UML導向之主從式物件資料庫績效評估之工作量模型

諶家蘭、方相宜
政治大學資訊管理學系

摘要

物件資料庫技術自九十年代帶領資訊科技和網路科技進入整合和連結世代，隨著二十一世紀來臨，資料庫科技不斷演進，接受新世紀挑戰和洗禮。自結構化資料庫到平結構超文件資料庫到少結構化多資訊資源異質存取資料庫，再由資料倉儲、資料發掘到多媒體網路資料庫、XML 文件資料庫和元件資料庫。我們都可以看到物件資料庫的應用和重要性，他們使得網路平台和交易平台能夠以資料獨立、流程獨立和平台獨立方式和分散式運算來達成。新興科技的成敗在技術面、管理面和市場面面面重要，如何提出比較客觀、公允、一致、正確、實用、量化的評估機制是非常重要的。績效評估自 1960 年代開始有系統性被開發，學界和業界投入大量心力研發改進，使得績效評估成爲最重要最有力的評估機制。資料庫領域中，績效評估一直扮演關鍵決定因素，而其中最有價值的是工作量模型的研究和開發。在物件領域中，主流和標準績效評估方法一直以問題領域設定和問題領域專屬爲工作量模型的基礎，所產生的績效數據，一直爲使用者、管理者、開發者的重要參考。但是，長久以來，使用者需要重新客製，才能眞正使用標準數據。本研究期望針對研究瓶頸，提出以使用者工作量需求爲導向，使用標準載具，透過物件需求分析方法，承載資料需求、交易需求、控制需求，將不同使用者需求，以系統化和模式化，先以圖形型態 UML 方法，再轉換成爲文字型態語法語言，透過產生器和驅動器自動產生使用者測試套組。

關鍵詞：物件技術、物件資料庫、主從式架構、績效評估、工作量模型、 UML

An UML-driven C/S OODBMS Workload Model for Benchmark

Jia-Lang Seng, Ian-I Fong
Department of Management Information Systems, National Chengchi University

ABSTRACT

Object-oriented database technology enables the integration of information technology and Internet technology. It facilitates the new and emerging development of data warehousing, data mining, multimedia network systems, XML document technology, multi-source Web information access, interoperability of heterogeneous Web databases, and mobile databases. Performance evaluation and assessment plays the vital role of the success of these new and emerging technologies. It is a scientific and quantitative method to analyze and assess the strong points and weak points of technology since 1960s. In the database area, benchmark is the name of performance study. Standard and major benchmark methods are domain specific and domain dependent. Users, managers, and developers use these benchmark results as a reference to make decisions on troubleshooting, procurement, breakthrough, and bottleneck detection. However, a re-occurring problem of using these benchmark results is that these statistics are only estimates. How close these numbers are to the actual performance depends on how close the standard workload is to the actual applications. When your problem domain differs from the standard domain, performance results vary and become misleading and meaningless. Workload is the core issue of the re-occurring problem. In this research, we perceive workload model as a systematic modeling of user requirements in applications. The process of workload requirements analysis and modeling becomes the core of a benchmark study. The workload model can be more open and more generalized. We adopt UML as the diagramming analysis technique. We develop corresponding textual and grammatical representation. From analysis, specification, to generation and execution, we present a coherent and cohesive requirements-driven benchmark model that is more representative, correct, and real.

Keywords : Object-Oriented Technology, Object-Oriented Database Management Systems,Client/Server Structure, Performance Evaluation and Assessment, Benchmark, Workload Model,UML

壹、緒論

一、研究背景

資料庫的技術隨著電腦及網路科技不斷演進，使用的目的、方法及呈現方式都與過去不同。以往廠商及使用者大都以關連式資料庫（Relational Database Management Systems, RDBMS）爲主，經過長時間的發展，關連式資料庫在可靠性上有相當不錯的表現，但是由於應用上逐漸複雜，使得關連式資料庫不免有些缺失。例如，單純的表格方式無法完全表達眞實世界的實體及實體間特殊性與一般性的關係，複雜的型別無法充分顯示，在工程上大量複雜物件的使用，其效能表現太差等，都是導致物件導向技術發展的原動力，大大地改變了很多傳統的做法。現代使用者的需求是多樣與無限的，使用者希望系統是根據他們的觀點及他們建構眞實世界的方式來設計，資料庫的技術和儲存的內容也隨之多樣化。然而，物件導向技術（Object-Oriented Technology, OOT）在建構多媒體及複雜資料的優點，使得物件導向成爲 90 年代以後重要的研究方向，物件導向資料庫（Object -Oriented Database Management Systems, OODBMS）和物件關連式資料庫（Object-Relational Database Management Systems, ORDBMS）仍是新興資料庫科技議題之一。

二、研究問題

不論是物件導向資料庫或是物件關連資料庫都需要經過績效評估來證明新技術的有效性和改良性。定性方法證明物件導向技術優於關連導向技術，可以自【OMG 1997】【Cattell 1994】【ANSI 1999】標準文件發現，眞實世界許多物件的複雜型態、複雜關係和複雜運算，關連技術在結構上無法處理。在電腦輔助設計、電腦輔助軟體工程、電腦輔助軟體工程、電腦輔助製造的應用上，關連技術無法提供足夠功能完成工作。定量方法證明關連技術在短交易型態、一般基本資料型態、大量商業基本運算、和表格式資料處理上表現優異。而物件技術在多維度分析、圖形處理、多媒體資料型態、長交易、多處理、眞實世界實體完整抽象化上表現突出。所以，九十年代以後，網際網路技術（Internet Technology）和行動運算技術（Mobile Technology）使得物件技術被看重和使用。例如元件導向開發、行動商務、 JAVA 與資料庫跨網際網路平台連接、多網際網路資訊資源存取和整合、超文件和超媒體資料庫，都是物件技術應用例子。因此，如何公允、客觀、正確、一致來評估物件技術和資料庫技術結合非常重要。更重要是能夠針對每一位使用者的應用系統、應用環境和應用平台，評估物件技術和資料庫技術的優缺點。定性研究方法在績效評估領域著重了解新技術的功能有哪些？對管理者、使用者、開發者的影響爲何？使用新技術在品質、效能、方法、流程、工作上有什麼影響？定量研究方法屬於工程方法、實驗和實作研究方法，以量化方法表達新技術的效能，以量化方法來比較新技術是否優於既有的技術。量化方法通常觀察（Observed）、設定（Set）、或隨機（Randomized）以便合成的資料來設計實驗中的各項變數。

本研究將採用定量研究方法，以探討物件資料庫的工作量模式是應該如何塑模？探討如何改進目前物件技術績效評估方法瓶頸？使得合理、正確、一致、客觀評估可以適用使用者他們的應用系統、應用環境和應用平台。管理者最常用來衡量電腦系統績效的方法不外兩者：1. 績效評估 (Benchmark) 2. 數學模式 (Mathematical Modeling and Analysis)。典型的績效評估都是由一組程式 (Program) 或互動式的語言 (Scripts) 來產生人工的工作量 (Artificial Workload)，來表示系統的預期運算。數學績效模式用來做長期的容量 (Capacity) 規劃是相當好的工具【 Dey and Seidmann 1994 】，績效評估在相對於數學模式而言需較多的時間和金錢，有些項目很難用模式來建構的，如安全性、信賴度、及使用者親和度等，雖然，績效評估在模擬具有代表性情境的測試，以及實作應用系統在資料庫實際運算的速度兩方面優於數學模式。但是現今所公布的資料庫績效評估規格皆是針對特定的範圍領域 (Domain Specific or Domain Dependent or Domain Proprietary) 建構出合成的工作量 (Synthetic Workload) 作爲績效評估的測試依據。形成若使用者實際使用資料庫時的問題領域與績效評估所設定的領域存在差異時，結果會與績效評估工具所得的結果有落差。一般所公布的績效評估工具不太符合使用者的需求，實作上變得困難。

三、研究目的

資料庫績效評估普遍存在缺點，包括工作量模式是合成工作量，根據特定的應用領域來建構。使用者拿它們來做績效評估時，會與使用者的使用環境及工作量不符合， 績效評估無法在使用者應用的領域內，提供有效的預測與衡量。

本研究在主從式資料庫管理系統的架構下，針對物件導向資料庫的特性，利用物件導向系統分析方法，讓使用者將使用需求建構成圖形型態的物件模型，轉換成正式的文件型態物件模型，以利電腦內部的儲存與運作。再根據使用者所建構的物件模型，發展一個針對使用者的需求與代表使用者工作量的模型 (Workload Model)。

工作量模型常是績效評估的對等詞，因爲不同工作量形成不同績效結果。如果工作量 1 和工作量 2 不同，形成兩種績效評估，又二個資料庫產生四種不同資料庫工作量，經過測試會產生四種不同績效結果，如圖 1 所示。如果我們一般化工作量模式， Bi 是 i 個績效評估方法，由 i 個不同工作量形成，要比較 j 個資料庫，首先產生 i * j 個測試資料庫，然後取得 i * j 組績效結果。本研究重點在工作量模式建立，(1)期望突破領域預定和專屬的不正確和不適用，(2)突破只針對預設應用系統進行評估造成使用者需求無法被眞正評估，(3)突破績效評估只在於比較不同資料庫的數據，忽略評估的源頭在工作量模型的設計。

要完成這三項突破是非常困難不易驗證。首先，問題根源在於工作量的需求多元、複雜、經常改變、常有錯誤、又不確定、且無法了解。因此，本研究期望自系統化和模式化角度來突破研究困境。我們決定自工作量的使用者需求收集開始，企圖使用一種標準載具 (Common Carrier) 來收集和表達物件工作量的使用者需求。OMG (Object Management Group) 的 UML (Unified Modeling Language) 是目前物件系統分析和設計的標準方法論，提供完整物件模式 (Object Model) 和表達方法 (Representation) 。使用 UML 來系統化和模式化工作需求，加上客製化績效評估功能在 UML 中，建立物件圖形、文字和語法。本研究限制在於不易將學界和業界常用的績效評估方法，一一實作並且重新建置多種物件資料庫，彼此相互比較和驗證。此種驗證方法在執行面上有困難，在成本效益上，重製規模龐大不易進行。本研究因此期望透過標準載具的使用、採用 UML 方法並且加以客製化專門爲績效評估使用、再加上如果能夠以 OMG 的物件模式爲我們的塑模基礎，就可以有開放性和一般化工作量的能力，

