

物件導向方法論於 Web-based 應用系統設計 與績效評估之研究

楊欣哲、高國峰
東吳大學資訊科學系

摘要

近年來由於全球資訊網 (World Wide Web; 以下簡稱 Web) 技術快速發展及應用普及化, 所以企業如何在 WWW 的環境下有效率地建置適合的 Web-based 應用系統就是一項重要的課題。目前, 有許多物件導向資訊技術已經針對 WWW 架構下提供 Web-based 應用系統整合資料與行為之基礎。由於 WWW 具備的特性有: (1) 所有資訊的呈現都是透過網頁 (Web pages) 的型式來展現, (2) 網頁與網頁間呈現出脈絡交錯的連結關係。因此如何將網頁與網頁間的連結關係具體地於系統設計階段表現出來, 對於 Web-based 應用系統的開發相當重要的。同時 Web 應用系統之設計模型能於系統設計階段即施行績效評估, 將可在早期即對 Web-based 應用系統提供診斷與品質量測並可提供未來系統設計之參考, 如此之技術整合對複雜的 Web-based 應用系統開發將是迫切且成效顯著的。有鑑於此, 本研究首先運用關係管理結構法 (RMM: Relationship Management Methodology) 整合至物件導向分析與設計 (OOA/OOD) 中, 以及使用 UML (Unified Modeling Language) 工具加以延伸擴充所分析出的網頁關聯結構, 再搭配三層次 MVC (Model-View-Controller) 與快速雛型法 (Prototyping) 等方法以設計 Web-based 應用系統物件模型。其次, 應用 QMOOD (Quality Model for Object-Oriented Design) 技術並考慮 WWW 的特性對 Web-based 應用系統設計模型進行績效評估並作品質度量, 使得系統之物件模型結構可有效地量化以檢視其設計模型。本研究所建議之物件導向方法論是結合了物件模型設計與模型品質測量技術, 將更有效地建置 Web-based 應用系統。最後, 將實際建立一 Web-based 應用系統實例以驗證其方法之可行性與實用性, 並且進行績效評估以提昇其效益與效率。

關鍵詞：物件導向、關係管理結構法、快速雛型法、物件品質模型、
Web-based 應用系統

Design Issues and Performance Evaluation for Web-based Applications using Object-Oriented Methodology

Shin-Jer Yang · Kuo-Feng Kao

Department of Computer & Information Science, Soochow University

Abstract

Due to the technology advances and high popularity of the Web-based applications, how an organization to efficiently implement Web-based applications is very essential to WWW environment. Currently, some object-oriented technologies are to develop for further integrating with the WWW infrastructure, and also to provide the basis for combining data and behavior of Web-based applications. To address some WWW characteristics: (1) all information on the WWW is presented through Web pages, (2) Web pages are inter-connected, so it is required to clearly identify the relationship and presence of the Web pages in order to develop Web-based applications efficiently. Meanwhile, the design models of Web-based application can evaluate the performance at the early design stage. Therefore, these models provide diagnosis and quality metric for future references in designing Web-based application. The integration of the technologies is imperative and significant to the development of the complex Web-based applications. First, we can integrate OOA/OOD (Object-Oriented Analysis and Design) and RMM (Relationship Management Methodology) via utilizing UML (Unified Modeling Language) and 3-tier MVC (Model-View-Controller) to design the models for prototyping the Web-based applications. Then, we propose quality metrics based on QMOOD (Quality Model for Object-Oriented Design) with considering WWW properties. The purposes of this metrics model are to quantify and validate the design model structures of the Web-based applications. The proposed object-oriented methodology integrates object model designing and quality metrics to implement and assess the Web-based applications. Finally, we set up a practical instance of the Web-based application to verify its feasibility and usability. At the same time, we make the performance evaluation to validate its effects and efficiencies.

Keywords: Object-Oriented, RMM, Prototyping, Object Quality Model, Web-based application

壹、導論

近年來由於全球資訊網 (World-Wide-Web) 之技術快速發展以及應用日益普及，應用軟體系統的作業平台已逐漸地移至以 Web 為基礎的平台環境來建置，究其因主要是因為其跨作業系統平台、用戶端需求較有彈性和標準的操作介面以及便於集中的管理維護等優點。

