

趨近泛用型之物件和物件關聯資料庫績效評估 之工作量模型

譚家蘭

政治大學資訊管理學系

黃敏男

中華電信研究所

摘 要

物件導向(Object-Oriented)與物件關聯(Object-Relational)資料庫管理系統和技術(Database Management Systems and Technology)是網際網路多媒體資料庫、工程資料庫、XML 資料庫的重要和利基的整體解決方案。績效和效能評估(Performance Measurement and Evaluation)係一項主要工具去評估和衡量物件與物件關聯資料庫技術績效和效能。在常用的效能評估方法中，例如：OO1、HyperModel、OO7、BUCKY，依然承襲傳統關聯資料庫的績效評估設計，在基本結構上缺失和不足(Structural Deficiency)有其不可突破的瓶頸和缺點：(1)一般績效評估標準只針對特定應用領域，無法考慮和表達使用者的特定需求和工作量模型(Workload Model)；(2)一般績效評估標準所發展的工作量產生器(Workload Generator)，因為只針對其特定領域作設計與測試，而失去了正確性和代表性。本研究以使用者需求導向(Requirements-Driven)，新創共同載具方法(Common Carrier)，以 ODMG 及 SQL3 標準為基礎，採用其學名結構(Generic Constructs)，形成物件與物件關聯資料庫工作量載具，同時進行物件分析(Object Analysis)、運算分析(Operation Analysis)、和控制分析(Control Analysis)，產生工作量特徵組和變數組(Workload Characterization)，以建立趨近泛用型之工作量模型(Toward A More Generalized Workload Model)和工作量產生器(Workload Generator)。本研究希望能以此工作量模型的設計使物件和物件關聯資料庫效能評估能具一致性、代表性、與正確性，解決目前效能評估方法在領域鎖定的困境。

關鍵字：物件導向資料庫、物件關聯資料庫、SQL3、ODMG、資料庫績效評估、績效評估工作量模型、工作量產生器、需求分析方法。

Toward A More Generalized Workload Model For Object And Object-Relational Dbms Benchmark Method

Jia-Lang Seng

Department of Management Information Systems, National Chengchi University

M.N.Hwang

System Engineer, Chung Hwa Telecom

Abstract

Object-oriented database management systems and object-relational database management systems have emerged as the niche resolution to the Internet data management, hypertext and hypermedia data management, and XML data management. The core of success of the niche technology rests upon the computing, storage, searching and retrieval capability and capacity. Database benchmark is the vital tool to measure and evaluate the ability and performance of object-oriented and object-relational database management systems. However the current and standard benchmarks are mostly relational benchmarks. Well-known object benchmark such as OO1, HyperModel, OO7, and BUCKY, are domain-specific and application-specific. When the user domain differs from that of the benchmark and the application scenario deviates from that of the benchmark, test results cannot be used or reproduced. In fact, in many cases, they mislead the users, management, and developers. These object benchmarks suffer from the structural deficiency of domain dependency and the lack of a more generic workload model. In this study, we present a requirements-driven and a generic-construct-oriented approach to tackle the research issues of workload characterization and compilation. We adopt two key industry standards of SQL3 and ODMG and adapt their object model, transaction model, and control model to create a more generic and systematic workload requirements development method from the user's perspective. From this new method, test results can be reused and reproduced. More realistic and representative workload can be generated. Furthermore, we have constructed a prototype to implement this new method and to demonstrate the feasibility and generalization of this new approach.

Keywords: Object-Oriented Database Management Systems, Object-Relational Database Management Systems, Database Benchmark, Workload Model, Benchmark Workload Generation, Performance Measurement and Evaluation, SQL3, ODMG, Requirements Analysis.

壹、緒論

一、研究動機

隨著網際網路的盛行，資訊的內容與呈現方式也越來越多樣化，例如：多媒體的呈現、動畫的使用等。然而，在儲存資訊的資料庫系統方面，傳統的關聯式資料庫卻因其資料結構設計的簡單性而無法因應此一資訊需求的改變。所以，物件導向技術因其在建構多媒體及複雜資料上的優點，資料庫系統開始加入一些資料結構以支援物件導向的觀念，形成物件關聯式資料庫(Object-Relational Database Management Systems, ORDBMS)。此種方式的優點保留了技術成熟的關聯式資料庫，且將資料以物件的方式來予以表達，使得物件關聯式資料庫更具包容性，而成為異質性資料庫系統的共通模型。另一種資料庫的設計完全擺脫傳統關聯式的概念，而從物件導向程式語言的觀念著手來發展資料庫，形成物件導向資料庫(Object-Oriented Database Management Systems, OODBMS)。

為了促進物件與物件關聯式資料庫的發展，標準的制訂是一個重要且迫切的工作。因此，ANSI(American National Standards Institute)/ISO(International Organization for Standardization)與 ODMG(Object Data Management Group)標準組織分別針對物件關聯與物件導向資料庫著手制訂一些標準，使得這兩個標準演變成為業界之產業標準。但是，目前這兩項標準尚未針對資料庫績效評估(Database Benchmark)進行規範，對於什麼方式的資料庫和系統在什麼樣的設定環境下，它的運算和執行是最有效率和效益？現今所公佈的一般資料庫績效評估規格皆是針對純關聯式資料庫、特定的範圍領域建構出某些設定的應用領域作為績效評估之測試依據。它們所造成的問題，除物件和物件關聯不適用，當使用者應用資料庫之範圍領域與績效評估工具之問題領域存在極大差異時，其所得到的效能可能與評估工具所得到的結果有極大的落差，目前一般的績效評估方法便無法發揮實際功能。

二、研究問題

誠如上述所提，本研究的議題分為：

1. 由於針對特定的範圍領域所設計的資料庫績效評估方法，當使用者問題領域與其不同時，並無法完全滿足使用者需求。因此，如何以使用者需求為導向，來發展趨近泛用型的工作量模型與建立工作量產生器？
2. 雖然物件與物件關聯資料庫皆支援物件導向的觀念，但其設計方式迥異。因此，在建立工作量模型時，其各自的工作量差異為何？

如何分析 SQL3 與 ODMG 標準？建立共同載具以承載使用者需求，設計物件關聯與物件導向資料庫的一般化工作量模型。

貳、文獻探討

一、物件關聯資料庫與物件導向資料庫的定義

Cattell (1994)以資料模式(Data Model)和資料庫結構(Database Architecture)兩個維度來區分資料庫管理系統。所謂資料模式，是指資料庫管理系統中，描述資料型態與操作之型態系統。而資料庫結構，則是指資料庫管理系統中，本身實行及與其他系統元件整合的方式。其以此兩個維度，對物件關聯資料庫及物件導向資料庫所做的定義，如下所述：

(一) 物件關聯資料庫

1. 資料模式：此類資料庫之資料模式是，擴增關聯資料模式(Expanded Relational Model)，使其能包含程序(Procedurality)、物件(Object)、版本(Version)與其他新功能。
2. 資料庫結構：此類資料庫之資料結構是，經由新的或延伸的資料庫查詢來提供新的功能。此結構描述程式設計者有兩個分開的環境：應用系統程式語言和延伸的資料庫查詢語言。

(二) 物件導向資料庫

1. 資料模式：此類資料庫之資料模式根源於物件導向程式語言。物件導向程式語言主要包含抽象資料型態的概念，此型態清楚地定義一個資料結構或物件之公有與私有的部分。抽象資料型態在物件導向程式語言中，稱為類別(Class)，封裝了物件之私有的資料部分與公有的程式部分稱為方法(Method)。封裝(Encapsulation)的理由，主要是想透過模組化來簡化程式的建立與維護工作。
2. 資料庫結構：此類資料庫之資料庫結構是，擴大目前的程式語言，使其提供永續(Persistence)、同時存取控制(Concurrency Control)、與其他資料庫能力。此類資料庫管理系統傳統上提供一個查詢語言，但查詢語言與程式語言都在應用系統程式環境裡執行，分享相同的型態系統和資料工作空間。