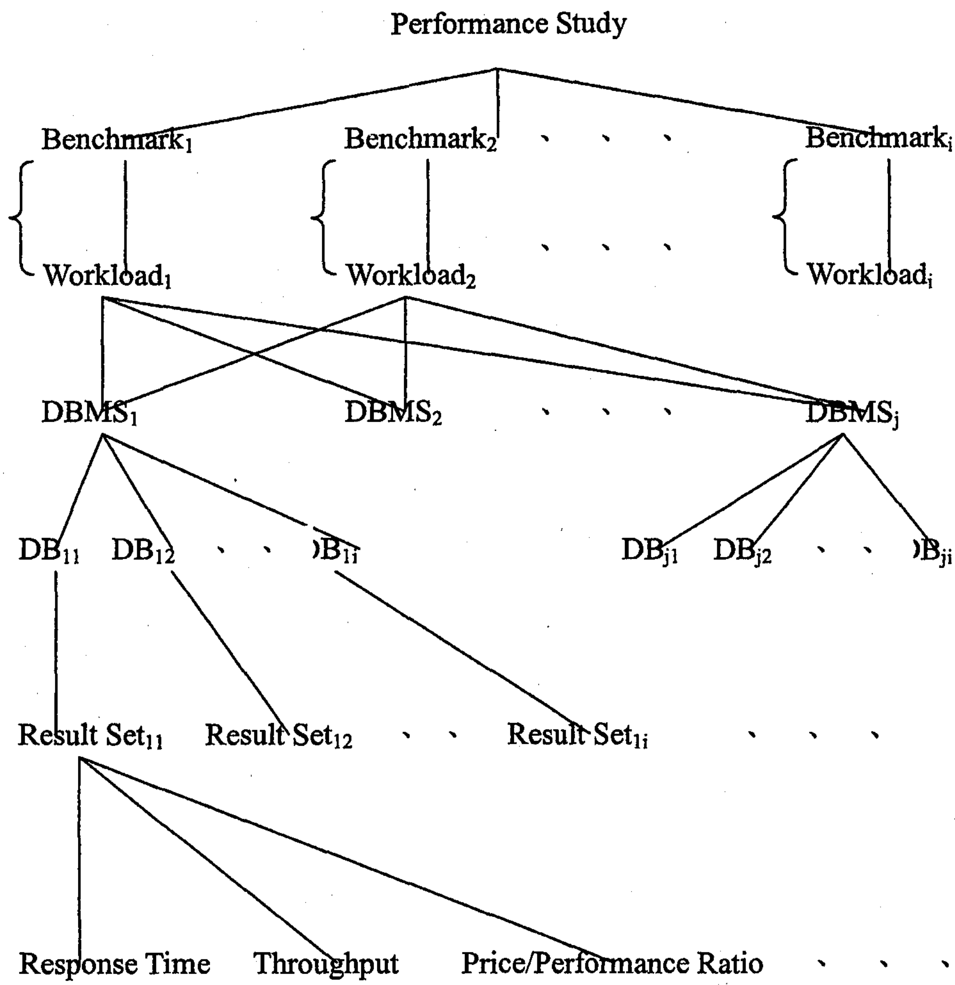

圖1：工作量模型與績效評估之間關係圖

證明本研究方法的可行性，達成三項期望，確實提供系統化和模式化的新解決方案。

貳、文獻探討

一、前言

本研究在於研發問題領域獨立（Domain Indepence）和問題領域代表（Domain Representativeness）的新工作量模型，符合使用者物件資料庫需求的新工作量模型。因此文獻探討中將試以這些議題探討目前相關文獻的研究成果與本研究的關係。本研究的工作量模型將主從式的物件技術中的資料工作量屬性加以系統化和模式化。首先研究物件模式的建立，物件模式指的是物件資料庫的物件、類別、關係、組合、繼承、再用、參考、封裝函式和複雜和複合資料型態的屬性。然後，物件交易模式的建立，呼叫物件函式、引用物件關係、長交易、交易版本控制、單一和複合交易型態，並行存取控制

和回復控制。接著，績效評估控制模式要如何設計方能符合正確、客觀、一致、合理的原則？我們根據標準績效評估方法【 Gray 1993 】【 Carey 1997 】【 TPC 1999 】中基本 6 項控制變數開始設計。因此，本節文獻探討將直接探討物件資料庫中標準績效評估方法 001 （ Object Operation 1 ）、 007 （ Object Operation 7 ）、 BUCKY ，探討以 UML 爲基礎的績效評估方法的自變數、因變數和控制變數。

二、物件導向資料庫的績效評估

在物件導向資料庫的績效評估，目前較廣爲應用的有： 001 、 007 、 BUCKY 等。 001 是衡量資料庫在工程應用上 (CAD/CAM) 效能表現， 001 是第一個針對物件導向資料庫的評估規格，其內容也力求簡單與容易實行。 007 績效評估的方向與 001 類似，衡量物件導向資料庫在工程應用上的效能表現，所不同的是 007 是對系統作多元測試的績效評估標準。因此，相對於 001 而言， 007 具有較複雜的資料庫綱目與較多的運算。 BUCKY 則是用於衡量物件關連式資料庫的效能，針對物件屬性作各種查詢測試運算： 繼承、物件之間的參考、集合值屬性、物件的方法、與抽象資料型態。以下分別簡述這三項績效評估方法。

(一) 001

001 績效評估方法發表於【 Cattell et al., 1992 】，在當時是首創方法也廣爲應用，因爲當時電腦輔助製造與電腦輔助軟體工程會使用到的資料，大部份都是複雜性物件，但傳統資料庫系統績效評估對於這方面應用的規格設計並不十分適當。 001 資料庫主要是由兩種物件所構成：

(1)元件物件： 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、坐標 x 變數、坐標 y 變數、與建構日期 每個元件物件有三個連出去 (out going) 的連結，以及不定數目的連進來 (in going) 的連結。

(2)連結物件： 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。爲了提供參考方面所具備的區域性（ locality ）， 001 的元件物件乃是以其元件代碼爲邏輯上的排序依據，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，所謂的相鄰指的是以代碼值域內的百分之一爲依據。至於 001 的績效評估運算有三個：

a. 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。

b. 物件圖形的走訪運算 (traversal operation) ：利用先深後廣的演算法來做七層的走訪，以走訪總計 3280 個相互連結的元件。

c. 插入運算（ insert operation ）： 產生新的 100 個元件並將它們加入到資料庫中，以便觀察磁碟機 I/O 的效率。

(二) 007

007 績效評估方法發表於【 Carey et al., 1993 】，是爲了能廣泛地測試物件導向資料庫的效能而發展的。由於 001 在這方面的測試簡單， 007 定義了複雜物件型態，並設計複雜物件的運算，基於物件導向資料庫系統的特性 007 提出了多種不同運算，在各種不同運算下的效能表現，將各種測試結果整理後，也可看出系統的整體效益。

007 績效評估之資料庫的主要元素是合成物件，所有合成物件所成的集合稱爲 007 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的一個文件物件，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，單元物件所擁有的屬性如下： Id ， Builddate ， Type ， X ， Y ， DocId ， and Type ，分別表示其代碼，建構日期，型態，變數 X 及 Y ，及文件代碼，連結物件則有如下的屬性： Length ， Type ，分別代表其長度及型態。

007 所測試的運算分爲三類：

(1)走訪 (traversal)：即在物件之間瀏覽，並在經過物件時，呼叫適當的方法 (method) 來執行，走訪運算如下 ：

T1. 純粹走訪的速度 (raw traversal speed)：走訪整個組合階層，當每個基本組合物件被拜訪時，繼續拜訪它的合成物件，即利用先深後廣的演算法來拜訪到最底層的單元物件。

T2. 更新式的拜訪 (traversal with update)：拜訪整個組合階層，並有以下的三種更新方式：

T2a. 更新每個合成物件的一個單元物件

T2b. 更新所有的單元物件

T2c. 更新所有單元物件四次

T3. 更新索引資料的拜訪 (traversal with indexed update) ：與 T2 相同，但更新的資料欄位上有索引，測試在索引欄上更新的負擔。

T6. 稀疏拜訪的速度 (sparse traversal speed) ：拜訪整個組合階層，當每個基本合成物件被拜訪時，繼續拜訪它的合成物件，當每一個合成物件被拜訪時， 只拜訪它的根單元物件。

(2)查詢 (query) ：指一般查詢語言的查詢結果，查詢運算包括：

Q1. 完全符合之搜尋 (exact match lookup) ：任意產生十個單元物件的 ID ，每個 ID 產生後，尋找出此 ID 的單元物件。

Q2. 選出單元物件最近 1% 的 Builddate 之範圍，並找出 Builddate 在此範圍內的單元物件。

Q3. 選出單元物件最近 10% 的 Builddate 之範圍，並找出 Builddate 在此範圍內的單元物件。

Q4. 路徑搜尋 (path lookup) ：任意產生十個文件物件 titles ，當每個 title 建立時，找出與此文件物件所指的合成物件有關係的基本組合物件，並算出這些相關基本組合物件的數目。

(三) BUCKY

BUCKY 績效評估方法發表於【 Carey et al., 1997 】。它主要是查詢導向的績效評估，用來測試物件關連資料庫的功能，並期能與關連式資料庫做比較，以便瞭解物件關連資料庫對於關連式資料庫的技術成熟度爲何。

BUCKY 的測試資料庫以模擬校園師生資料庫，因而其測試運算有：

(1)繼承 (inheritance) ：資料庫的根型態爲 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。

(2)物件間的參考 (inter-object reference)：物件關連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的關係，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連資料庫的精神。

(3)集合屬性 (set-valued attribute)：它允許集合屬性值的儲存，例如：Person_t 擁有名稱為 KidNames 的屬性，其型態為 Set(Varchar(10))，表示其屬性為字串的集合每一個字串代表一個小孩的姓名。

(4)抽象資料型態 (abstract data type)：抽象資料型態可以讓使用者自行定義自己的資料型態，使用者所定義的資料型態也可以用在 SQL 語法上，尤如系統內建的型態一般。

以上 BUCKY 物件關連測試資料庫和運算，很明顯將純物件和純關連的基本差異進行測試，並且強調在同一個資料庫系統支援傳統關連系統和混合物件關連系統時候，應用 BUCK Y 比較適合。

三、物件導向系統分析之方法

系統分析是將眞實世界中的人事物關係找出來，並加以設計模型，物件導向系統分析與設計方法提供再使用與擴充性的架構。1994 年 Rational 軟體公司的 Grady Booch 與 Rumbaugh ，開始致力於合併 Booch OMT (Object Modeling Technology) 的方法，取此兩者長處的 UML (Unified Modeling Language) 方法於是誕生，至 1995 年秋季 Jacobson 亦加入統一物件導向系統分析與設計的方法論。1997 年由 OMG (Object Management Group) 認定為物件技術標準系統分析與設計之圖形語言方法論。

UML 是一種模型語言 (Modeling Language)，主要是用來說明與建構物件資訊系統，UML 的作者特別將焦點用於解決同步性 (Concurrency) 與分散性 (Distribution) 的問題，同步性代表交易異動並行控制和版本控制，在多使用者環境中；分散性代表交易異動在本地端和遠端交易進行；兩者都需要協力 (Collaboration)、順序 (Sequence) 和狀態 (State) 來表達；亦即作者在 UML 中加入了可以表達這些問題的元件。

