 將會依照作業的複雜度，往下設計出相對應的 Use-Case Diagram。

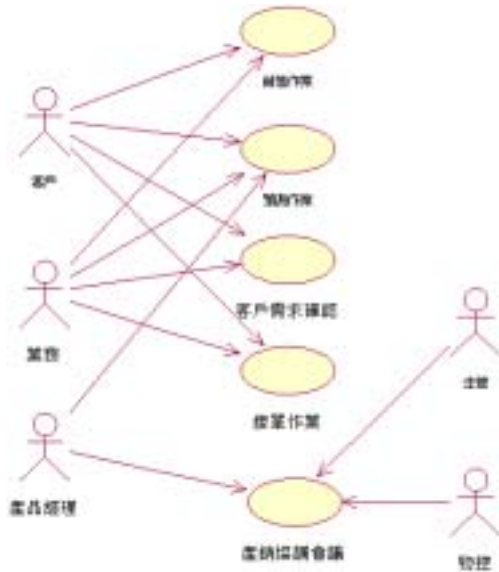


圖 4-2 業務銷售作業使用案例圖

#### 4.2 Logical View

Logical View 所要呈現的是業務銷售作業的類別圖，如圖 4-3 所示，於業務銷售作業中的使用案例去思考，需要哪些類別來完成業務銷售作業，將存在於使用案例中的類別逐一定義出來，最後再以類別圖來呈現類別間的靜態關係。因此，想要從由上一小節的 Use-Case Diagram 推導出 Class Diagram 時，應該經過下列兩個步驟，分別為步驟一：從 Use Case 中定義出類別，與步驟二：從 Use Case 中定義出類別的 Attribute，即將來資料倉儲的維度資訊。詳述於 4.2.1 定義類別與 4.2.2 定義類別 Attribute 小節。

##### 4.2.1 定義類別

類別是將擁有類似特性的物件抽象化而來的，因此在歸納 Use Case 中有哪些類別之前，必須先了解有哪些物件共同來完成一個 Use Case 的需求，再將這些物件抽象化成我們所需要的類別。以圖 4-2 的業務銷售作業使用案例圖為例，可從行為者(Actor)與使用案例(Use Case)分別去歸納，行為者部分，可以歸納出客戶、業務、產品經理、生管與物控等人員，而這些人員也就是操作這些使用案例的物件，抽象化後可以得到兩個類別，即客戶類別與企業員工類別；使用案例部分有銷售作業、

預測作業、客戶需求確認、產銷協調會議與接單作業等等，以接單作業為例，接單作業主要在處理業務接到客戶所下的訂單，流程上業務需要在規定的時間內與客戶確認購買發票是否正確，再製作銷售訂單交給生產製造單位，進行產品的生產作業。因此，在這樣的情境下，可以歸納出客戶購買發票、銷售訂單、產品、時間等物件，因為購買發票與銷售訂單所紀錄的內容為產品編號、價格、日期與購買數量，可將其歸納出產品類別與時間類別，其中產品類別具有產品編號、產品價格與產品數量等資訊。

##### 4.2.2 定義類別 Attribute

類別的 Attribute 可以從流程分析中的資訊取得，歸納表 4-1 業務銷售作業的資訊，可獲得產品 Bottom Price、BOM 總成本、客戶報價、長期預測量、短期預測量、年度銷售目標、客戶短期需求量、分配量、半成品數量、維修料件分配量、加單需求量、砍單需求量、銷售訂單數量、採購訂單數量、採購發票數量、預測量、砍單數量、工單數量等資訊。以上這些資訊可定義為銷售類別所需要的 Attribute，而其它輔助觀看銷售類別的維度類別 Attribute 也可以從這些資訊中得到，以產品類別為例，客戶所下的銷售訂單是針對產品的 Model 編號，而產品經理分配的各業務員的分配量資訊，是針對產品的系列編號分類的，因此，針對產品類別的 Attribute 而言，可歸納出產品大類、產品中類、產品系列與產品 Model 等四個 Attribute。遵循上述做法，從流程分析中的流程與資訊，對應至使用案例與類別，以銷售作業為例，下面為銷售作業的類別圖。