二、物件導向資料庫績效評估方法

績效評估(Benchmark or Performance Measurement and Evaluation)是一套以量化的方式來評估系統效能之方法，其常見目的為：(1)比較不同系統的效能；(2)彰顯某系統的特色所在；(3)在系統配置新硬體、修改演算法或是增加資料結構後，測試系統前後的相對績效表現；(4)衡量系統參數對系統的效能影響；(5)確認系統瓶頸所在；(6)設計某個可用來預測績效表現的模型[Gray 1993]。一般而言，用來測試效能的規則、規格、和方法，都可稱為績效評估。在績效評估測試時須建立工作量模型。所謂的工作量模型(Workload Model)，指的是在特定時段內，指定要完成的工作。而在工作量模型的建立上，可分成

兩類情形，一是硬性規定輸入的工作量之細節，另一則是使用者根據自己的需要決定所輸入的工作量。

至於在物件導向資料庫之績效評估方面，目前較知名的有：OO1、HyperModel、OO7 與 BUCKY 等。OO1 是用於衡量物件導向資料庫在工程應用上的效能表現，由於其設計簡單、容易實行，且為第一個針對物件導向資料庫所擬定的績效評估規格，而成為一個實際的標準。HyperModel 是建構在超文字的應用模型，用以更廣泛地測試在工程應用上資料庫的效能表現。OO7 評估的對象如同 OO1，其有較複雜的綱目與較繁多的運算。BUCKY 則是用於衡量物件關聯式資料庫之效能，其主要是針對物件關聯式資料庫所延伸的「物件」特質，做查詢運算測試。

(一) OO1 績效評估

OO1 的主要目的，在於衡量工程應用的資料庫作尋訪與簡單更新的效能表現，並衡量資料庫之客戶端快取對系統效能的影響[Cattell 1993]。OO1 的規格簡單且容易實作。

1. 測試資料庫

OO1 資料庫主要是由兩種物件所構成：

- (1) 元件物件(Part)：每一個元件物件有五個資料欄位：Part ID、Type、X、Y、Build date，分別代表其代碼、型態、座標 x 變數、座標 y 變數、與建構日期。每個元件物件有三個連出去的連結，以及不定數目的連進來的連結。
- (2) 連結物件(Connection)：連結物件是用來記錄元件之間的關係，每個連結物件皆有兩個資料欄位：Type、Length，分別代表連結物件的型態與長度。為了提供物件圖型裡的區域性概念，OO1 的元件物件是以其元件代碼為邏輯上的排序依據，對於連出去的物件選擇，有百分之九十的機率是選擇其相鄰的物件，這裡的相鄰是指以代碼值域內的百分之一為依據。

2. 測試運算

至於 OO1 的績效評估運算有以下三種：

- (1) 查詢運算(Lookup Operation)：隨機產生 1000 個元件代碼，並從資料庫中找出所要的元件。
- (2) 物件圖形的尋訪運算(Traversal Operation)：利用先深後廣的演算法來做七層的尋訪，以尋訪總計 3280 個互相連結的元件。
- (3) 插入運算(Insert Operation)：產生新的 100 個元件並將他們加入到資料庫中，以便觀察磁碟機 I/O 的效率。

(二) HyperModel 績效評估

HyperModel 績效評估的設計者是以超文字的應用程式模型為基礎，所發展出來更廣泛的工程資料庫效能測試。若與 OO1 相比較，HyperModel 包含比較豐富的資料庫綱目與範圍更廣的運算[Anderson et al. 1990]。不過，由於 HyperModel 僅針對超文件應用進行測試，其需求的測試運算多達數十種，因此，令使用者望之卻步，能夠產生的效益比較有限。

1.測試資料庫

HyperModel 的資料庫如 OO1 般，是由相互連結的節點所組成的。HyperModel 之節點間的關係並不似 OO1 般只有一個關係，而是其節點間存在多個關係，包括一個一對多關係 (Parent/Children)，及兩個多對多關係 (PartOf/Parts 與 RefTo/RefFrom)；亦不似 OO1 般，只有一種節點型態存在，而是存在三種節點型態包括 Node、TextNode 與 FormNode。

2.測試運算

HyperModel 包含了以下幾種不同類型的測試運算：

- (1) 完全符合的查詢：主要是以屬性的值或者 OID 作為查詢之選擇。
- (2) 範圍的查詢：包括 1%與 10%範圍的物件查詢。
- (3) 群體的查詢：從隨機挑選的物件，循著物件的關係尋訪其他相連結的物件。
- (4) 參考值的查詢。
- (5) 順序性的掃描。
- (6) 物件的尋訪：從 1：N 關係階層的第三層中，任意挑選一個物件出發，循著三個物件連接關係，作物件尋訪的運算。
- (7) 物件的編輯：包括更新字串欄位與樹節點的圖形欄位。

(三) OO7 績效評估

OO7 績效評估的主要目的是為了能廣泛地測試物件導向資料庫的效能。由於工程上的應用(CAD/CAM/CASE)大部分都會利用到物件導向資料庫的特性，因此，OO7 以 CAD/CAM/CASE 的工程應用程式為發展模型。OO7 定義了複雜物件型態，並設計了多種應用於複雜物件的運算。[Carey et al. 1993]。

1.測試資料庫

OO7 績效評估之資料庫的主要元素是合成物件(Composite Part)，所有合成物件所成的集合稱為 OO7 的設計庫，每個合成物件有以下幾個屬性：Id, Builddate, Type，分別代表合成物件的代碼、建構日期及型態，每個合成物件都有對應一個文件物件，用來記錄其文件資料，每個文件物件也有以下的屬性：Id, Title, Text，分別表示其代碼、標題及文件的內容。此外，每個合成物件還有由單元物件(Atomic Part)間所形成相對映的圖形，單元物件間的連接關係由連結物件來記錄，單元物件的屬性為：Id, Builddate, Type, X, Y, DocId, and DocType，分別表示其代碼、建構日期、型態、變數 X、變數 Y、及文件代碼，連結物件屬性為：Length, Type，分別代表其長度及型態。在設計庫上 OO7 還有一組合階層(Assembly Hierarchy)的架構，這個組合架構共有七層，最底層為基本組合物件(Base Assembly)，每個基本組合物件有如下的屬性：Id, Type, Builddate，分別表示其代碼、型態及建構日期。複雜組合物件下可接三個組合物件，基本組合物件下則接三個合成物件。

2.測試運算

OO7 所測試的運算分成三類：走訪(Traversal)、查詢(Query)、結構修改(Structure Modification)。

(四) BUCKY 績效評估

BUCKY 績效評估係針對物件關聯式資料庫的績效評估方法，它主要是查詢導向的績效評估，用來測試物件關聯式資料庫的功能，並期望能與關聯式資料庫做比較，以便瞭解物件關聯資料庫相對於關聯資料庫的技術成熟度為何。[Carey et al. 1997]。

1.測試資料庫

BUCKY 績效評估的資料庫是以大學校園資料庫為模型，設計上考慮到同時移植在純粹關聯式資料庫。

2.測試運算

BUCKY 的測試運算有五個主要類型：

- (1) 繼承(Inheritance)：資料庫的根型態為 Person_t，因此 Person_t 擁有大學資料庫內各種人員的共同屬性，其中 TA_t 是 Instruct_t 與 Student_t 的子類別，這主要是測試多重繼承的能力，資料庫內非葉結點並無實例(Instances)。
- (2) 物件間的參考(Inter-Object Reference)：物件關聯資料庫的資料定義語言能夠直接支援物件間的參考，例如：Student_t 與 Department_t 兩個型態間具有雙向關係，雖然學生/主修的關係，以 Student_t 的屬性即可清楚表達，但如此的雙向關係更能表達出物件關聯資料庫的精神。
- (3) 集合屬性(Set-Valued Attribute)：BUCKY 績效評估與關聯式資料網目的另一差異，在於它允許集合屬性值的儲存，例如：Person_t 擁有名為 KidNmaes 的屬性，其型態為 Set(Varchar(10))，表示其屬性為字串的集合，每一個自串代表一個小孩的姓名。