本研究使用 UML 作為工作量模型的標準載具，因為其物件模式廣泛完整、物件理論嚴謹、與物件程式語言可以連結、圖形與符號的語言表達豐富。本工作量模型在資料模式和交易模式將應用 UML 圖配合績效評估方法論的自變數和因變數的設計，以 UML 各圖形和符號需要文字和語法表達，提供可程式化基礎，系統化的工作量需求物件。UML 基本原理為：

(一)觀點 (Perspectives)

UML 使用圖形來表達模型的語意，但在不同的系統發展階段，圖形所傳遞的語意也會有所不同。以下三種觀點與工作量模型的研發、規格產生和實驗執行是一致的。

(1)概念觀點 (Conceptual)：在此觀點下，圖形所代表的就是問題領域內的概念，其與所實作的軟體無關，亦即是語言獨立的階段。

(2)規格觀點 (Specification)：在此觀點下，我們的焦點是放在軟體的介面而不是它的實作。

(3)實作觀點 (Implementation)：在此觀點下，我們就眞的以類別內部的實作為重點，這個觀點是最常被使用，但有時規格觀點反而能更清楚地表達系統的模型語意。

㈡圖形 (Diagrams)

根據觀點， UML 定義基本物件模式，包括物件、類別、函式、資料型態、連接關係、繼承關係、組合關係、參考關係表達、靜態和動態物件的建立、修改、應用，使得物件得以封裝化成爲抽象資料型態，具有資料獨立性和物件再使用性。圖形是用來描述模型語意的圖案， UML 共提供了九種不同類型的圖形，來供使用者用來描述系統的不同觀點。

(1)類別圖與物件圖（ Class Diagram and Object Diagram ）： 所謂類別就是一群具有類似結構，行爲和關係之物件的描述器，類別圖分成三個部份，第一個部份是類別的名稱，第二個部份爲類別的屬性，第三個部份則是類別的方法。

(2)使用案例圖（ Use Case Diagram ）： 所謂使用案例圖就是用來顯示一系統內行爲者與使用案例之間的關係，而行爲者與使用案例之間的溝通是由一條實線來完成。

(3)順序圖（ Sequence Diagram ）：順序圖是用來描述一群物件之間的互動，然而它們之間訊息的傳遞是以時間順序來安排，順序圖並不顯示物件間的關聯。

(4)合作圖（ Collaboration Diagram ）： 合作圖與順序圖不同之處在於除了表達物件之間的訊息互動外，合作圖更進一步表達出物件彼此間的關連，但不像順序圖一樣能表達出時間的先後順序，因此訊息的時間順序只好以數目字額外標示。

(5)狀態遞移圖（ State Diagram ）：狀態遞移圖是表示物件在它的生命週期內接受到刺激後，所引起一連串狀態的改變，物件狀態遞移圖又可分成狀態、事件與遞移來作說明。

(6)活動圖（ Activity Diagram ）：活動圖是狀態遞移圖的特例，所不同的是在活動圖中所有的狀態都是活動狀態，且每一次活動狀態的遞移都是由上一個狀態的動作完成來驅動，活動圖的目的是著重在由內部程序所驅動的活動流程。

(7)元件圖（ Component Diagram ）：元件圖是用來表示軟體元件間的關係，軟體元件可能存在於編譯時期，連結時期或者執行時期，元件圖只用來表示元件型態，若想表示元件實例就必須使用採用圖。

(8)採用圖（ Deployment Diagram ）： 採用圖主要是用來顯示系統執行時，處理單元與在其上執行的軟體元件，程序與物件的架構，採用圖是由許多節點經由通訊關聯的連接所形成的圖形。

參、研究方法與研究模式

一、前言

本研究的工作量模型開發分成三個階段，如圖 2 所示。

1. 工作量需求分析階段：本階段的主要目的是建構一個簡單，易於明瞭的圖形型態工作量模型，主要的工作在於選擇一個適當的系統分析方法，能夠建構出代表使用者需求的工作量模型。

2. 工作量規格描述階段：主要是將系統分析階段所建構產生的圖形型態工作量模型，對應至電腦可以處理的文字型態工作量模型，主要的工作是將圖形型態工作量模型對應至事先定義好的語法，以產生文字型態規格描述，這些工作量規格描述將是建構績效評估程式之依據。

3. 績效評估測試階段：這個階段的主要工

作是根據前一階段所產生的文字規格描述，依據系統與資料庫平台的種類，產生相對應的程式碼建構出相應的工作量程式碼，進行實驗執行。

圖 2 中第一階段工作量需求分析階段資料模式使用物件圖和類別圖，將物件模式以標準載具方式實作。交易模式以狀態圖、合作圖、分配圖，將靜態和動態交易類別，單一和複合交易類別、長交易、版本控制、異動管理以及主從式架構以標準載具方式實作。控制模式如前所述，基本六項控制變數將以 Scripts 方式獨立承載。第二階段工作量規格描述階段，本研究以 BNF 定義之語言和語法，經過編譯產生物件建立語言（Object Definition Language）和物件查詢語言（Object Query Language），符合 ODMG（Object Data Management Group）對於物件資料庫的語言規範，使得後端物件資料庫可以執行績效評估實驗。第三階段績效評估階段，根據工作量規格書由系統自動產生程式碼去建立測試用物件、類別、資料型態、關係等，並且建立實驗順序、次數、狀態，驅動物件資料庫執行各項實驗。機率分佈和隨機產生機制非常重要，在合成工作量中，資料模式有賴機率分布來產生測試資料庫，交易模式有賴機率分布來產生交易輸入值，控制模式有賴機率分佈來決定執行順序、權重、次數、狀態。

本研究以建立趨近泛用型（Toward Generalized）的載具（Common Carrier），來承載不同的工作量需求。此載具不是實際物體，而是需求分析方法和技術，能夠承載不同使用者可能的工作量需求。自研究文獻中，我們得知不同使用者需求工作量，不太可能使用同一個方法和技術達成，因此有了領域設定 (Domain Specific) 的工作量模型方法 (Workload Modeling)，針對不同應用環境來設計不同工作量。本研究認爲領域設定有其優點，亦爲 001 、 007 、 BUCKY 等標準績效評估方法所採

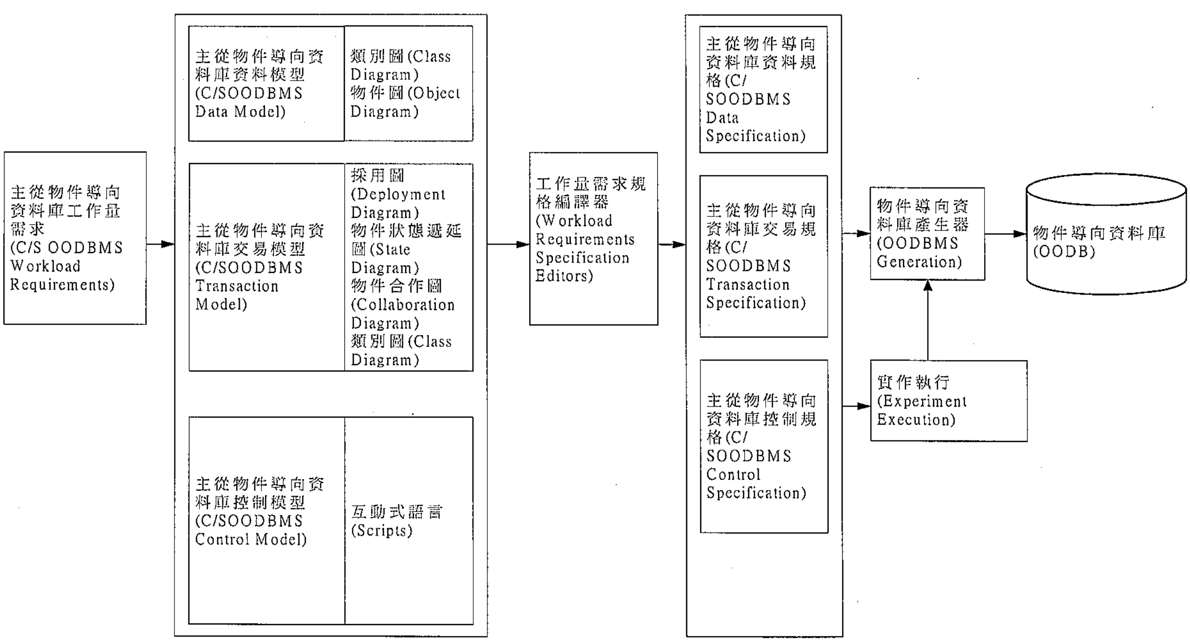

圖2：本研究的研究模式架構圖

行，但是其主要缺點在於每一個領域設定的績效評估結果不能用來表達自己本身的績效評估結果。因爲使用者的工作量，應該和 001 、 007 、 BUCKY 不同，所跑出來的 001 、 007 、 BUCKY 數據只能做爲參考用估計，因爲一定和 001 、 007 、 BUCKY 不同。所以，目前的領域設定的做法有其瓶頸，亦爲【 Carey et al., 1993 】【 Cattell et al., 1992 】【 Gray 1993 】【 DeWitt 1993 】所亟思解決。

有鑑於此，本研究之趨近泛用型載具概念是一項突破，讓使用者提供其工作量需求，透過載具成爲高階需求規格書，具有語法和語意，再經由編譯程式，轉換高階規格書成爲程式碼，產生測試資料庫，發出物件查詢、運算、和移動要求。使用者導向的工作量測試套組，評估結果就會趨近未來使用者使用物件資料庫的狀況。因此，載具的產生很重要，它的一般化和泛用性，可以決定工作量的正確性和眞實性。

本研究使用 OMG 物件模式（ Object Model ）的基本架構作爲載具，因爲它是基礎（ Generic ），包含大多數物件資料庫組織和元素。同時，使用 OMG 物件分析方法 UML 表達物件模式，如圖 3 所示。由於物件資料庫和績效評估所需要用的元素和原本 OMG 的物件模式是超集合和子集合的關係，並非全部元素都使用。由於本研究基於資料模型、交易模型、控制模型的應用需求，因此，採用 UML 的 Class Diagram 、 Collaboration Diagram 和 State Diagram 來表達測試需求。

特殊化 UML 有兩個步驟，第一個步驟是選擇圖形工具，可以根據 OMG 的物件模式的元素來決定，例如：如何表達型態，如何表達類別，如何表達物件和關係。第二個步驟是根據物件資料庫的模式和元素，描述出泛用型的圖形規格，例如：如何表達物件遷移，如何表達單一交易，如何表達複合交易，如何表達版本控制。第三個步驟是設計語法和語意，將圖形表達的內容，以高階需求語言方式，依照每一項元素和關係，以 BNF 方式設計。本研究的工作量需求分析載具，透過原生性 OMG/ODMG Object Model ，達成趨近一般化和泛用型的設計。它的缺點在於本質仍是合成型工作量（ Synthetic Workload ），趨近的意思受限於有多少機率分佈可以設計在載具中。