一、Web-based 應用系統開發相關文獻探討

但是 Web 系統起初設計目標是作為資訊相互傳遞的媒體，Web 系統的資訊內容是以網頁 (Web pages) 的方式呈現，而內容的交互參考資訊則是以超連結 (Hyperlink) 的方式來表示。使用 Web 平台建置之應用軟體系統 (以下簡稱 Web-based 應用系統) 與一般作業平台上建置之應用軟體系統的主要差異性也在於必須考慮 Web 平台的超媒體特性，所以 Web-based 應用系統也稱為超媒體應用系統 (Hypermedia application) (Garzotto 93)。現今網頁開發技術的標準已由 HTML 演化成為 XML (Extensible Markup Language) (Bary 98)。XML 與 HTML 主要的差異在於 XML 沒有固定的標籤集(tag set) (Bary 98)，可允許使用者透過 XML-Data 可自行的定義 XML 文件結構(Layman 98)，再透過 XSL (Extensible Stylesheet Language)可使 XML 能更有彈性地展現文件的內容 (Clark 98; Decker 00)。設計 Web 應用系統除了網頁技術之外，更需要有妥善的系統開發方法以確保系統能順利地進行設計與建置，可是傳統應用系統的開發方法並不能有效地反應 Web-based 應用系統的超媒體特性(Garzotto 93)，因應 Web 應用系統的超媒體特性，在傳統應用系統設計時可整合下列兩種方法來加以改善，以反應其超媒體的特性：(1) 以 Web 的超媒體 (Hypermedia) 特性為基礎的結構化開發方法即 - 關係管理結構法 (RMM: Relationship Management Methodology) (Isakowitz 95)，其提供以結構化的方法改善其他軟體設計方法上對導覽設計的缺陷；(2) 物件導向為基礎的設計方法如 - 物件導向超媒體設計(OOHDM: Object-Oriented Hypermedia Design Model) (Gaedke 99)，其提供以物件模型的方式呈現出導覽前後關係 (Navigational context)、使用 XML 的 Meta Data 延伸的 WebComposition 模型與使用 UML 工具來建構 Web 應用系統的 Jim Conallen 設計模型(Conallen 00; Gellersen 99)。

綜觀結構化設計與物件導向設計兩種設計方式，可歸納出 Web-based 應用系統開發時應具備有三種的觀點來說明所建置的系統模型分別是(Rossi 97)：(1) 概念模型 (Conceptual model)：說明應用系統的概念上的組織與架構模型此可以是獨立於 Web 系統建置技術的模型、(2) 導覽模型 (Navigational model)：反應 Web 平台上超連結相互參考關係的模型與 (3) 介面模型 (Interface model)：此為實際與網頁製作相關的模型。而目前 UML (Unified Model Language)工具已是軟體業界公認的物件模型圖示標準 (Larman 98)，透過 UML 工具可以有效地表示物件模型。對於超媒體特性而言，以上所述之系統模型中如何有效地建立導覽模型是建置 Web 應用系統的主要關鍵。然而

XML 的 DOM(Document Object Model)規範利用繼承的樹狀結構建立 XML 文件中介面(interfaces)間的關聯性，可以 XML 的技術比 HTML 更有效地將物件模型轉化成 XML 格式以縮小物件模型與網頁建置技術的差異(Decker 00)。

二、軟體系統品質評量技術相關文獻探討

對於超媒體特性而言，以上所述之系統模型中如何有效地建立導覽模型是建置 Web-based 應用系統的主要關鍵。然而 Web-based 應用系統的績效評估主要是希望於早期即能針對其系統的相關特性予以測量評估，以提供系統改善與未來應用的依據。這樣的需求可以透過測量系統品質的技術來達成，而此類技術之原理大都是先設定欲測量之品質屬性 (Quality Attributes)，其次建立品質模型或預測模型以決定品質屬性與系統之關聯性，最後決定系統中實際可量化之品質度量值 (Quality Metrics)。近年來 Web 應用系統的品質測量技術亦逐漸的被廣泛研究，然而目前相關研究大都以 Web 應用系統中的超媒體特性來做為品質測量的目標，此類技術有以 Cocomo II 測量模型演進而來的 WebMo (Web Model)測量技術以及建立預測模型後利用統計方法 MMRE (Mean Magnitude of Relative Error)的測量技術(Medes 01; Reifer 00)。WebMo 技術中將視 Web 應用系統由眾多的 Web 物件所構成，品質度量是經由計算 Web 物件相關數量來獲取，其利用 Cocomo II 與 SoftCost-OO 軟體成本預測模型並定義九項影響成本的因素：產品穩定與複雜度、Web 平台差異性、個人能力、個人經驗、開發工具便利性、時程管理、重複使用、團隊工作與處理效率。WebMo 品質評量技術可針對不同的 Web 系統應用領域如：電子商務、財務/貿易應用、B2B 應用與 Web 的資訊管理系統，使用不同的特性參數以修正不同應用領域對於系統品質度量的影響。接下來利用 MMRE 度量技術的品質度量方法依據以往專案或專案中早先開發階段來獲取經驗性的觀察資料，辨認尺度衡量標準與成本的導因並且將之間影響的關係公式化，最後確認預測模型的有效性。此品質度量方法中依據五類度量值(Metrics)：專案長度(Length size)、重複使用性(Reusability)、複雜度(Complexity size)、影響(Effort)與困惑因子(Confounding factors)，收集不同專案實例的量化資料再經由預測程序(Estimation process)予以計算品質屬性值。