抽象資料型態(Abstract Data Type)：物件關聯資料庫允許被使用的函數以 SQL 語法撰寫，或者以外部的程式語言撰寫，例如：BUCKY 為了測試這方面的功能，型態 Person_t 與每個子型態皆擁有一個計算薪資的函數。

參、研究方法

本研究方法係針對目前績效評估方法的領域鎖定和測試落差的研究瓶頸，特別以新興資料庫中的物件和物件關聯資料庫為研究對象；提出以共同載具概念(Common Carrier Concept)承載使用者需求，建立以需求為導向(Requirements-Driven)的工作量模型設計方式。此種工作量需求為核心(Workload-Centric)研究方法是以物件和物件關聯的學名結構(Generic Construct)來形成(Formulation)共同載具以承載實際需求。學名結構將具有一般

性、再用性、規格性。我們採用 SQL3 規範以粹取物件關聯模型學名結構；採用 ODMG 規範以擷取物件模型學名結構。

一、SQL3 標準

(一) 簡介

SQL3 為 ANSI/ISO 為了改善傳統關聯 SQL 的不足而著手制訂的新一代之 SQL 標準。SQL3 除了加強原來 SQL-92 的語法之外，還增加了支援物件延伸方面的功能。主要包括八個部分如圖 1：

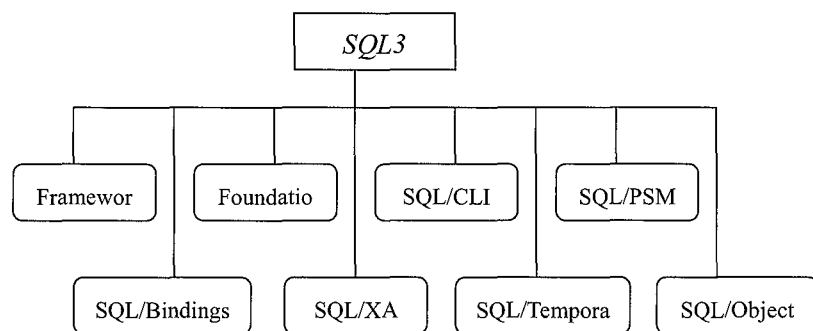


圖 1：SQL3 標準之組成圖

1. SQL/Framework：主要在簡介 SQL3 標準，提供一些一般性的概念與定義，並說明此標準的組成架構。
2. SQL/Foundation：主要在於說明其語言本身的敘述語法，包括關聯的部分和物件的部分有關抽象資料型別方面的領域。
3. SQL/CLI: CLI(Call-Level Interface)主要說明 SQL 資料庫與使用者介面語言之間的繫結架構。
4. SQL/PSM：PSM(Persistent Stored Modules)說明預儲程式的敘述，包括計算上的完整性(Computational Completeness)。SQL3 的預儲程式與使用者定義函數可以以第三代程式語言(3GL)或 SQL 來撰寫。
5. SQL/Binding：描述 SQL3 標準的語言與應用系統特定的語言之間的運作方式。
6. SQL/Transaction：主要目的是澄清 SQL 對分散式處理實行上的交易介面。
7. SQL/Temporal：描述其為了支援暫時性的資料儲存與回復所增加的語法。
8. SQL/Object: 描述特定應用系統的資料型態如何被資料庫管理系統瞭解的機制。

(二) 資料定義(Data Definition)

1. 新的內建資料型態

SQL3 在原有的資料型態之外還新增了四個內建的資料型態，簡述如下：

- (1) **LARGE OBJECT**：SQL3 為了讓資料庫能提供大型多媒體資料基本的儲取機制，因此提出支援兩種大型的物件資料型態，包括 CLOB(Character Large Object) 與 BLOB(Binary Large Object)。此兩種大型物件資料型態皆可直接儲存於資料庫中，而非外部檔案。
- (2) **BOOLEAN**：允許 SQL 能直接紀錄 True、False 和 Unknown 三個布林邏輯值。
- (3) **ARRAY**：SQL3 支援 ARRAY 的收集型態(Collection Type)，此型態允許在資料庫表格的一個欄位元裡儲存多個值的集合，而這些值是有順序性的且皆為相同的資料型態(例如皆為整數或字元型態等)。
- (4) **ROW**：由一個 ROW 的名稱和一組資料型態的順序所組成，類似一個平坦的關聯定義。

2. 抽象資料型態 (Abstract Data Type, ADT)

SQL3 對物件導向支援的部分，最基本的功能就是支援抽象的資料型態(Abstract Data Type, ADT)，或稱使用者定義型態(User-Defined Type)。所謂抽象資料型態，其主要特徵為：

- (1) 可定義擁有一個或多個屬性，每個屬性可以是任何的 SQL 型態，包括內建的型態(例如 INTEGER)、收集型態(例如 ARRAY)、或是其他抽象資料型態。
- (2) 有關其行為的所有方面都是經由方法(Methods)、函數(Functions)與程式(Procedures)來提供。
- (3) 其將屬性封裝起來，須經由系統產生的"Observer"和"Mutator"兩函數(即"Get"和"Set"函數)，分別來取出和儲存它們的值。
- (4) 若要比較它們的值，使用者需自訂一個比較規則的函數。
- (5) 可加入型態階層(Type Hierarchy)中，亦即子型態(Subtypes)除可繼承父型態(Supertypes)的所有屬性和方法外，並可增加新的屬性和方法。

3. 子型態(Subtype)

SQL3 亦包含繼承的觀念，其使用"UNDER"語法來表達型態的階層。

4. 物件(Object)

Eisenberg & Melton (1999)指出雖然 SQL3 的抽象資料型態有封裝、型態階層等種種的特質，但終究只是一個簡單的值，因為其除了提供其本身的值之外，沒有任何的身份識別。因為上述的情況，為了要允許 SQL 也能提供物件(Objects)的特質，亦即使其在一些情況下能被參考(Reference)有身份識別(Identity)的功用。

(三) 資料操作(Data Operation)

本節描述 SQL3 物件導向延伸部分，其資料的基本的操作方式。

1. 點和函數的表示法

為了能讓使用者存取抽象資料型態裡的屬性，SQL3 提供點(Dot)和函數 Functional)的表示法。

2. REF 型態的使用

SQL3 將指標的表示法(->)應用於 REF 型態的使用上。

3. 控制敘述

為了加強 SQL 語法使其如同程式語言般有計算的能力，SQL3/PSM 導入一些新的控制敘述的語法。使用者可以使用這些控制敘述來定義函數(Function)和程式(Procedure)，以及利用這些敘述，完全以 SQL 來編寫應用程式。

(四) SQL Bindings

SQL Bindings 是描述 SQL3 語言與其他應用程式語言之間的關係和運作。SQL Bindings 允許 SQL 碼內嵌於程式碼中，提供動態啟動的功能。最常見的形式是 EXEC SQL 敘述，此為執行一個內嵌的 SQL 資料操作語法。SQL Bindings 可用於建立表格、操作資料、取得 SQL 狀態、檢查與評估限制、執行 SQL 例外處理、及提出 SQL 警告訊息等。

二、ODMG 標準

ODMG(Object Database Management Group)是由許多發展物件導向資料庫的廠商於 1991 年 9 月組成，其主要的目的是著手制訂一些標準，以讓物件導向資料庫管理系統的使用者，能開發可攜式的系統也就是應用系統可以在不同的物件導向資料庫管理系統上執行，並允許不同的物件導向資料庫管理系統的產品間能相互的運作[Cattell 1994]。

ODMG 標準主要是將 SQL、OMG(Object Management Group)和物件導向程式語言標準(C++、Smalltalk 與 Java)綜合成一個標準，提供給使用者建立物件導向資料庫管理系統的應用程式。其第一個規格書是在 1993 年 8 月所發表的，稱為 ODMG-93。經過不斷地修訂與改良，其陸續發表了 ODMG-93 Release 1.2、ODMG2.0、及目前最新的版本為 2000 年 1 月所發表的 ODMG3.0。