本研究的實作雛型系統，可以表達 001 、 007 、 BUCKY 的工作量。如同文獻探討， 001 工作量較爲簡單，僅用 refer to 關係。 007 工作量較爲複雜，包括參考關係、繼承關係、組合關係、路徑搜尋和函式呼叫。 BUCKY 工作量複雜度介於其中，有基本參考、繼承、組合關係，以及查詢物件。所以，在 UML 中表達 001 、 007 、 BUCKY 並無困難，轉換爲本研究之需求規格，它們的工作量爲本研究工作量模式所涵蓋，成爲子集合。

本研究的實作系統，著重在證明載具和需求分析概念的可行性，不在證明不同物件資料庫的績效比較，也不在執行實驗以產出數據。所以，我們使用一項通用例子，顯示如何輸入工作量到載具中，如何建立資料庫，如何依照控制變數執行交易和運算。事實上，解析器（ Parser ）扮演重要角色，因爲，透過它方能將需求規格語言轉換成 UniSQL 可以接受執行的程式碼建立資料庫，執行物件交易和收集績效數據等。使用者不同的工作量需求，可以自動產生測試套組，反應正確和眞實的需要，得到的績效數據比較能夠反應系統的優劣。

二、工作量需求分析階段

此階段的主要目的是建構一個能代表使用者應用需求的工作量模型，依照分析的各個步驟進行將可建構出一個物件模

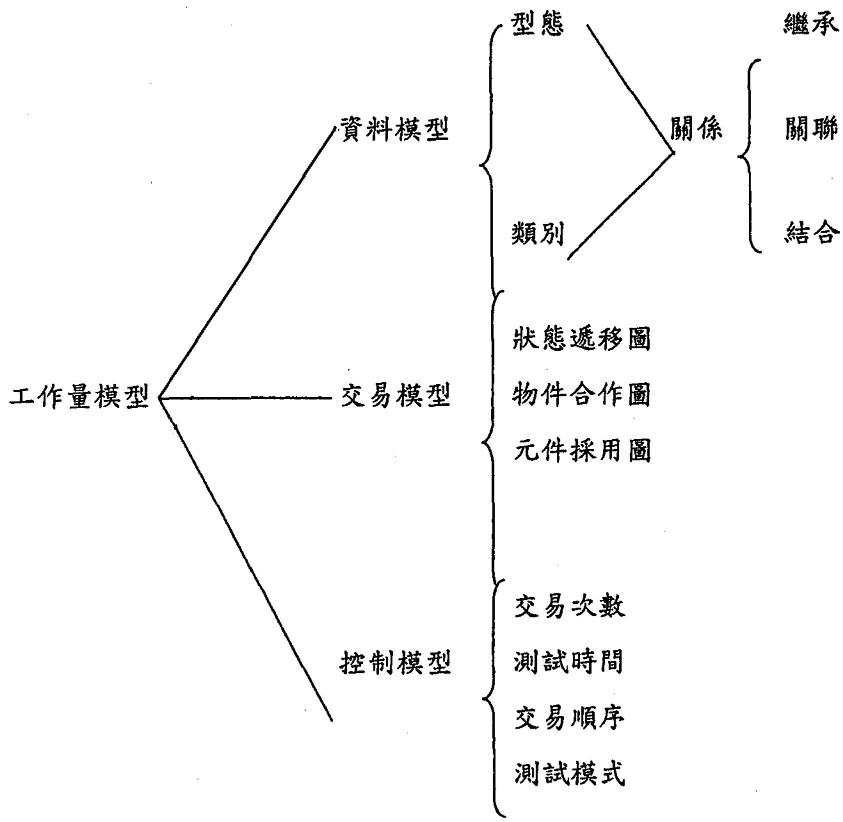

圖3：工作量模式的UML應用分解圖

型。我們將以ＵＭＬ為基礎，作為建構工作量模型的主要工具，並加以適當的改良，以期能符合物件導向資料庫的績效評估特性。

㈠資料模型（ Data Model ）

資料模型主要是以系統靜態的觀點來描述工作量模型裡之資料的結構與關係，建構資料模型亦須對使用者的需求作分析，以試圖找出問題領域內的實體，實體的結構與實體間的關係。

1.型態

型態是用以描述值，因此值就是型態的案例，值沒有識別值來識別其間的差異，值存在物件的屬性或運算的係數裡。

2.類別

類別圖分成三個部份，第一個部份是類別的名稱、第二個部份為類別的屬性，屬性依照種類與複雜度又可分成數種。第三個部分是函式的定義和名稱。

(1)簡單屬性 (simple attribute) ：是指屬性型態為系統內建的基本型態，例如： integer ， real ， string and Boolean 等。
(2)參考屬性 (reference attribute) ：其資料型態為物件識別的屬性，參考屬性被用來代表物件間存在的關係，類似程式語言中的指標。
(3)集合屬性 (collection attribute) ：其資料型態為結構化的型態。
(4)衍生屬性 (derived attribute) ：其屬性可以不以值的方式儲存，而是

以定義程序的方式來儲存。

3.型態與類別間的關係

其代表型態或類別間存在著某些邏輯上的連接關係，型態與類別間的關係可分爲以下三種：

(1)繼承 (inheritance)：其能夠表達型態與類別間特殊性與一般性的概念。

(2)關聯 (association)：指出型態與類別間彼此邏輯上的關係，主要爲朋友關聯(friend)、使用關聯(using)、與實例化關聯 (instantiate)。

(3)組合 (aggregation)：其表達著整體部分的概念，高階的類別可以封裝低階的構成類別，因此，高階的類別提供界面以供外界溝通之用。

(二)交易模型 (Transaction Model)

交易模型主要是從使用者應用系統動態行爲與時間上的觀點來描述，目的在表達系統交易的流程與處理邏輯。在分散式的概念下，我們擬將交易分成三種概念。第一種爲物件遷移 (Object Migration)，是指在分散式系統下， 物件會隨著時間的不同，而在不同地點之間遷移，對於此狀況我們使用採用圖來表示。第二種爲單一交易 (Singular Transaction)，指的是系統執行時與案例相關的行爲，我們使用物件狀態遞移圖來表示。第三種爲複合交易 (Compound Transaction)，指的是許多物件間互相溝通訊息與彼此間的互動行爲來達成某種交易行爲，我們使用物件合作圖來表示。第四種爲版本控制 (Version Control)。

1.物件的遷移

對於物件遷移的狀況，我們是使用採用圖來表達，採用圖主要是用來顯示系統執行時，處理單元與在其上執行的軟體元件，程序與物件的架構，採用圖是由許多節點經由通訊關聯的連接所形成的圖形，節點可能包含元件實例，元件可能包含物件，這關係表示物件是元件的一部份，元件之間的連接是經由依賴關係，這表示元件使用了另一元件的服務。物件的遷移表達績效評估中物件工作量的新增、刪除、修改和查詢。

2.單一交易

單一交易是使用物件狀態遞移圖來表示，物件狀態遞移圖是表示物件在它的生命週期內接受到刺激後，所引起一連串狀態的改變，物件狀態遞移圖又可分成狀態、事件與遞移來作說明。單一交易表達績效評估中物件函式呼叫工作量和函式建立、新增、刪除、修改，包括長交易工作量的生命週期。

3.複合交易

複合交易是使用物件合作圖來表達交易的概念，物件合作圖主要是用來描述物件間的交互作用與訊息傳遞關係，而物件間的交互作用與訊息傳遞的現象就是所謂的行爲，因此物件合作圖可以用以表達交易所牽涉到的資料與物件函式。物件合作圖分成兩個部份，第一個部份是有那些物件在物件合作圖內扮演角色，這屬於靜態部份，第二部份是物件間爲達某些目的，彼此間相互溝通的訊息順序等，這屬於物件合作圖的動態部份，訊息的表示主要是在物件的連接關係旁加一箭頭，代表由來源物件傳遞訊息至目標物件。複合交易表達績效評估中物件交易與物件交易互動，物件關係類型的啓動，異動管理中同時存取控制和回復控制。

4.版本控制

物件類別的版本控制主要是應用在當有二人以上的使用者對同一個物件類別作更新時，就會有許多的物件類別版本同時存在於系統之中。因此，資料庫管理系統就必須提供使用者不同版本間管理與合併的功能，以符合不同使用者對於物件版本內容的需求。至於物件類別版本控制的表

達方式，我們是使用UML的類別圖中之屬性方式和物件函式的語意來加以表達。

三、工作量規格描述階段

在發展測試績效評估系統的過程，第二階段的主要工作是將工作量分析階段所繪製的圖形型態（Diagrammatic Representation）對應至電腦可以處理與儲存的文字型態（Textual Representation），我們使用類似綱目定義語言的方式來對映工作量模型的靜態觀點---資料模型，再利用程序與運算的表達方式來對映工作量模型的動態觀點---交易模型與控制模型。爲了能更嚴謹地定義工作量模型語言，且可讓下一步的語法分析器或編譯器（Parser）作分析，我們使用了BNF (Backus Nauru Form) 或生產規則（Production Rule）的方式。每一個符號皆擁有一個生產規則，符號可能是句子，變數名稱或者分號以下是績效評估的規格描述語言之語法定義，再接下來的部份則依序說明如何使用規格描述語言來描述工作量模型的資料模型，交易模型與控制模型。

績效評估工作量模型定義：

```
<benchmark specification>
::= DEFINE BENCHMARK FOR <bench-
mark name>
<workload specification>...
END BENCHMARK
<benchmark name>
::= <identifier>
<workload specification>
::= DEFINE WORKLOAD FOR <work-
load number> <workload name>
<data specification>
<transaction specification>
<control specification>
END WORKLOAD
<workload number>
::= <integer>
<workload name>
::= <identifier>
<data specification>
::= DEFINE DATA SPECIFICATION
<class definition>. ..
END DATA SPECIFICATION
<transaction specification>
::= DEFINE TRANSACTION SPECIFI-
CATION
<migration specification>...|
<singular transaction specification> ... |
<compound transaction specification>...|
<version control specification>...
END TRANSACTION SPECIFICATION
```

基本資料型態定義：

```
<character>
::=<digit> | <letter> | <special character>
<digit>
::= 0|1|2|3|4|5|6|7|8|9
<letter>
:: <lower case letter>|<upper case letter>
<lower case letter>
:: a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<upper case letter>
::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|
U|V|W|X|Y|Z
<special character>
::= ! | @ | # | $ | ^ | & | *
<data type>
::= <basic data type> | <user-defined data
type>| <structured data type expression>
<basic data type>
::=INTEGER|REAL|BOOLEAN | STRING
<user-defined data type>
::=<uppercaseletter>[{<,uppercaseletter>}]
<structured data type expression>
```

::=<structured data type> (<basic data type>)
<structured data type>
::= SET | BAG | LIST | TUPLE | ARRAY | MULTILIST
<compare operator>
::= <>|=|<|>|<=|>=
<string>
::=<character>...
<identifier>
::= <upper case letter> [{ [_] { <upper case letter> | <digit> } }]
<numeric literal>
::= <exact numeric literal> | <approximate numeric.literal> |
[+ | -] <unsigned integer> [. <unsigned integer>] |
[+ | -] <unsigned integer>. |
[+ | -] .unsigned integer>
<unsigned integer>
::= <digit>...
<approximate numeric literal
::= <mantissa> E <exponent>
<mantissa>
::== <exact numeric literal>
<exponent>
::= [+ | -] <digit> ...