上述兩種技術可直接由網頁的超連結與使用的媒體數量獲取模型的所需的測量數據，因此可以直接對既有 Web 應用系統測量其品質。但是其也有兩項缺點：(1) 測量技術中部分度量值量化不易，如開發人員能力、開發人員經驗以及開發工具的便利性等，並不容易精確量化因此不利於自動化品質測量以及 Web 應用系統開發前期的使用性；(2) 測量技術並無法針對物件導向技術之模組性、再利用性、可維護性等特性建立有效地測量模型。而以物件導向為基礎的 QMOOD (Quality Model for Object-Oriented Design)品質測量技術(Bansiya 02)，其特色有三：(1) QMOOD 以 ISO 9126 為基礎設定品質屬性(Dromey 96)，其屬性涵義能廣泛的為現有系統開發人員所熟知；(2) 此測量技術並不納入人員能力、人員經驗以及開發工具等度量值量化評估，其所有的度量值取得可由系統模型或原始碼，因此可於系統設計期間或實作期間執行

測量並且適合製作自動化的品質測量工具系統以利測量的執行；(3)其可針對物件導向所設計之應用系統測量其模型的再利用性、彈性、可瞭解性、功能性、延展性與有效性，故此技術適合作為 Web 應用系統物件模型之品質測量方法。

因此本篇論文將提出一套整合系統設計方法論以開發 Web-based 應用系統，亦即以物件導向技術為基礎並整合關係管理結構法 (RMM: Relationship Management Methodology) 並整合物件導向技術以有效地設計出 Web-based 應用系統導覽模型與延伸擴充 UML 工具以說明其模型，之後整合物件導向品質測量技術 QMOOD 以期能在早期即可針對模型的品質予以評估，其技術可直接由設計模型或雛型系統中獲取測量數據，驗證系統的有效性以及應用循環式 (Iteration) 物件導向開發技術改善系統開發品質。本文將針對所述研究分為五個部分介紹，首先討論 Web 應用系統之設計方法現況與其整合物件導向技術之原理及方法。第二章將說明物件導向設計技術導入於 Web 應用系統設計之整合細節，且提出導覽類別圖由原 UML 中類別圖擴充以因應 Web 應用系統特色。第三章為 Web-based 應用系統的實例探討，將以公共事業災害管理系統為例說明導入物件導向技術於 Web-based 應用系統設計以及導覽類別圖之應用。第四章針對整合物件導向技術所建置之 Web-based 應用系統提出績效評估方法，此章中首先將說明 QMOOD 品質度量技術原理與應用於物件導向 Web-based 應用系統之品質測量方法。第五章則以前章所建置之災害管理系統為例說明利用品質屬性度量方法進行設計模型品質評量，進而驗證設計方法的有效性，程序上將採用循環式系統開發技術將比較系統演化前後之 Web-based 應用系統的品質屬性度量。最後為結論將總結本研究之整體成果與效益，以及提出未來研究方向。

貳、Web 應用系統原理與方法

本章將概要地介紹 Web 應用系統的設計原理，首先將對傳統的物件導向設計與結構化 Web 設計作一簡單地說明，最後將整合此二者設計方法進而提出物件導向的 Web 應用系統設計觀念。

一、物件式導向設計與結構化 Web 設計現況探討

目前應用系統建置已逐漸使用物件導向方法來進行設計，此種設計方法的主要特徵就是透過物件與物件間互動的模型來建立應用系統。原則上，經由物件導向設計方法可以歸納出三種物件類型以建立應用系統架構，此三種物件類型分別為：(1) 實體物件 (Entity Object)：用於定義出系統中的資訊、(2) 介面物件 (Interface Object)：用於定義系統中與介面相關地行為與資訊、(3) 控制物件 (Control Object)：定義不屬於上述兩者之功能行為。由此三者物件類型所建構之物件導向應用系統可以達到有效的功能分配，使應用系統具模組性、再利用性與可維護性。近年來，由於應用系統須使用並資料庫系統並加以連結，所以應用系統的開發上須將其系統的物件模型加入資料庫物件 (Database Object) 類型物件，再將實體物件與控制物件整合成企業物件