在 ODMG 標準中，對物件導向資料庫管理系統做了簡單的定義是：「物件導向資料庫管理系統是一個資料庫管理系統，整合了資料庫能力以及物件導向程式語言能力。」ODMG 標準之主要部分包括：物件模型(Object Model)、物件定義語言(ODL)、物件查詢語言(OQL)和語言繫結(C++、Java 與 Smalltalk Language Bindings，如圖 2 所示。

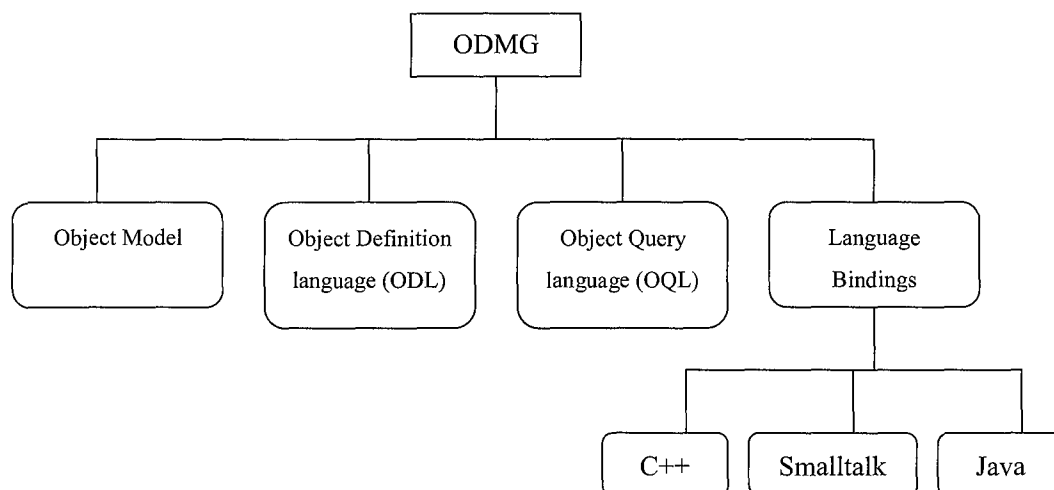


圖 2：ODMG 標準之組成圖

(一) 物件模式(Object Model)

ODMG 物件模式主要目的是提供一個共同的資料模式給支援 ODMG 標準的物件導向資料庫系統使用。ODMG 物件模式是由 OMG 物件模式擴大而來，其增加了對於定義物件與它們關係方面的功能。其主要的概念包括：

1. 最基本的模式元件為物件，每個物件有唯一的物件識別碼(OID)。
2. 物件可以被分類成不同的型態(Types)，同一型態中的所有物件具有共同的「行為」(Behavior)以及「狀態」(States)。
3. 物件中的行為被定義為一些運作(Operations)的集合，而這些運作可以被此型態的物件執行。
4. 物件中的狀態被定義為它們所攜帶的一些特性(Property)的值(Value)。這些特性指的是物件本身的屬性(Attribute)，或此物件與其他(一個或多個)物件之間的關係(Relationship)。
5. 子型態(Subtype)繼承父型態(Supertype)所定義的屬性、關係、與運作。子型態可以自己增加額外的特性與運作，也可以修正其所繼承的特性與運作。
6. 可以宣告一個範圍(Extent)，此範圍可包含一個型態的所有實例(Instances)。
7. 一個型態的個別實例可以以某種特性或一組特性的值來唯一識別。這些特性或一組特性稱為鍵，就如同在關聯式資料庫一般。

(二) 物件定義語言(Object Definition Language, ODL)

物件定義語言(ODL)是一個規格描述語言，使用來定義 ODMG 物件模式裡物件型態的規格描述。物件定義語言是使用來支援物件導向資料庫系統的資料庫綱目(Database Schema)的可攜性(Portability)。

(三) 物件查詢語言(Object Query Language, OQL)

ODMG 物件查詢語言(OQL)是一個類似 SQL 的敘述式語言，用來提供一個能有效查詢資料庫物件的環境。OQL 的設計是以 SQL-92 的查詢部分為基礎，因此其提供了所有 SQL-92 的 SELECT 語法，再增加了有關物件導向方面的語法。

(四) 語言的繫結(Language Bindings)

在 ODMG 標準中的 Java、C++ 與 Smalltalk 語言繫結是用來定義物件操作語言(Object Manipulation Languages, OMLs)，ODMG 擴充了這些程式語言使其能支援永續性的物件(Persistent Objects)。這個繫結亦包含了支援 OQL、瀏覽(Navigation)與交易(Transaction)。

三、SQL3 與 ODMG 標準之比較

根據前面 SQL3 與 ODMG 標準之探討，可知其目的地是各自為物件關聯資料庫與物件導向資料庫制訂標準。因此，此兩套標準我們以物件模式、定義語言、操作語言及語言繫結四個維度來比較它們的差異，見表 1。

表 1：SQL3 與 ODMG 之比較表

	SQL3	ODMG
物件模式	ADT：為物件之基礎，封裝一組屬性和方法，且可繼承自別的 ADT。 Object ADT：將 Table 宣告為 ADT Type Table，此種 Table 的行承襲 ADT 的屬性，而列則儲存 ADT 的實例，稱為 Object ADT，有系統給定的唯一識別碼。	Type：物件以型態(Type)來分類，同一型態的物件有共同的行為(Behavior)與狀態(States)。其亦可繼承自別的型態。 Object：型態的實例(Instance)稱為物件(Object)，有物件識別碼(OID)。
定義語言	在傳統關聯的資料定義語言(DDL)中，增加了物件延伸之資料型態的定義。	為了資料庫綱目的可攜性，ODMG/ODL 僅作規格描述，而資料操作則需透過語言繫結來達成。
操作語言	在傳統關聯的資料操作語言(DML)中，增加了處理物件型態資料的語法，並有流程控制敘述，使 SQL3 成為有完全計算能力的語言。	ODMG/OQL 僅作物件查詢之用，至於其他物件操作則需透過語言繫結功能來達成。
語言繫結	用意為提供動態執行內嵌於這些程式中之 SQL 碼的功能。目前定義了 SQL 與 C, C++, Basic, Ada, COBOL 之間的繫結機制。	用意為定義物件操作語言(OML)，擴充這些程式語言有資料操作之能力。目前定義了 Java, C++, Smalltalk 語言繫結。

資料來源：本研究整理

肆、研究設計

一、研究架構

本研究設計係以分析物件關聯資料庫的標準(SQL3)與物件導向資料庫的標準(ODMG)來著手，分別整理出物件關聯資料庫與物件導向資料庫的工作量特徵。然後再利用所分析的工作量特徵模型，建立相對應的工作量產生器系統，以提供使用者導向之資料庫的績效評估方法。我們將建立物件與物件關聯資料庫工作量模型設計和產生器程式分成兩個階段。在工作量模型建立階段：此階段的主要目的是分析物件與物件關聯資料庫工作量模型的特徵，以作為我們工作量共同載具的基礎。我們將工作量模型依照不同觀點區分成三個部分：(1)物件模式(Object Model)：主要探討被物件與物件關聯資料庫所存取的物件。(2)交易模式(Transaction Model)：主要探討物件與物件關聯資料庫用來運作物件的方法。(3)控制模式(Control Model)：主要探討績效評估實驗測試時的控制要求。在工作量產生器雛形階段：此階段則是以前一階段所分析的工作量模型為基礎，依據系統與資料庫平臺的種類，產生相對應的程式碼，建構出工作量產生器系統，以便讓使用者利用此系統依照實際需求進行實驗。工作量產生器系統分成五個模組元件如圖 3 所示。(1)物件產生器(Object Generator)：針對工作量模型的物件模式產生測試資料庫。(2)交易產生器(Transaction Generator)：針對工作量模型交易模式產生測試交易。(3)交易排程器(Scheduler)：是用來排定績效測試時，交易順序與次數。(4)測試驅動器(Test Driver)：則是依據控制模式中的交易順序與交易次數等控制參數，使這個資料庫績效評估系統運作起來，並產生資料庫之測試評估數據。(5)測試結果蒐集器(Results Collector)：則是蒐集並儲存此系統的測試結果。