(一)靜態觀點的資料模型：

如前所述資料模型主要是描述資料庫內包含哪些類別，類別間的關係與類別的結構等。

<class definition>
::= DEFINE OBJECT CLASS FOR<class name>
NUMBER_OF_ROWS <number of rows>
[SUPERCLASSES_ARE <superclass name>...]
ATTRIBUTES <attribute definition>...
[OPERATIONS <operation definition>...]
[KEY <key definitions>]
END OBJECT CLASS
<class name>
::= <identifier>
<superclass name>
::= <identifier>
<attribute definition>
::= <attribute name> ： <data type>
<attribute name>
::= <identifier>
<operation definition>
::=<operation name>(operation augment) ：<return value data type>
<operation name>
::= <identifier>
<operation augment>
::= <data type> [{ ,<data type>}]
<return value data type>
::= <data type>
<constraint definition>
::=<attribute name> <compare operator > <compare target>
<compare target>
::= <attribute name> | <numeric literal>
<key definition>
::= (<attribute name>[{ ,<attribute name > }])

(二)動態觀點的交易模型與控制模型

工作量模型的動態觀點主要包含交易模型與控制模型，交易模型我們又將之分成物件的遷移，單一交易模型，複合交易模型與版本控制，在圖形型態的工作量模型裡，分別使用狀態遞移圖，物件合作圖，採用圖與類別圖來表達這些概念。

1. 物件的遷移

物件的遷移在物件導向資料庫的主要目的是用來衡量，物件類別在不同的儲存

點之間，其移動複製的績效。以下則是使用文字型態的方式，來表達物件遷移的例子：

<migration definition>
::=DEFINE MIGRATION <transaction number>
<class name>
<migration entry>
END MIGRATION
<transaction number>
::=<integer>
<class name>
::=<identifier>
<migration entry>
::=TIMER BEGIN <clock>
MIGRATE FROM SERVER <source address number>
MIGRATE <migrate string>
MIGRATE TO SERVER <destination address number>
TIMER END <clock>
<clock>
::=<timer name> ： <return time>
<return time>
::=hh|mm|ss
<source address number>
::=<integer>.<integer>.<integer>.<integer>
<destination address number>
::=<integer>.<integer>.<integer>.<integer>
<migrate string>
::=<migrate name>(<parameter>...)

2. 單一交易模型

單一交易模型使用狀態遞移圖來描述其交易流程，狀態遞移圖可以藉由物件的運算對應至文字型態的語言，以運算爲基礎的設計，主要是指定運算在狀態遞移階段的運算責任。因爲運算有一個條件可以決定行動是否要予執行，當條件成立時，運算才予以執行，由以上的描述可知，運算可分成兩個部份，運算的條件部份表達狀態遞移圖的條件，此條件檢查運算是否必須執行。通常這必須檢視物件的狀態，運算的行動部份是在條件成立時，藉由適當的更新屬性值以改變物件的狀態，以下則是使用文字定義單一交易的語法：

<singular transaction definition>
::=DEFINE SINGULAR TRANSACTION FOR < transaction number>
<class name>
<singular transaction entry>...
END SINGULAR TRANSACTION
<singular transaction entry>
::=STATE FROM <state name>
<state transition entry>
STATE TO <state name>
<transaction number>
::=<integer>
<state name>
::== <identifier>
<state transition entry>
::= <event signature> [<guard condition>]/ <action expression> ^ <send-clause>
<class name>
::= <identifier>
<event signature>
::= <event name> (<parameter> ...)
<guard condition>
::= <attribute name> <compare operator> <numeric literal>|<string>
<action expression>
::=<operation name>(<operation argument>)
<operation argument>
::= <data type> [{ ,<data type>}]
<send clause>
::=<object name>.<event signature>
<real value augment>
::= <string> | <numeric literal>[{ ,<string>| ,<numeric literal>}]

3. 複合交易模型

複合交易模型使用物件合作圖來表達交易的邏輯與各物件間的函式呼叫，所以前述的物件合作圖包含兩部份，包括有那些物件及彼此間的交互作用的關係，我們使用以下的語法來對應物件合作圖：

<compound transaction definition>
::= DEFINE COMPOUND TRANSACTION <transaction number>
<compound transaction name>
<message entry>...
END COMPOUND TRANSACTION
<transaction number>
::=<integer>
<transaction name>
::= <identifier>
<message entry>
::= NUMBER <sequence number>
[CONDITION <condition expression>]
MESSAGE FROM CLASS <class name>
MESSAGE <message string>
MESSAGE TO CLASS <class name>
<condition expression>
::=<condition>
<class name>
::= <identifier>
<object name>
::= <identifier>
<message string>
::=<operation name>(<operation augment>)

4. 版本控制

在多人使用的物件導向資料庫之環境下，根據應用系統領域的需求，想符合不同的使用者能看到各自所需的物件內容，就必須依賴資料庫管理系統所提供的版本控制功能，使得不同物件類別的管理與合併對於使用者而言是具有透通性，以下則是使用文字型態來表達的例子：

<version control specification>
::=DEFINE VERSION CONTROL <transaction number>
<class name>
<control entry>
END VERSION CONTROL
<transaction number>
::=<integer>
<class name>
::=<identifier>
<control entry>
::=TIMER BEGIN <clock>
MERGE MULTIPLE VERSIONS <number_of_merge>
<merge entry>
TIMER END <clock>
<clock>
::=<timer name> ： <return time>
<return time>
::=hh|mm|ss
<merge entry>
::=<merge signature>/<action expression>
<merge signature>
::=<merge name>(<parameter>...) ： <return new version>
<action expression>
::=<operation name>(<argument>)
<argument>
::=<return new version>

5. 控制模型

控制模型並沒有圖形來表示，我們用以下的語法來表達多種控制變數定義。

<control definition>
::=DEFINE CONTROL SPECIFICATION
<transaction entry>...
END CONTROL SPECIFICATION
<transaction entry>
::=<transaction type> <sequence number>
<steady state>
TIMES <repetition number>

MODE <mode type>
DURATION <duration time>
METRICS <scale>
<transaction type>
::= MIGRATION|SINGULAR TRANSACTION |
COMPOUND TRANSACTION|VERSION CONTROL
<sequence number>
::= <integer >
<steady state>
::=<start_time>
<repetition number>
::= <integer>
<mode type>
::=COLD|WARM|HOT
<duration time>
::=hh|mm|ss
<scale>
::=REPONSE TIME|THROUGHPUT

以 BNF 表達，研究模式首先將使用者的工作量需求圖形表達，轉換文字形工作量，符合高階需求分析，透過翻譯器將規格轉換成爲建立資料庫的物件程式碼。經過這三個階段轉換，使用者不同的工作量需求，可以自動產生測試套組，反應正確和眞實的需要，得到的績效數據比較能夠反應系統的優劣。第一階段是高階圖形表達，減少輸入工作；第二階段是高階文字表達，可跨平台不受實作系統限制；第三階段是依不同平台，透過翻譯器處理程式碼，以便執行實驗。經由編譯器，將關鍵字（ Token ）和語型（ Pattern ）分析成爲語法。如圖 4 所示。

四、績效評估測試階段

工作量需求分析結果要在這一個階段驗證其可行性，驗證以標準載具來承載不同使用者應用需求是可行的。圖 5 表示測試階段是由驅動程式驅動，根據規格書，分別產生測試資料庫、測試交易和測試控制變數，由測試收集器收集各種統計值。測試控制變數定義：(1)績效指標（ Performance Metric ），包括回應時間（ Response Time ）和其倒數指標（ Inverse Metric ）產出（ Throughput ），(2)測試期間（ Test Duration ），(3)測試穩定狀況（ Steady State ），(4)測試次數（ Replication ），(5)測試順序（ Test Sequence ），(6)交易優先性（ Transaction Precedence ），和(7)使用人數（ Number of Users ）。透過控制模型轉換成程式碼，控制實作進行。

肆、雛形系統建置

我們除了建構工作量模式的理論架構外，我們進行研究的實作，用以驗證本研究所提出的工作量模型理論架構之可行性。本研究的雛型系統之整體架構亦可作爲未來欲發展物件資料庫績效評估的研究人員或企業內欲對主從式資料庫系統作績效評估的人士所參考。

一、系統開發工具與雛型系統架構

本研究之雛形系統所使用的硬體是 Sun Microsystem 公司的 Ultra 工作站，其上的作業系統是 Solaris ，基於取得上的限制，因此在物件導向資料庫系統的採用上，我們選擇了具備關連資料庫的功能外，能處理物件特性的物件資料庫系統---Unisql/X 和 Unisql/M 。此外，爲了達到具有主從式資料庫的特性，經多方的討論研究，決定將網際網路資料庫的概念引進到本實作系統中。因此，我們選用了 Apache Server 成爲我們的 Web Server 。爲了溝通 Web Server 與 Unisql/X ，我們採用 UniWEB 中介軟體應用程式。使用者就可以在客戶端單純地以瀏覽器爲其界