(Business Object) 類型，然後再配合介面物件類型物件以達成其整體系統功能，這樣的物件模型也就是現今應用系統設計時所經常使用的三層式 (3-tier) 應用系統架構或稱為 Model-View-Controller 模型 (以下簡稱為 MVC 模型)。上述所簡介的物件導向應用系統模型架構，其目的是為了使應用系統達成模組性、再利用性、可維護性、資料與程序之有效地進行功能分配等優點以利於應用系統開發之需求及效益。

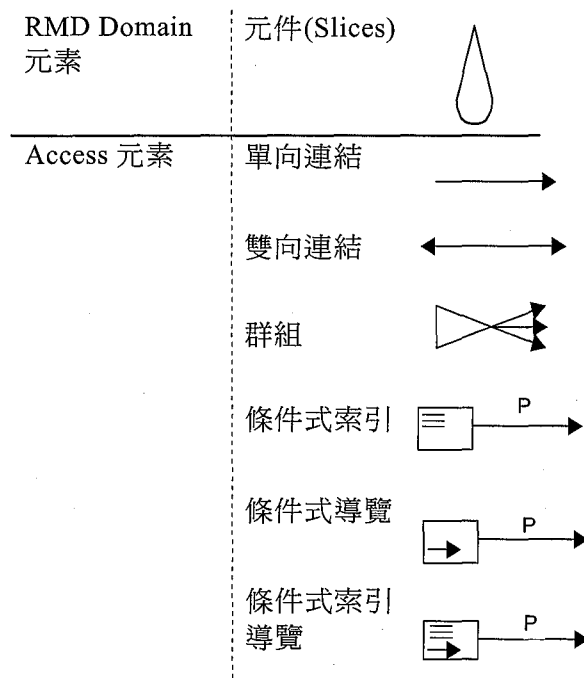


圖 1：RMM 方法論物件導覽模型基本圖示

傳統上 Web 應用系統的設計目標只是為了提供資訊傳遞的媒體，所以如何設計網頁資訊的內容便成為 Web-based 應用系統設計主要的問題。而關係管理結構法 (RMM) 主要是針對超媒體的一種結構化設計方法，藉由關係管理資料模型 (RMDM: Relationship Management Data Model) 來表達 Web-based 應用系統中的超媒體導覽模型，RMDM 主要是由一些基本圖示所組成 (如圖 1 所示)。經由 RMM 方法建立的 RMDM 模型可表達 Web 應用系統中網頁的導覽模型，使用 RMM 方法設計 Web 應用系統分為下列三個步驟：(1) E-R 設計：此階段中以需求定義文件開始一般的實體關連模型 (Entity-Relationship Model) 的設計，其產出為 E-R 圖 (2) 元件 (Slice) 設計：此階段中將前一階段中的實體 (Entity) 予以分割成為較小且有意義的元件，其產出為元件圖，(3) 導覽設計：此階段中將前一階段的元件連結成導覽模型，其產出為 RMDM 圖，經由 RMM 方法使設計人員能有一清晰且明確的方法以建立 Web-based 應用系統中的導覽模型。

二、Web-based 應用系統使用物件式導向設計方法

Web-based 應用系統設計時將由三個觀點來考慮其應用系統之模型。首先是概念模型，任何應用系統（不論是不是 Web 應用系統）皆是以概念模型為其核心，它反映了應用系統中物件的關係與行為。因為 Web-based 應用系統具有超媒體的特性，所以必須清楚的定義出網頁（或視為 Nodes）與其間如何瀏覽的方式（可視為 Links）。總之，概念模型可使用一般的物件式導向方法來設計。其次是導覽模型，它主要是定義一個概念模型的 View，如同 MVC 模型中對於 View 以及其關聯性的定義，這也是 Web-based 應用系統系統設計最重要的部分。最後是介面模型，主要是定義使用者介面的規格，前述兩個模型是無關於資訊系統建置實作，然而於介面模型中則是與系統建置實作有較密切相關的模型。經由上述的觀念，可以將 Web-based 應用系統的架構與物件式導向的觀念作一整合，希望藉由這樣的整合使 Web-based 應用系統開發時可利用現有物件導向方法之優點。但是 Web-based 應用系統與傳統應用系統設計主要差異是導覽模型的設計，所以對於 Web-based 應用系統而言如何能妥善地建立導覽模型是非常重要的。本文將提出一套整合式物件式導向的技術，其目的在整合物件導向相關方法與導覽模型的建立。下章將先介紹此整合式物件導向方法並探討雛型系統與說明模型改善方式，最後將使用此方法建置一 Web-based 應用系統實例以進行實証研究與效能評估。