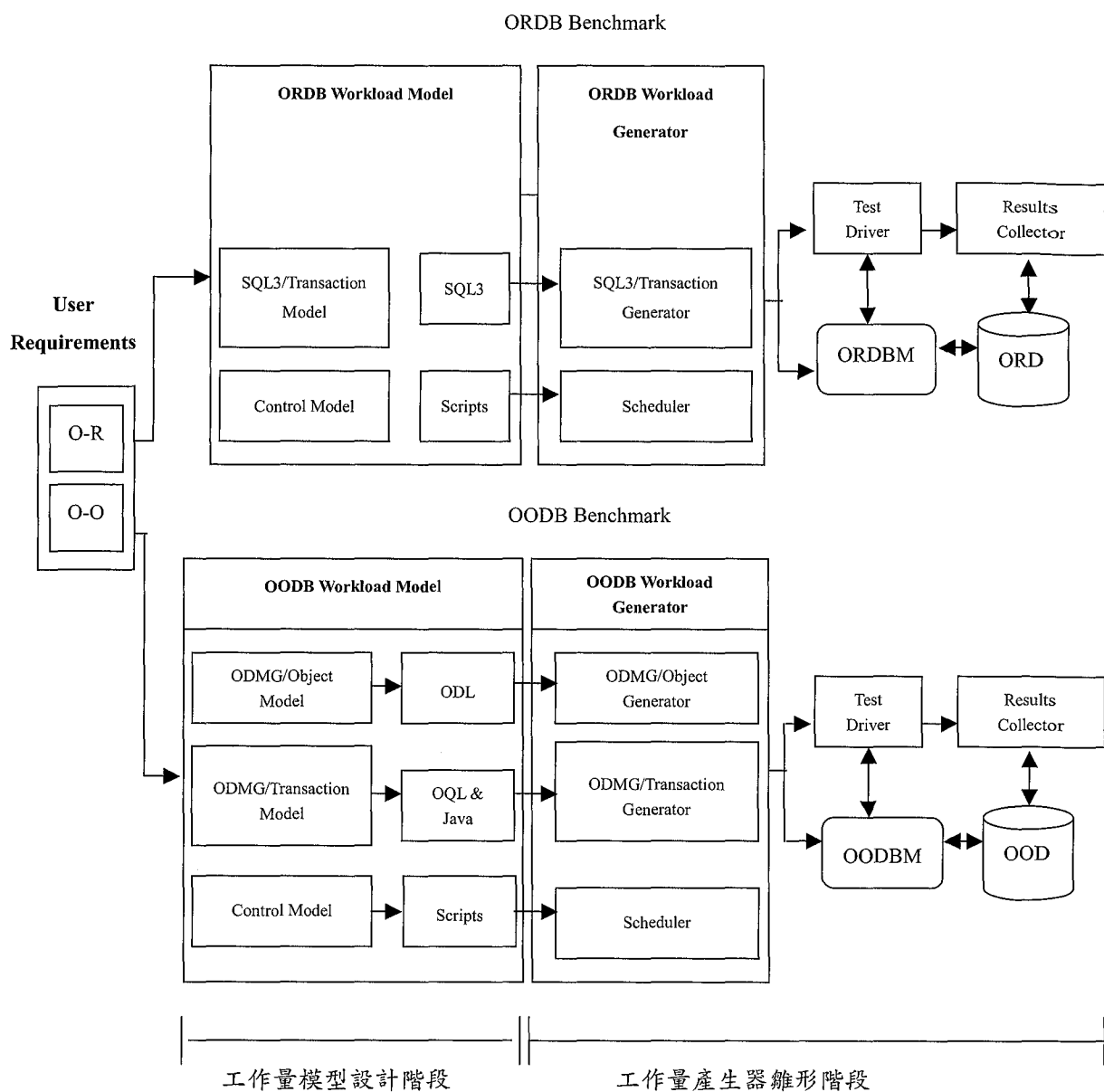


圖 3：本研究之研究架構圖

二、工作量模型設計階段

此階段我們以 SQL3 標準來分析物件關聯資料庫工作量的特徵；而以 ODMG 標準來分析物件導向資料庫工作量的特徵。下面我們就分別描述此兩種資料庫的工作量模型。

(一) 物件關聯資料庫工作量模型

1. SQL3/Object Model 物件模式

在這個模式中，我們主要目的是要探討 SQL3 標準的物件延伸部分。SQL3 對物件方面的支援，最主要功能是提供「抽象的資料型態」(Abstract Data Type, ADT)。抽象的資料型態封裝了一組屬性與方法。此外，可宣告 ADT Type Table 用以儲存此 ADT 的實例，稱為 Object ADT，並由系統給定唯一的物件識別碼。我們將 SQL3/Object Model 整理如表 2。

表 2：SQL3/Object Model 之整理

SQL3/Object Model	說明
新增的內建資料型態： 1. BOOLEAN 2. BLOB, CLOB 3. ROW 4. ARRAY	SQL3 目前增加了四種內建的資料型態，分別為： 1. BOOLEAN：包含 true, false, unknown 三個布林邏輯值。 2. BLOB, CLOB：分別用以儲存二元大型物件與字元大型物件，主要目的是支援多媒體資料的儲存。 3. ROW：由一個名稱和一組資料型態所組成。 4. ARRAY：為一種收集型態，用以儲存一組有順序之相同資料型態的值。
ADT(抽象的資料型態)	1. 封裝一組屬性和行為。 2. 屬性型態可為內建的資料型態、收集型態與 ADT。 3. 行為由 Function 與 Procedure 來建構。 4. Object ADT：Object ADT 儲存於 ADT Type Table，有系統給定的物件識別碼(OID)。 5. 可繼承父型態(Supertype)的屬性與行為。
REF(參考型態)	用以表達 Object ADT 的關係，其值為 OID。

資料來源：本研究整理

2. SQL3/Transaction Model 交易模式

SQL3 為了操作這些物件延伸方面的資料型態，支援了點(.)、函數(Function())與指標(->)的表示法，以讓使用者在 SQL DML 語法中使用來操作這些資料。

3. 控制模式(Control Model)

在這個模式中，主要目的是要探討本研究資料庫績效評估測試實驗進行時的控制要求。我們探討了六個控制要素，如下所述：

- (1) 測試期間(Test Duration)：測試期間為需在系統達到穩定狀態之後才開始，並且需持續不中斷一段時間以完成測試並記錄結果。
- (2) 測試順序(Test Sequence)：即測試時交易的執行順序。
- (3) 穩定狀態(Steady State)：當測試的機器在不穩定狀態的情況下(如機器尚未暖機完畢等)就進行測試，可能會產生一些失真的測試數據。
- (4) 績效指標(Performance Metric)：績效標竿為績效評估比較的依據，通常都是根

據此績效評估的目的來設計績效標竿。一般常見的績效標竿有系統回應時間(Response Time)及效能產出(Throughput)等。

- (5) 測試模式(Test Mode)：在績效評估測試時，我們可根據快取記憶體的情形，分成冷模式、溫模式、熱模式三種測試模式。
- (6) 測試重複次數(Number of Repetition)：即表示測試時交易被執行的重複次數。

(二) 物件導向資料庫工作量模型

1. ODMG/Object Model 物件模式

根據前面的文獻探討可知，ODMG Object Model 之主要特徵為，物件是以型態來分類，同一型態有相同的行為與狀態。型態可繼承而構成了型態階層。物件為型態的實例，有唯一的物件識別碼。表 3 中，我們整理了 ODMG Object Model 之物件特徵，並列示 ODMG/ODL 之語法。

表 3：ODMG/Object Model 之整理

ODMG Object Model	ODMG ODL
1.物件為一個型態的實例，有狀態、行為和身份識別。 2.物件以 OID 做為身分識別。 3.物件可以有名稱。 4.物件可以有鍵(key)值。	<pre>Interface type_name [:inheritance_list] (extent extent_name key(s) key_list) { attribute domain_type attribute_name relationship target_path inverse traversal_path };</pre>