圖 4-3 中歸納了 Customer、Employee、PM、Sales、Time、Product、Item 與 Sales Fact 等類別，其中除了 Sales Fact 以外，均繼承於 Dimension 類別，PM 與 Sales 繼承於 Employee 類別；Sales Fact 類別是用來表示業務銷售作業所需呈現的資訊，並

透過與其它維度的關係,定義出可由不同維度觀看業務銷售作業資訊的各種情境。其它的類別如 Customer、Sales、Time 與 Product,是用來表示於業務銷售作業中所需要分析的各種不同的角度,其中 Product 類別是由 Item 類別所聚合而產生的,因為在業務銷售作業中,產品是由許多原料所組成的。

在整個完整的業務銷售作業中,除了 Sales Fact 類別外,還需要加入 Inventory Fact、Shipping Fact、Promotion Fact 等類別,以完成的呈現業務銷售作業所需要管理的資訊,在 Dimension 類別方面也會增加 Hub、Warehouse、Shipping Type、Sales Type、Order Type 與 Promotion 等分析資訊的角度。在此,本研究只提出與銷售相關的資訊與維度作為範例,說明物件導向資料庫架構建立的過程。



圖 4-3 業務銷售作業類別圖

#### 4.3 Component View

Component View 將以元件的角度來呈現系統 Implementation 時的架構,本研究採用 MVC(Model、View、Controller)的概念將元件分成三個部分,如圖 4-4 所示,Viewer 是由 ReportWindows 與 DimensionList 所組成,屬於負責呈現資訊 View Layer 的元件,ReportWindows

負責呈現報表內容,DimensionList 負責呈現維度資訊,他們會接收由 Model Layer 元件所傳遞過來的結果,將報表呈現於用戶端;Sales Fact、Dimension 與 DataModel 屬於負責處理產生報表的元件,其中 Sales Fact 繼承於 ReportWindows,負責產生報表想要呈現的資訊內容,Dimension 繼承於 DimensionList,負責產生報表想要呈現的維度資訊。DataModel 是產生報表的核心機制,負責產生查詢報表的查詢句,並轉換成 XML 語法,再透過 XSL 的套用產生完整的報表資訊,最後將資訊傳遞給 Viwer 元件,負責展現報表內容。Controller Layer 的元件為 DataAccess,負責與後端資料庫聯繫,提供 Model Layer 元件與資料庫端的異動交易,因此 Sales Fact 與 Dimension 元件在處理過程中,都會與 DataAccess 元件保持互動,以支援 DataModel 元件完成產生報表的任務。