參、整合式物件導向於 Web 應用系統

綜觀前章對於 Web 應用系統的設計原理，本文所提出之整合式物件導向 Web-based 應用系統方法是結合超媒體特性之物件導向開發技術。首先針對本方法其整體 Web 應用系統開發程序作一說明，並且定義各個階段所作的活動與產出；然後將再對本方法中導覽模型的定義與架構作一闡述，並且說明 Web-based 應用系統開發流程以及實作模型的建立。原則上，整合式物件導向 Web-based 應用系統開發程序可分為五個步驟其流程如圖 2 所示，現將各步驟說明如下：

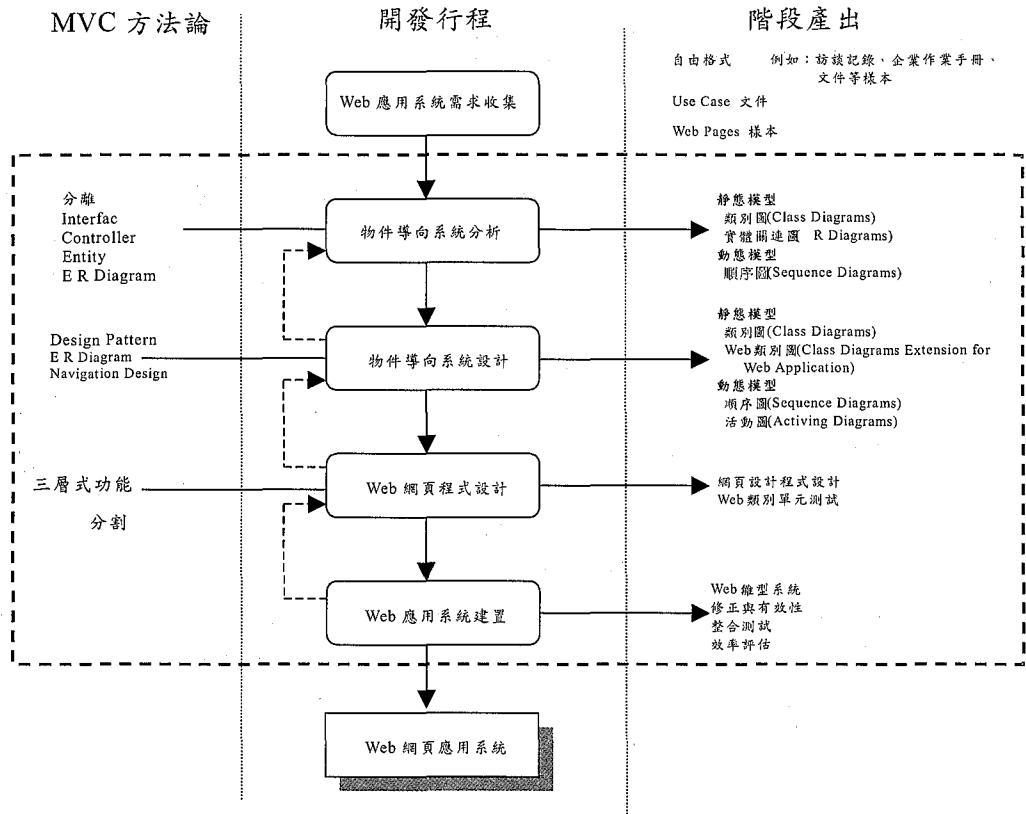


圖 2：物件導向 Web-based 應用系統開發流程

步驟一：Web 應用系統功能需求收集。此階段為 Web-based 應用系統功能定義與確認應用系統之需求，主要是經由與使用者和企業內部的專業人員訪談做為需求收集與建置 Use Case 模型的基礎(Jacobson 92)，由此建構 Use Case 功能模式結構圖與相關功能需求文件。