資料來源：本研究整理

2. ODMG/Transaction Model 交易模式

ODMG 的 ODL 只用來定義資料綱目，而沒有提供資料操作的語法。而 OQL 則純粹只提供查詢資料的功能。

3. 控制模式(Control Model)：如同 SQL3 的控制模式。

三、工作量產生器的雛型階段

(一) 物件產生器(Object Generator)

物件產生器的主要目的，是根據物件模型產生測試運算處理的資料。工作量模型的資料庫綱目之複雜度與資料庫之效能息息相關，因此若能正確地表達資料庫綱目與運作資料庫的大小，將是工作量模型能否具代表性與正確性的主要因素。因此，為了使測試資料庫能最真實地反應使用者實際的資料庫情況，我們設計本研究的物件產生器原則為，可讓使用者完全根據其實際需求來建構測試工作量的資料庫。其運作方式如圖 4 所示。

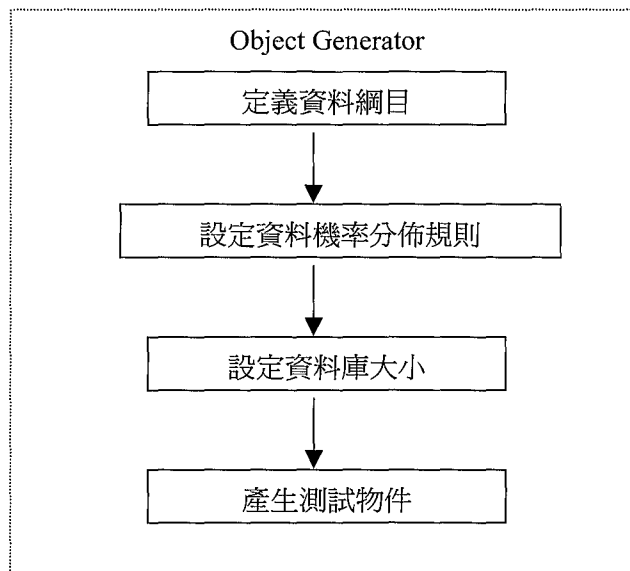


圖 4：Object Generator 之運作方式

1. SQL3/Object Generator 物件產生器

根據前面 SQL3/Object Model 分析的物件模式，我們描述 SQL3 物件產生器裡的基本要素 ADT 建置時之表示法，如下所述。值得注意的是，這些要素裡的元件，還需視測試的資料庫管理系統有無支援，而做適度的調整。ADT 抽象資料型態的表示法：

- Type name：指定此型態的名稱。
- Supertype name：若直接繼承時，需指定其所要繼承的父型態名稱。
- Attribute：指定此型態包含哪些屬性，其表示法為：
- Operation：宣告此型態包含哪些方法，其表示法為：

2. ODMG/Object Generator 物件產生器

我們描述 ODMG 物件產生器裡的基本要素 Type 建置時之表示法。

(二) 交易產生器(Transaction Generator)

交易產生器主要目的是產生測試資料庫運算的相關程式碼。同樣地，為了使績效評估能真正測試使用者所要測試的交易，我們設計本研究的交易產生器原則為，可讓使用者完全根據其實際需求來建構工作量的測試交易。

(三) 交易排程器(Scheduler)

根據前面控制模式之探討，我們建立交易排程元件讓使用者排定績效測試時之交易順序與次數。其運作方式如圖 5。

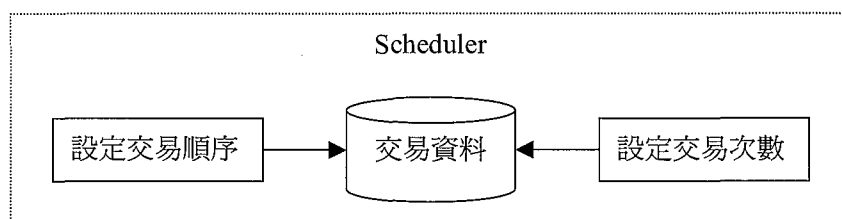


圖 5：Scheduler 之運作方式

(三) 測試驅動器(Test Driver)

測試驅動器是績效評估系統的核心，負責統籌績效評估測試之運作執行。其主要功能其運作方式如圖 6：

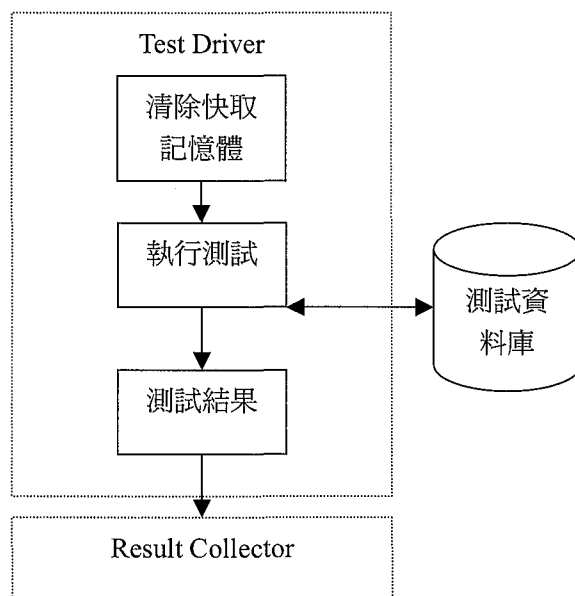


圖 6：Test Driver 之運作方式

伍、雛形系統設計

本研究除了建構工作量之一般性發展程序與方法，還依照此模式，實際建構了工作量產生器的雛形系統，用以證實趨近泛用型之領域獨立性和不鎖定性。

一、簡介

本研究實作雛型採用 Oracle9i 和 Jasmine 資料庫系統，選用 Java 為開發語言。

二、系統架構與功能

我們依照前面所提出的工作量模型來設計我們的雛形系統。因此，本雛形系統可分為五大功能，分別為物件產生器(Object Generator)、交易產生器(Transaction Generator)、交易排程器(Scheduler)、測試驅動器(Test Driver)與結果蒐集器(Result Collector)，如圖 7 說明本研究之系統程式結構。