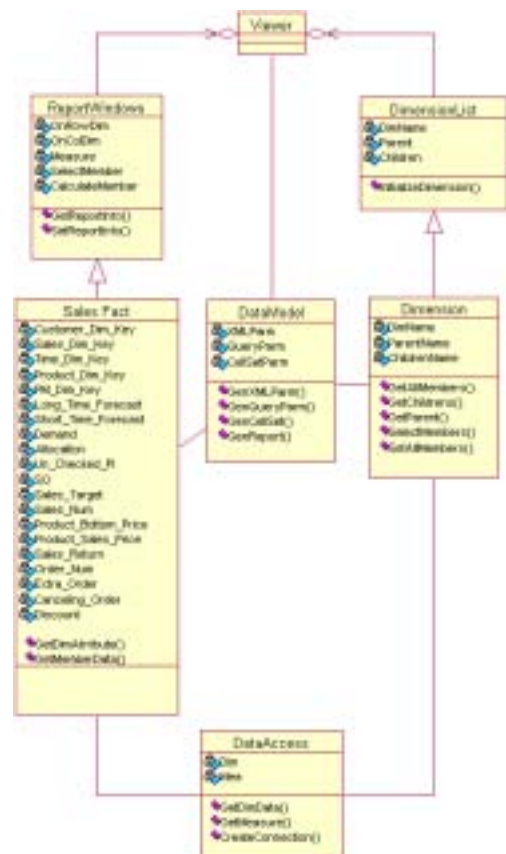


圖 4-4 業務銷售作業元件架構圖

## 五、結論

資料倉儲對於資訊的整合與應用是很重要的技術，為了能快速且彈性的提供企業分析性資訊，資料倉儲架構的設計方法是個極重要的研究問題，之前絕大多數這方面的研究都著重於關連式資料庫中建立 Start Schema 架構的探討，但隨著物件導向技術在資料庫上的應用日趨成熟，使得在物件導向概念下建構資料倉儲的研究也日益受到重視。因此本研究提出以物件導向方法設計資料倉儲，並以業務銷售作業為例，探討從流程分析所得到的結果，如何利用物件導向方法 RUP 來設計資料倉儲的研究。以解決傳統關連式資料模式下，資料倉儲架構缺乏彈性與無法分析非結構資料的問題。

在本研究中，我們提出了一個適用於業務銷售作業的物件導向資料倉儲架構，運用了物件可獨立運作與繼承的特點，提供了企業面對外部環境快去變動的趨勢下，所必須因應的彈性調整資料倉儲架構需求，讓企業能迅速的得到分析性資訊，掌握客戶需求。

## 參考文獻

1. A.H. Filho, H.A. Prado, S.S. Toscani, "Evolving a Legacy Data Warehouse System to an Object-Oriented Architecture," International Conference of the Chilean Computer Science Society (SCCC'00) November 16 - 18, 2000 Santiago, Chile, p. 32.
2. Booch, G., Jacobson, I. and Rumbaugh, J. "The Unified Software Development Process," Addison-Wesley, 1998.
3. Booch, G. et al.: The Unified Modeling Language for Object-Oriented Development, Documentation Set, Version 1.0, Rational Software Corporation, 1997.
4. Chaudhuri, S. and Dayal, U., "An Overview of Data Warehousing and OLAP Technology", SIGMOD Record, Vol.26, No. 1, 1997.
5. Chen, D. and Doumeings, G., "The GRAI-GIM reference model, architecture and methodology ", Architectures for Enterprise Integration, 1996.
6. Elmasri, R. and Navathe, S.B., Fundamentals of Database Systems, Addison-Wesley, 2000.
7. Inmon, W.H., Building the Data Warehouse, Addison-Wesley, 1992.
8. Jacobson, I. : Software Reuse – Architecture, Process and Organization for Business Success, Addison-Wesley, New York, N. Y., 1998.
9. Joryze, H. R. and Verradat, F. B., 1990 , " CIM-OSA Part 1:total enterprise modeling and function view , " Int. J. Computer Integrated Manufacturing, Vol.3 Nos.3 and 4, pp.144-156.
10. Kruchten, P.B., "The Rational Unified Process an Introduction," Addison-Wesley, 1999.
11. Rational Software Corporation, "Rational Unified Process Best Practices for Software Development Teams," 1998.
12. Rational Software Corporation, "The Ten Essentials of RUP The Essence of an Effective Development Process," 2000.
13. Rational Software Corporation, "Rational Unified Process for Systems Engineering," 2000.
14. James E. Rumbaugh: OMT: The Object Model. [JOOP 7\(8\): 21-27 \(1995\)](#)
15. Jacobson, I. (1995) "The Use-Case Construct in Object-Oriented Software Engineering." In J. M. Carroll (ed.) *Scenario-Based Design*. NY: Wiley.
16. Mayer, R. J., Benjamin, P. C., Carawaz, B. E. and Painter, M. K., "A Framework and a Suite of Methods for Business Process Reengineering", 1998, available at <http://www.idef.com/articles/framework/>.
17. Murata, T., "Petri Nets: Properties, Analysis and Application," Proceedings of IEEE, Vol. 77, No. 4, pp. 541-580, 1989.
18. Scheer, A. W., ARIS – Business Process Modelling. Third Edition, Springer-Verlag, Berlin, 2000.