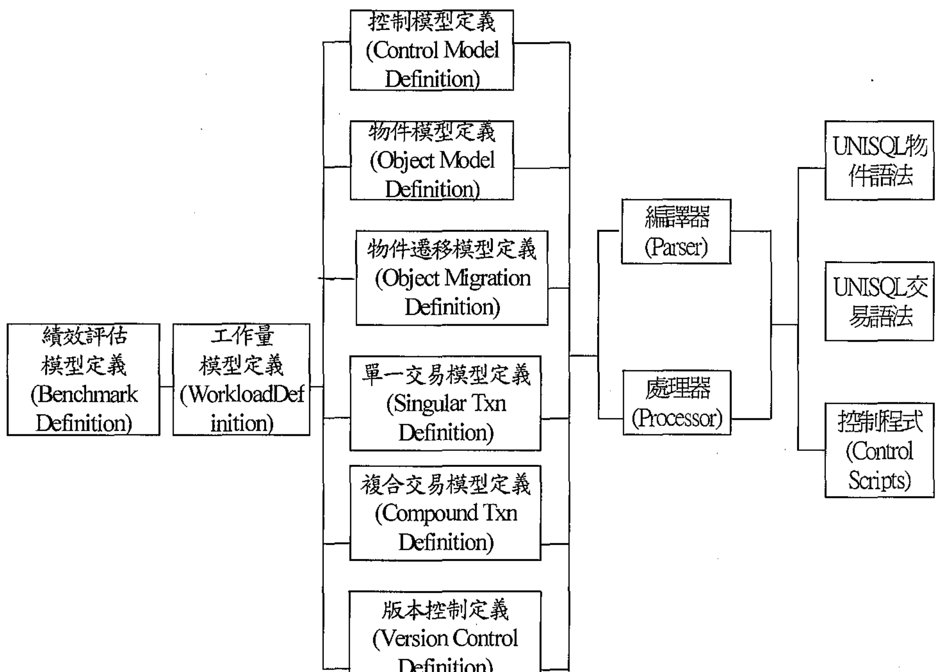

圖4：工作量規格書和工作量程式碼關係圖

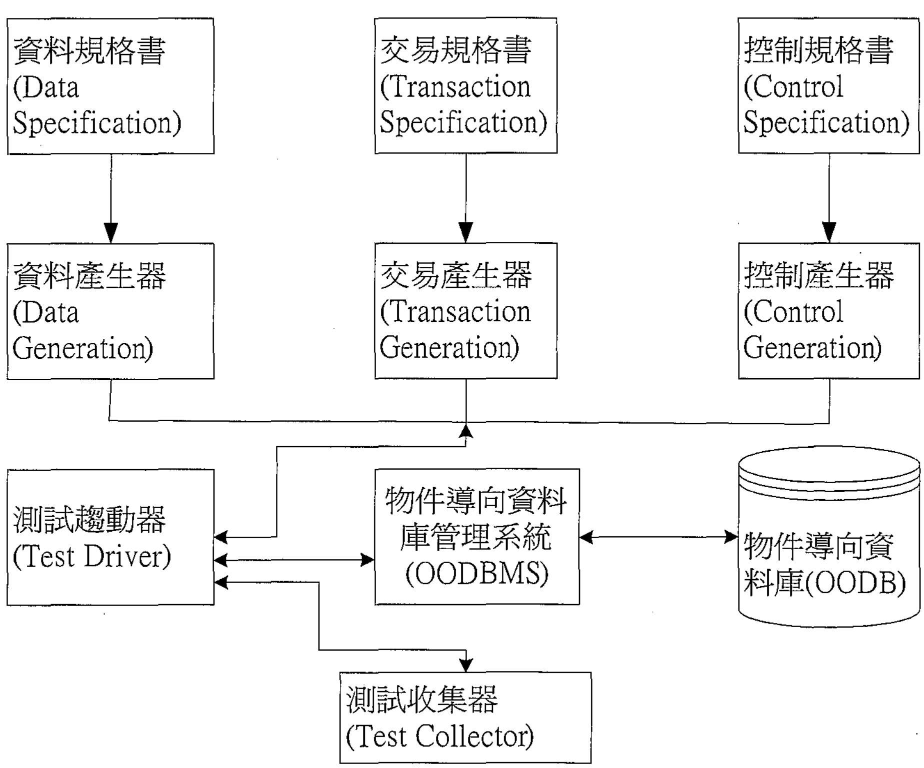

圖5：績效評估測試流程示意圖

面，針對其本身的需求，經由網際網路來輸入其需求的資料庫綱目、交易邏輯和控制變數。因爲物件資料庫多爲工程應用系統，因此，我們選擇主流物件資料庫系統以 Solaris 爲開發平台。目前成熟穩定符合 ODMG 標準的物件資料庫有十幾種，其中能提供 SQL 及關連技術介面僅有 Unisql 及 Jasmine。本研究爲了同時考量網際網路連接軟體和主從架構開發工具的函式庫，選擇 Unisql/X 和 Unisql/M，因爲它的物件和 SQL 關連技術並存，又選擇 UniWEB 提供網際網路中介軟體，Unisql/M 主從架構中介程式。以下是本研究實作的系統架構圖，如圖 6 所示。

我們將工作量模式編輯系統分爲下列數個子系統：分別爲類別編輯器，物件類別遷移編輯器，單一交易編輯器，複合交易編輯器，物件版本控制編輯器與控制模型編輯器，以下是對各個編輯器說明。

(一)類別編輯器

使用者使用瀏覽器進入類別編輯器時，首先見到的畫面是依照工作量模式規格描述語言所定義的表達法所設計如圖 7。使用者進入畫面後，就可以依照自己的資料庫綱目中的類別定義，分別填入類別的名稱，屬性，資料型態與類別內部的運算方法，當一切都按照語法填入完成後，使用者只須按下 submit 按鈕，系統自動會將使用者的類別定義資料，從前端的瀏覽器傳送到後端的 Web Server，再經由 UniWEB Server 傳給 Unisql/X 資料庫系統，資料庫便會根據所收到的類別定義來建立使用者所需的類別。

使用者在填入類別定義資料後，按下 submit 鍵會連結至一網頁，詢問使用者是否要查詢其剛才所建立的類別資料綱目，或者是要立即輸入物件於剛才所建立的類別，其詢問的畫面如圖 8 所示：

若使用者在類別定義資料傳遞到後端資料庫後，想確定是否其類別資料綱目是否建立完成，可以按下 check 連結，並將查詢類別資料綱目的命令傳遞到後端資料庫，資料庫在收到查詢命令後，就會立即將剛剛所建立完成的類別資料綱目傳遞到使用者的網頁上，其畫面如圖 9 所示：

在使用者確定後端資料庫將類別資料綱目建立完備之後，即可按下圖 9 中的 insert 連結，以便進入建立類別實例的畫面，如圖 10 所示，使用者即可根據上半部類別屬性的提示，依序在空格中輸入資料，當實例的資料輸入完畢後，使用者即可按下 submit 鍵，同樣將實例的資料經由網際網路傳遞到後端資料庫，以便完成 commit 的步驟。

在使用者完成上述步驟後，類別物件的建立程序大致完成，最後系統會傳遞回給使用者一頁類別物件的目前資料表格，以及剛剛所建立的實例之物件識別號碼 (OID)，其結果如圖 11 所示：

以上之類別編輯器是由六個不同的程式所構成，其六個不同程式之間的關係包括首頁開始，輸入資料綱目，建立資料庫，新增、刪除、修改資料綱目，查詢物件資料庫的程式。如圖 12 所示：

(二)物件類別遷移編輯器

物件類別遷移編輯器主要是對於資料模式內的類別物件，在分散式的環境下提供一個類別物件複製的語法表示法，以便讓後端的資料庫程式產生器能夠據以產生所要的程式，物件類別遷移編輯器的畫面如圖 13 所示：

當使用者輸入完各個欄位所需的資料後，即可按下 submit 鍵，將建立工作量模型規格描述所需的資料傳遞至後端的系統，以便系統依據使用者所輸入的規格描述，來建立符合使用者需求的規格描述書。

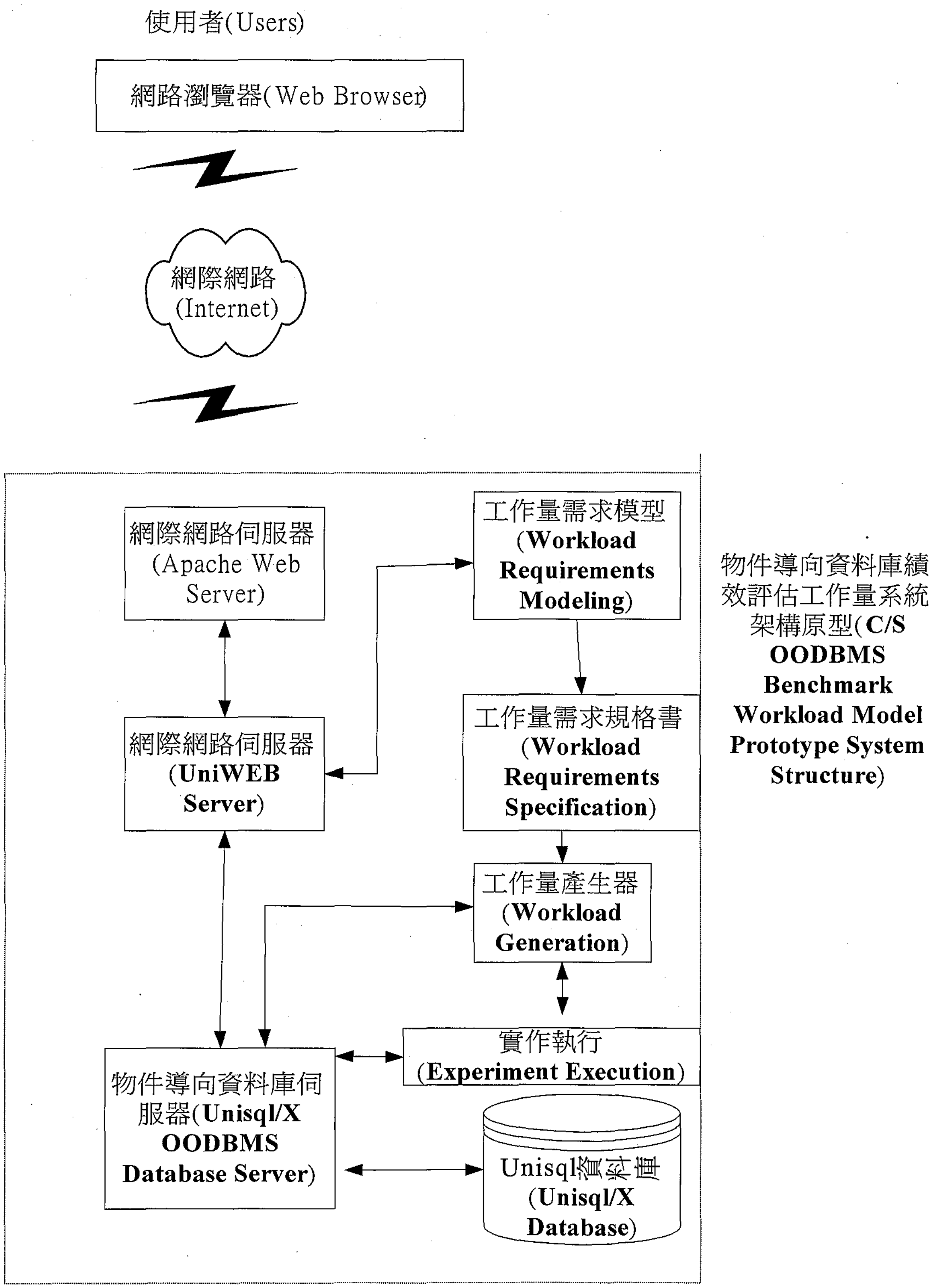

圖6：測試平台實作系統架構圖

Please input class schema according to syntax specification.

DEFINE OBJECT CLASS FOR :

ATTRIBUTES :

syntax --attribute_name data_type , attribute_name data_type ,...

OPERATION :

syntax --operation_name(arg_data_type) return_type,operation_name(arg_data_type) return_type,...

END OBJECT CLASS.