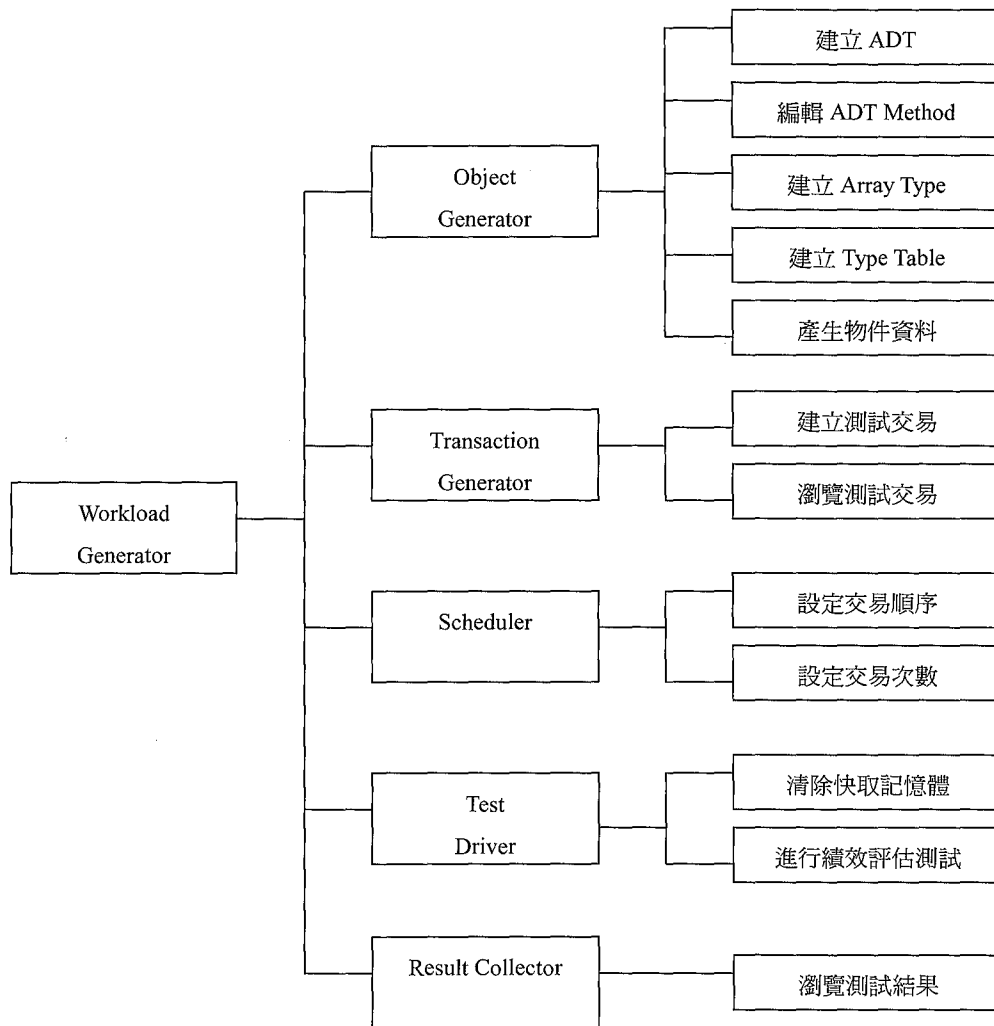


圖 7：本研究之系統程式結構圖

(一) 物件產生器(Object Generator)

物件產生器的主要目的，是讓使用者依據自己的需求來建立物件型態及設定資料值機率分佈，然後再依此幫使用者產生物件資料，以構成測試資料庫。此功能中包含了五個子功能，分別為：

1. 建立 ADT

此功能是讓使用者建立抽象資料型態，使用者需輸入型態名稱、屬性、和方法。
如圖 8 所示。

Create ADT Type

物件型態名稱:

Attribute

屬性名稱	屬性型態	其它型態名稱	變數1	變數2
CustNo	INTEGER		1	20
CustName	CHAR(10)		10	
address	ADT	addressType		
phone	ARRAY	array1		
salesRep	REF	salesType	sales	

Method

```
member function showname return char
```

圖 8：ADT 建立完成畫面

2. 編輯 ADT Method

在前面建立 ADT 功能裡面，我們先宣告此 ADT 有哪些方法，然後在用此功能編輯這些方法的運作。使用者可以 SQL、流程控制敘述等，來撰寫方法裡運作的語法，其編輯畫面如圖 9 所示。

Edit Method

Object Type Name:

Method Statement:

```
member function showname return char is
begin
    return self.CustName;
end showname;
end;
```

圖 9：ADT Type Method 編輯畫面

3.建立 Array Type

建立一 Array Type，使用者須指定型態名稱、Array 長度、資料型態，這裡的資料型態只支援 INTEGER 與 CHAR(10)，同樣地使用者指定完資料型態後需將資料值機率分配填入變數 1 與變數 2。如圖 10 所示。

Create Array Type

型態名稱：	array1
Array長度：	10
資料型態：	INTEGER
變數 1：	1
變數 2：	10

圖 10：建立 Array Type 之畫面

4.建立 ADT Type Table

與 SQL3 標準相同，在 Oracle 中，我們需建立 ADT 的 Type Table，以用來儲存 Object ADT，且 Oracle 會在每一列 Object ADT 新增時，自動指定一個 OID 值給此物件。其編輯畫面如圖 11 所示。

Create Type Table

Table name:	Customers
Type name:	CustomerType

圖 11：建立 Type Table 之畫面

5.產生物件資料

此功能的目的是依據此 ADT 的資料值機率分佈之設定來產生 Object ADT 的值。使用者需依次輸入測試編號、測試名稱、此 ADT Type Table 名稱、此 ADT 名稱、產生的實例筆數，然後按下 Submit 按鈕。如圖 12 圖 13 所示。

Benchmark Result

測試編號：a1

測試名稱：insert into Customers

測試順序	測試次數	交易名稱	回應時間(單位:毫秒)
1	1	insert objects	891

10 Objects have been added into Customers Type Table

圖 12：產生物件資料之設定畫面

Generate Objects

測試編號：	a1
測試名稱：	insert into Customer
ADT Type Table Name:	Customers
ADT Type Name:	CustomerType
Number of Instances:	10
Submit Reset	

圖 13：產生物件資料之完成畫面

(二) 交易產生器(Transaction Generator)

交易產生器的主要目的是讓使用者自訂其測試資料庫運算的相關程式碼。其分為建立測試交易和瀏覽測試交易兩個功能。

1. 建立測試交易

此功能的主要目的是讓使用者新增一筆測試交易。使用者需輸入此交易的名稱及此交易的 SQL 語法，然後按下 Submit 按鈕之後，系統會將此筆測試交易存入 Transaction Table 中。如圖 14 所示。

Add Transaction

交易名稱:

SQL statement:

圖 14：建立測試交易之畫面

2. 瀏覽測試交易

此功能的主要目的是提供給使用者瀏覽全部的測試交易。其畫面如圖 15。

Display Transaction

交易順序	交易次數	交易名稱	SQL Statement
1	2	use ADT Function	select c.showname() from Customers c
2	2	select Array Type	select c.phone from Customers c
3	2	select REF Type	select c.salesRep from Customers c
4	1	update Customers	update Customers set CustName='AAAAAAAAAA' where CustNo=1

圖 15：瀏覽測試交易之畫面

(三) 交易排程器(Scheduler)

交易排程元件即是用來排定測試交易的順序與次數。

1. 設定交易順序

此功能是用來設定交易的測試順序。我們設計了“目前順序”及“新的順序”兩個 List，使用者只需依次將交易指定至“新的順序”List 中，然後按下“儲存更改結果”按鈕，即可完成新的交易順序的設定。系統會據此更新 Transaction Table 中交易順序的設定。如圖 16 所示。

設定交易順序

目前順序		新的順序
use ADT Function select Array Type select REF Type update Customers	Copy >>> remove <<<	
儲存更改結果		

圖 16：設定交易順序之畫面

2. 設定交易次數

此功能主要目的是讓使用者設定每筆交易在測試時的重複次數。使用者只需設定其欲更改的交易次數，然後按下 Submit 按鈕，即可完成設定。如圖 17 所示。

設定交易次數

交易順序	交易名稱	交易次數
1	use ADT Function	2
2	select Array Type	2
3	select REF Type	2
4	update Customers	1

Submit

圖 17：設定交易次數之畫面

(四) 測試驅動器(Test Driver)

此功能的主要目的即是根據前面使用者所定義的測試交易、及交易順序、交易次數等控制參數，來執行資料庫績效評估的測試，並儲存測試結果。系統會傳回測試結果如圖 18 圖 19 所示。

Execute Benchmark Test

請輸入此測試的基本資料...

測試編號：

測試名稱：

請選擇要測試的交易名稱...

☒ use ADT Function

☒ select Array Type

☒

☐ update Customers

圖 18：執行績效測試之設定畫面

Benchmark Result

測試編號： a6

測試名稱： Test Function, Array, and REF

測試順序	測試次數	交易名稱	回應時間(單位:毫秒)
1	2	use ADT Function	441
2	2	select Array Type	10
3	2	select REF Type	10

圖 19：執行績效測試之結果畫面

(五) 測試結果蒐集器(Results Collector)

此功能的主要目的是提供給使用者瀏覽績效評估測試的結果之用。如圖 20 和圖 21 所示。

Display Result

請選擇欲瀏覽的測試名稱：

圖 20：瀏覽測試結果之選擇畫面

Benchmark Result

測試編號：a1

測試名稱：insert into Customers

測試順序	測試次數	交易名稱	回應時間(單位:毫秒)
1	1	insert objects	891

圖 21：瀏覽測試結果之傳回畫面

陸、結語與未來研究方向

一、結語

物件導向(Object-Oriented)與物件關聯(Object-Relational)資料庫管理系統和技術(Database Management Systems and Technology)是網際網路多媒體資料庫、工程資料庫、XML 資料庫的重要和利基的整體解決方案。績效和效能評估(Performance Measurement and Evaluation)係一項主要工具去評估和衡量物件導向與物件關聯資料庫技術績效和效能。在常用的效能評估方法中，例如：OO1、HyperModel、OO7、BUCKY，依然承襲傳統關聯資料庫的績效評估設計，在基本結構上缺失和不足(Structural Deficiency)有其不可突破的瓶頸和缺點：(1)一般績效評估標準只針對特定應用領域，無法考慮和表達使用者的特定需求和工作量(Workload Model)；(2)一般績效評估標準所發展的工作量產生器(Workload Generator)，因為只針對其特定領域作設計與測試，而失去了正確性和代表性。本研究以使用者需求導向(Requirements-Driven)，新創共同載具方法(Common Carrier)，以 ODMG 及 SQL3 標準為基礎，採用其學名結構(Generic Constructs)，形成物件與物件關聯資料庫工作量載具，同時進行物件分析(Object Analysis)、運算分析(Operation

Analysis)、和控制分析(Control Analysis)，產生工作量特徵組和變數組(Workload Characterization)，以建立趨近泛用型之工作量模型(Toward A More Generalized Workload Model)和工作量產生器(Workload Generator)。本研究希望能以此工作量模型的設計使物件和物件關聯資料庫效能評估能具一致性、代表性、與正確性，解決目前效能評估方法在領域鎖定的困境。

本研究之雛形設計以物件關連和純物件之資料庫績效評估工作量為核心，在設計上完整分析、運算分析、控制分析的模型要素。但是，在實作系統上，資料型態有限；運算機制有限，是本雛形系統的限制。不過，資料型態上一般資料型態(Simple Type)定義在 SQL3 和 ODMG 的部份都已實際建立，但是，複雜型態(Complex Type)目前實作 BLOB 及 ARRAY TYPE。在運算控制部份，大多關連查詢和物件查詢均已建立，但是深度尋訪(In-Depth Traversal)及複雜物件參考(Object Reference)則未能建置。與目前標準領域導向的績效評估方法比較，例如 BUCKY、007、HyperModel、001，在工作量模式設計上是一項超集合並且符合需求性和開放性，提出解決領域設定設計的瓶頸，形成採用 SQL3 與 ODMG 標準模型形成共同載具之學名結構，在雛形產生器建置上較 BUCKY 和 007 為簡單，主要差異在於機制分佈種類有限，產生資料型態簡單和一般性之查詢限制，這是本研究待繼續努力之處。

	本研究	BUCKY	007	HyperModel	001
物件關聯模式					
ADT	V	V			V
Table	V	V			V
Reference	V	V			V
BLOB	V	V			
物件模式					
Interface	V		V	V	
Type	V		V	V	
Subtype	V		V	V	
OID	V		V	V	
查詢模式					
單一查詢	V	V		V	V
聯合查詢	V	V		V	V
範圍查詢	V	V		V	V
順序查詢	V	V		V	V
尋訪查詢					
鬆散尋訪	V		V		
緊密尋訪			V		
路徑尋訪			V		
更新查詢		V	V		
控制模式	V				

二、未來研究方向

我們發現擴充建立物件與物件關聯資料庫工作量模型和產生器的研究工作可以朝下列方向繼續進行：

1. Internet 網站的 Web pages 之績效探討：Internet 網站的 Web pages 有著複雜的連結構造，此構造是否能用物件導向資料庫來作有效管理，而若要幫助使用者測試這樣構造的資料庫效能，應如何測試。
2. 大型資料庫之效能表現：本研究建立資料庫測試的雛形系統，未來使用者可以此為基礎，將工作量加大以測試並探討大型資料庫之效能表現。

在 Internet 環境中，資料庫管理系統的交易機制可能有鎖定(Lock)、回復(Rollback)、或版本控制(Versions Control)等機制。因此，未來研究將在 Internet 環境中的進階的績效評估作更深入之探討。

參考文獻

1. Anderson, T., A. Berre, M. Mallison, H. Porter and B. Schneider, "The HyperModel Benchmark," *Proceedings of the 2nd International Conference on Extended Database Technology*, March 1990, pp. 317-331.
2. Barry, Douglas and David Jordan, "ODMG: The Industry Standard for Java Object Storage," *Component Strategies*, September 1998.
3. Carey, M. J., D. J. DeWitt and J. F. Naughton, "The OO7 Benchmark," *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, May 1993, pp. 12-21.
4. Carey, M.J., D.J. DeWitt, J.F. Naughton, M. Asgarian, P. Brown, J.E. Gehrke and D.N. Shah, "The BUCKY object-relational benchmark," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, May 1997, pp. 135-146.
5. Cattell, R. G. G. and J. Skeen, "Engineering Database Benchmark," *ACM Transactions on Database Systems*, 17(1): 1-31, March 1992.
6. Cattell, R. G. G., "An Engineering Database Benchmark," appears in The Benchmark Handbook for Database and Transaction Processing Systems, Ed. by Jim Gray, Morgan Kaufmann, Inc., 1993, pp. 397-434.
7. Cattell, R. G. G., Object Data Management: Object Oriented and Extended Relational Database System, Addison Wesley Publishing Company, Inc., 1994.
8. Cattell, Rick and Douglas Barry, ODMG 2.0, Dr. Dobb's Sourcebook, 2000.
9. Chaudhri, A.B., "Object database management systems: an overview," *BCS OOPS Newsletter*, No. 18 (Summer 93), pp. 6-15.

10. Chaudhri, A.B. and N. Revell, "Benchmarking object databases: past, present & future," *Proceedings of the Seminar on Object-Oriented Databases: Realising their Potential and Interoperability with RDBMS*, London, UK, May 1994, pp. 166-183.
11. Date, C.J. and Hugh Darwen, A Guide to the SQL Standard, Reading, Mass.: Addison-Wesley, 1999.
12. Davis, Judith R., "Extended Relational DBMSs: The Technology, Part 1," *DBMS*, June 1997(1).
13. Davis, Judith R., "Universal Servers: The Players, Part 2," *DBMS*, July 1997(2).
14. DeWitt, D. J., P. Fittersack, D. Maier and F. Velez, "A Study of Three Alternative Workstation Server Architectures for Object-Oriented Database Systems," *Proceedings of the 16th International Conference on Very Large Data Bases*, August 1990, pp.107-121.
15. Eisenberg, Aufsatz A. and J. Melton, "SQL:1999, formerly known as SQL3," *ACM SIGMOD Record*, 1999 (1), March 1999.
16. Fortier, Paul J., SQL-3: Implementing the Object-Relational Database, McGraw-Hill, 1999.
17. Gray, J. N., The Benchmark Handbook for Database and Transaction Processing Systems, Ed., Morgan Kaufmann, Inc., 1993.
18. Jepson, Brian, "What's Inside the ODMG-93 Standard?," *DBMS*, July 1997.
19. Khoshafian, S. and Abnous, R., Object orientation: concepts, languages, databases, user interfaces, John Wiley & Sons, New York, 1990.
20. Kim, W., Introduction to object-oriented databases" *MIT Press*, Cambridge, 1990.
21. King, Nelson H., "The Object Database Goes Online," *DBMS*, January 1997.
22. King, Nelson H., "Java in the Database Server," *DBMS*, June 1998.
23. King, Nelson H., "ODBMS: Now or Never," *DBMS*, July 1998.
24. Mattos, N. M. and J. Melton, "An Overview of the Emerging Third-Generation SQL Standard (Tutorial)," *SIGMOD Conference*, 1995, p.468.
25. Melton, J. and A. Simon, "Understanding the New SQL:A Complete Guide," 2nd Ed., *San Mateo*, CA: Morgan Kaufman, 2000.
26. Melton and Jim, "SQL Language Summary," *Computing Surveys*, Vol. 28(1), 1996, pp.141-143.
27. Sawyer, T., "Doing Your Own Benchmark," appears in The Benchmark Handbook for Database Transaction Processing System, Ed. by Jim Gray, Morgan Kaufmann, Inc., 1993, pp.543-562.
28. Zorn, B.G. and A.B. Chaudhri, "Object database behavior, benchmarks, and performance," *Addendum to the Proceedings of the ACM International Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA '95)*, Austin, Texas, October 1995, pp. 159-163.