Submit Reset

圖7：類別編輯器的畫面

If you want to check the result of schema, press
Check

If you want to insert row into class, press
Insert

圖8：類別資料傳遞後的詢問畫面

ATTRIBUTES
customer_id:integer
product:character(20)
quality:integer
order_date:character(20)
deadline : character(20)
OPERATION
Ready_produce (character(20))

If you want to insert row into class, press

insert

圖9：個別資料綱目的查詢結果圖

customer_id	integer
product	character(20)
quantity	integer
order_date	character(20)
deadline	character(20)

class name :

syntax-->numeric,'string','character',...

Submit Reset

圖10：類別的實例輸入圖

this is the result of insertion

this is OID of the row inserted

@260|1|0

this is all objects in order sheet

customer_id	Product	quantity	order_date	deadline
123	chip	10000	26-4-99	26-8-99

圖11：類別物件的完成圖

(三)單一交易編輯器

單一交易編輯器主要的功能是用來讓使用者表達資料模型內的某一個類別物件，因某種原因或事件，所造成的一連串狀態的轉移過程，其單一交易編輯器的輸入畫面如圖 14 所示。

(四)複合交易編輯器

複合交易編輯器可以讓使用者編輯資料模型內不同物件間的訊息傳遞關係，以達成邏輯上某種的交易目的，圖 15 則是複合交易編輯器的編輯畫面。

(五)物件版本控制編輯器

物件版本控制編輯器主要的功能是讓不同的使用者對於相同的類別物件 能夠針對自己的需求，來編輯符合自己需求的物件內容 (content)，圖 16 是物件版本控制編輯器的編輯畫面。

(六)控制模型編輯器

控制模式編輯器主要能夠讓使用者對於已輸入的交易，指定各交易在績效評估測試時，其交易的順序、次數、種類、模式、與交易的時間長短及績效衡量的單位等等，圖 17 則是控制模式編輯器的編輯畫面。

(七) 執行績效測試

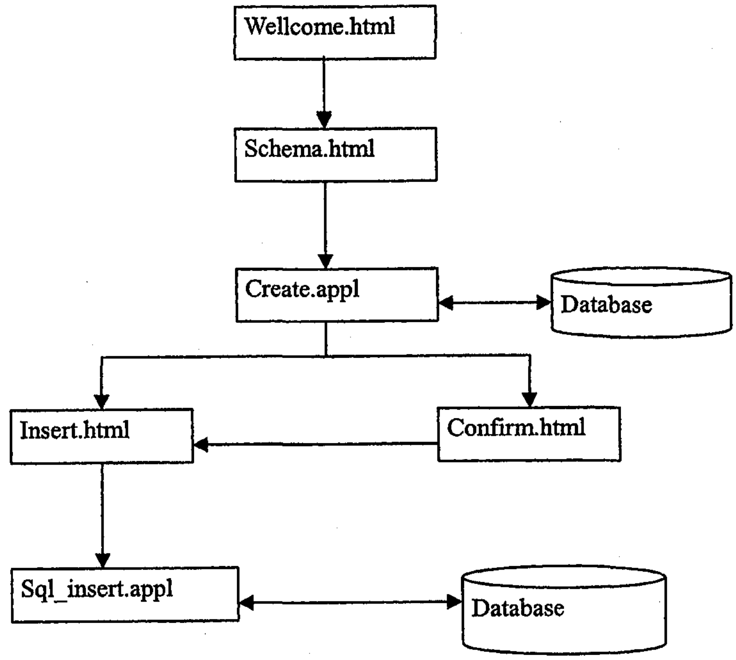

圖12：類別編輯器的程式結構圖

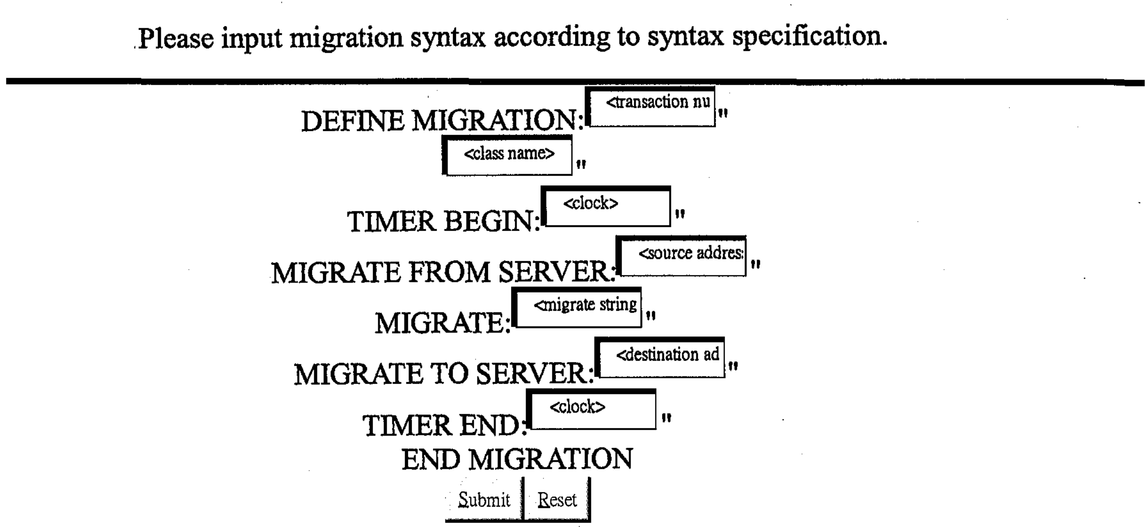

圖13：物件遷移編輯器

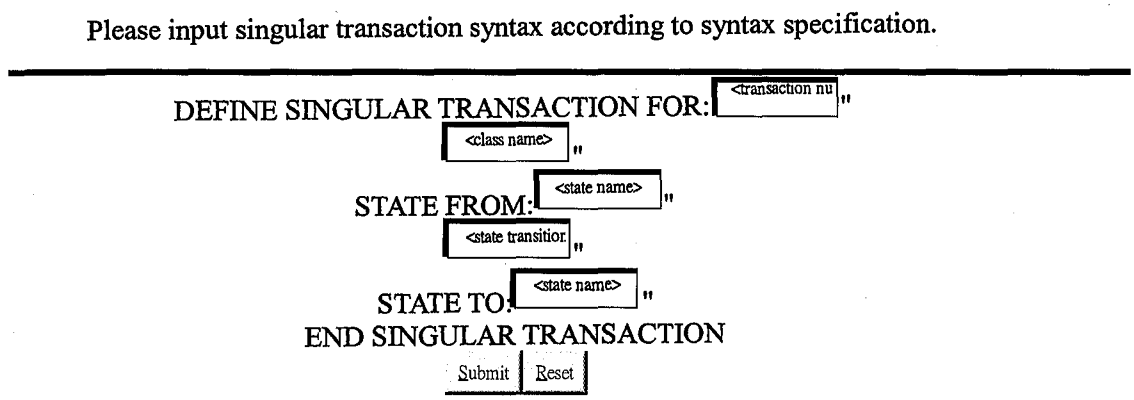

圖14：單一交易編輯器

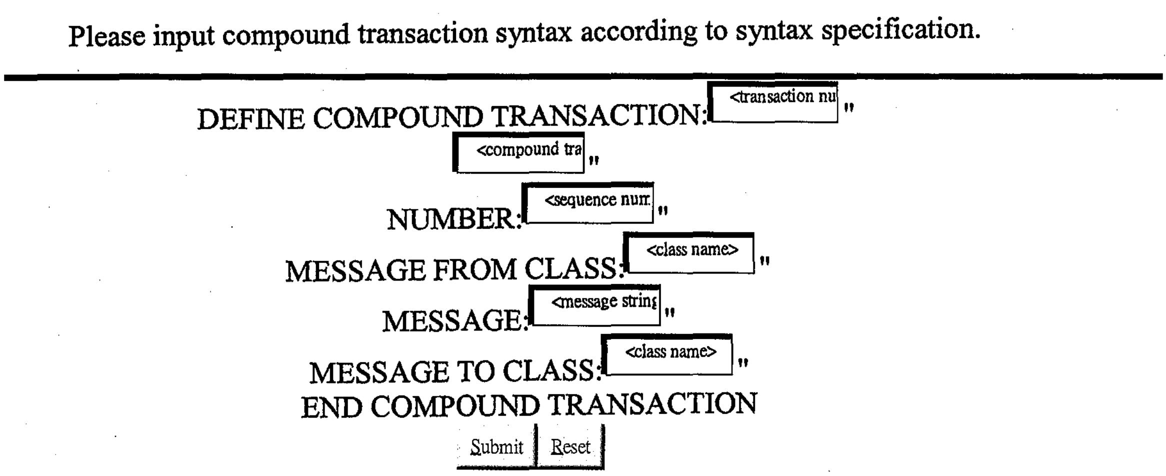

圖15：複合交易編輯器

圖 18 爲績效測試之設定畫面，系統先從物件資料庫中讀取客戶訂單所建立之測試交易，並按照交易順序依次排列在此畫面中。然後，使用者輸入測試編號與測試名稱，並可點選所要執行的測試交易，然後按下「執行測試」按鈕，系統就會依次根據使用者所點選的交易，以及此交易的執行重複次數，分別依序執行測試，並分別紀錄每筆交易測試的系統回應時間，等全部測試交易皆測試完成之後，系統會傳回測試結果如圖 19 ，並將此結果存入 Benchmark Result Table 裡面。

(八)測試結果蒐集器

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 ，使用者只需點選欲瀏覽的測試名稱，然後按下執行查詢按鈕。系統即會傳回該測試名稱的測試結果，如圖 21 。

以上以一簡易案例：客戶訂單物件資料庫，顯示(1)如何由類別編輯器輸物件綱要，可以查詢物件類別；(2)如何輸入案例資料，查詢內容的結果；接著，(3)如何由物件類別遷移編輯器輸入物件遷移交易邏

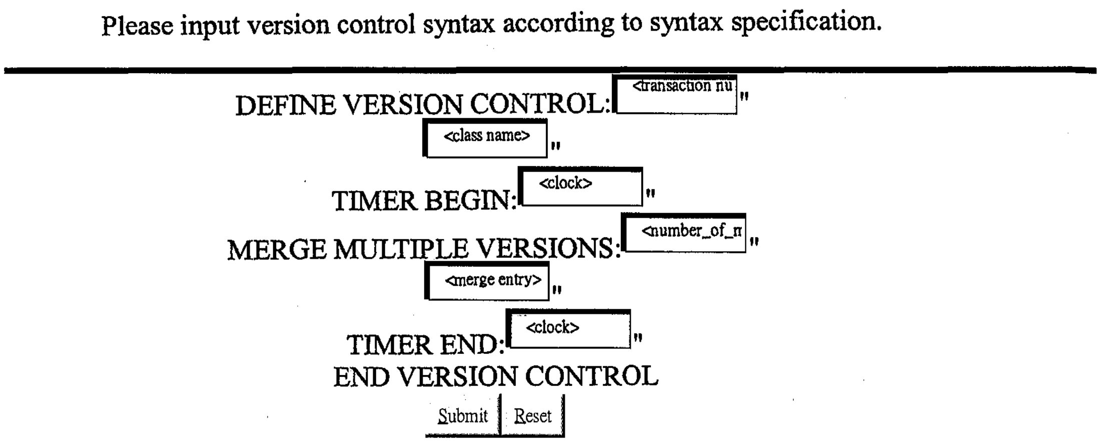

圖16：物件版本控制編輯器

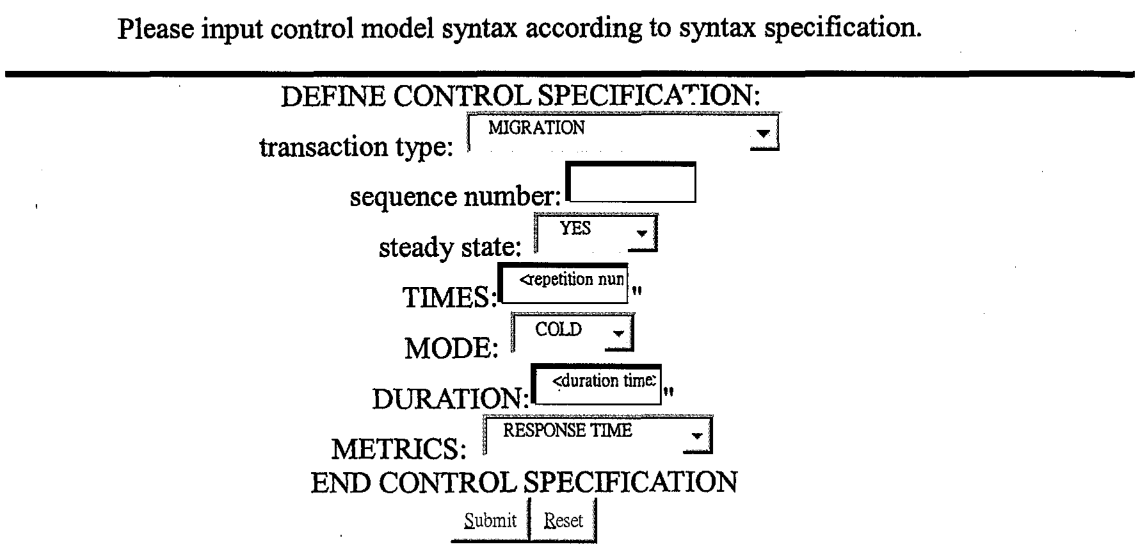

圖17：控制模型編輯器

輯；(4)如何由單一交易編輯器輸入單一交易邏輯；(5)如何由複合交易編輯器輸入複合交易邏輯；然後(6)如何由版本控制模型編輯器輸入版本控制邏輯；最後(7)如何由控制模型編輯器輸入控制邏輯。執行測試的設定畫面，可以設定實驗順序和類型，執行結果可以經由查詢畫面看到回應時間、測試名稱、測試次數、測試順序等簡易和基本資料。

伍、結論與未來的研究方向

一、結語和討論

績效評估在新興科技廣為使用時扮演成敗關鍵角色，主從式物件資料庫系統在網際網路平台、無線運算技術和全球資訊網資料庫演進中一種基礎技術。績效評估可以說明新興科技對我們的益處是什麼？

Execute Benchmark Test

請輸入此測試的基本資料...

測試編號： a6

測試名稱： Test Function, Array, and REF

請選擇要測試的交易名稱...

- [x] use ADT Function
- [x] select Array Type
- [x] select REF Type
- [ ] update Customers

執行測試

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

Display Result

請選擇欲瀏覽的測試名稱：insert into Customers

執行查詢

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

如何應用這項新興科技？它可能的優點和缺點是什麼？如何善用其長處避免其短處？目前 001 、 007 、 BUCKY 是標準物件資料庫績效評估，以預定方式只提供參考性，不能客製化使用者工作量，數據不代表應用環境眞實績效。本研究因此在工作量模型，提出一項新的思維，將使用者工作量需求以標準載具表達出來，經過系統化和模式化推導。將不同使用者的不同需求，產生不同規格書，建置不同測試套組，產生代表使用者工作量。期望突破領域專屬限制、突破績效評估的數據僅能做爲參考不能告訴我們準確結果限制；期望客觀、一致、正確的實際物件資料庫工作量，能夠告訴使用者。在本文中，我們提出了以物件導向系統分析方法來建構使用者績效評估的工作量模式。本研究使用文字型態的描述來對應圖形型態的工作量模型，採用 UML 模型語言，加以延伸，建立主從式物件資料庫的資料模型表示語法、交易模型表示語法、和控制模型，提供跨平台的工作量模型表示方法。

二、未來研究方向

未來研究工作可以朝下列的方向進行：

1. 建構大型和多使用者的工作量。現今資料庫績效評估之研究往往只偏向單一使用者的工作量作測試，大型多個使用者的工作量構面廣泛複雜，牽涉到多個同時存取交互作用，未來可朝這方向來研發。
2. 建構文字型態物件模型轉換成績效評估之程式產生器。由於本研究對於將目前各標準績效評估方法文字型態工作量模型轉換成實際可執行的績效評估測試程式未加以驗證，形成本研究限制，是否符合成本效益？亦應該進一步探討。

參考文獻

1. Burton, F. Bradley and Marek, W. Vivtor, "Application of JAVA Programming Language to Database Management" ,ACM SIGMOD record ,vol 27,no 1, March 1998 ,pp.27-34
2. Bort, Julie, " Entering ORB-it ", Software Magazine, January 1997, pp.90-93
3. Balasubramanian, V., Bang min ma and Joonhee Yoo, "A Systematic Approach to Designing A WWW Application", Communication of ACM, Aug 1995,vol 38 no8
4. Booch, G., Object-Oriented Analysis and Design with Applications, 2nd edition, Benjamin Cumminga, 1994.
5. Chang, W. K. "A Quadratic Programming Approach to Usage Modeling for Software Reliability Certification," Tunghai Journal, Vol.38, pp. 65-78, July 1997.
6. Craig, Thompson, "Response from Object Service and Consulting, Inc to the OMG Internet Services Request for Information", Object Services and Consulting Inc,14 Oct 1996.
7. Carey, J. Michael, David J. DeWitt, Jeffrey F. Naughton, Mohammsd Asgarian, Paul Brown, Johannes E. Gehrke, Dhaval N. Shah, "The BUCKY Object-Relational Benchmark", ACM SIGMOD 1997, pp.135-146
8. Carey, J. Michael, David J. DeWitt, Jeffrey F. Naughton, "The 007 Benchmark", Computer Science Department,University of Wisconsin-Madison, Version of April 12 1993.
9. Carey, J. Michael, David J. DeWitt, Jeffrey F. Naughton and Chander Kant, "A Status Report on the 007 OODBMS Benchmarking Effort", In Proceedings of the ACM OOPSLA Conference , Portland, OR, Oct 1994,pp.414-426.
10. Carey, J. Michael, David J. DeWitt, Jeffrey F. Naughton, Mohammsd Asgarian, Paul Brown, Johannes E. Gehrke, Dhaval N. Shah, "The BUCKY Object-Relational Benchmark", ACM SIGMOD 1997, pp.135-146.
11. Cattell, R. G. G., "Object Data Management -- Object Oriented and Extended Relational Database Systems", Addison Wesley Publishing Company, Inc., 1994
12. Cattell, R. G. G. and J. Skeen, " Object Operation Benchmark ", ACM Transations on Databases Systems, vol 17, no 1, March 1992, pp.1-31
13. Dyer, M., Tlie Cleanroom Approach to Quality Software Development, John Wiley & Sons, 1992.
14. Doppelhammer, J., Thomas Hoppler, Alfons Kemper and Donald Kossmann, "Database Performance in the Real World--TPC-D and SAP R/3",ACM SIGMOD 1997, pp.123-134
15. DeWitt, "Overview of Wisconsin Benchmark," Benchmark Handbook, 2nd Ed., Morgan Kaufmann, 1993
16. Dey, Debabrata, Abraham Seidmann, "Benchmark Decision Models For Database Management Syatems", Information Systems Research, vol 5, no 3, 1994.pp.175-293
17. Fowler, Martin and Kendall, Scott, "UML Distilled", Addison-Wesley,1997.
18. Francett, B., "Object Database and the Web :A Dream Team?", Software Magazine Oct 1996, pp.90-96

19. Gray, Benchmark Handbook, 2nd Ed., Morgan Kaufmann, 1993
20. Hughes, G. John, "Object-Oriented Databases", Prentice Hall International Ltd. 1991
21. Heuser, Lutz, Dilley, J., Madduri, H., Steven Rabin and Shawn Wood, "Development of Distributed and Client /Server Object-Oriented Application: Industry Solution ", OOPLSA'94, pp.317-323.
22. Hamlet, D., "Are We Testing for True Reliability?" IEEE Software, pp. 21~7, July 1992.
23. Hurson, A. R. and M.W.Bright, Object-Oriented Multidatabase Systems, Chapter1, pp.1-36
24. Leutenegger, T. Scott and Daniel Dias, " A Modeling Study of The TPC-C Benchmark", ACM SIGMOD 1993, pp.22-31
25. Musa, J. D., "Operational Profiles in Software-Reliability Engineering," IEEE Software, March 1993, pp. 14-32.
26. OMG, "UML Standard Specification", 1997, http://www.omg.org.
27. Rational Software Corporation , Unified Modeling Language, 1997.
28. Stonebraker, M., J. Frew, K. Gardels, and J. Meredith, "The Sequoia 2000 Storage Benchmark," Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, May 1993, pp. 2-11.
29. TPC-W, Transaction Processing Performance Council, TPC Benchmark W Standard Specification , 1999.
30. Wohin, C. and P. Runeson. "Certification of Software Components," IEEE Transactions on Software Engineering, June 1994, 20 (6), pp. 494-499.
31. Woit, D., "A Framework for Reliability Estimation," Proceedings IEEE 5th Int. Symp. on Software Reliability Engineering, Monterey, California, USA, pp. 18-24, 1994.
32. Whittaker, J. A. and J.H. Poore. "Markov Analysis of Software Specifications," ACM Transactions on Software Engineering and Methodology, January 1993, vol. 2(1), pp. 93-106.
33. Whittaker, J. A. and M. G. Thomason. "A Markov Chain Model for Statistical Software Testing," IEEE Transactions on Software Engineering, October 1994, 20 (10), pp. 812-824.
34. Yang, Jingshuang Jack and Kaiser, E. Gail, "An Architecture for Integrating OODBs with WWW", Computer Networks and ISDN Systems, 28(1996), pp.1243-1254
35. Yu, S. Philip, Ming-Syan Chen, Hans-Ulrich Heiss, and Sukho Lee, "On Workload Characterization of Relational Database Environments", IEEE Transactions on Software Engineering , vol 18, no 4, April 1992, pp.347-355