步驟二：物件導向系統分析。此階段將收集之需求如訪談記錄與企業內相關文件所製作而成之使用個案加以分析，依照傳統物件導向分析原理歸納出三種物件類型：實體物件、控制物件與介面物件，前兩者合稱為企業 (Business) 物件。依上述分析並整合 RMM 法將實體物件間關係規劃其關連性架構以建構成物件類別結構間的實體物件關連模型，最後將介面物件視為網頁方式表示如圖 3 所示，完成物件分析模型。

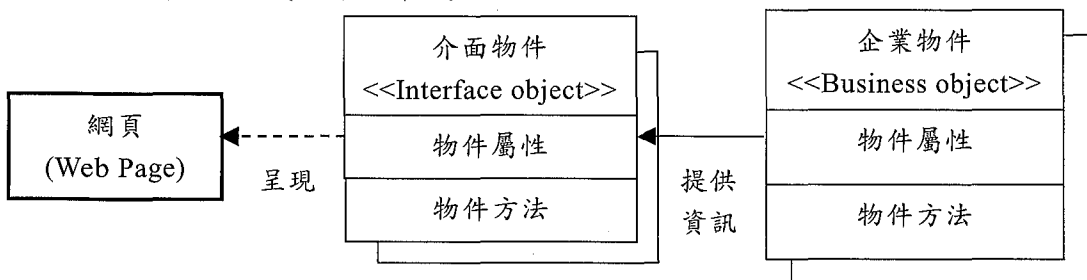


圖 3：網頁與物件模型架構

步驟三：物件導向系統設計。此階段首先將由前一階段分析出的實體模型開始，由實體物件模型中使用 MVC 方法來分割實體物件以找出元件 (Slice) 與節點 (Node) 物件(Buschman 96)，此方法主要的架構圖如圖 4 所示，圖中 Views 表示對於顯示資訊予使用者的物件。再經分析模型中的實體物件關連模型來對此元件模型進行導覽模型設計。如於設計時發覺原分析模型無法反應設計的需要時可回前一階段修正之，然後控制物件將進行細部設計。由於設計階段物件會分割成較小的物件，須經 MVC 法再確認控制物件是否已進行適當的功能分割，其管理每個 View 元件所接收到的使用者訊息，依並使用 UML 工具以有效地完成系統物件設計模型。

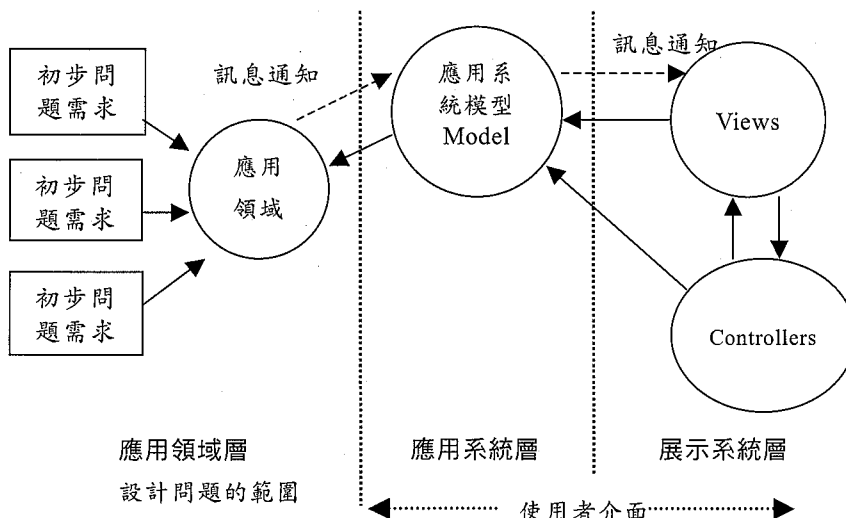


圖 4：MVC 設計方法架構圖

步驟四：Web 網頁程式設計。由設計出的 Web 類別圖，選擇適當網頁設計語言（如 ASP, Java Servlet, JSP 等）進行網頁與程式的撰寫。此階段中將對設計出的每個 Web 類別實際地建構與程式撰寫，所以此步驟對於系統的網頁編排與部署 (Deployment) 需提出妥善的規劃。

步驟五：Web 應用系統建置。在 Web 網頁程式設計完成後將完成之各 Web 類別組合成 Web-based 應用系統的雛型系統。首先，測試各整合後之單元的功能性是否完備，並驗證 Web-based 應用系統的有效性是否符合功能需求。然後，依據使用者實際的功能需求以重覆 (Iterative) 和漸進方式來增強雛型系統，並可使用物件導向測試方法來測試此雛型系統之各子單元 (Kung 00)。最後，各個子單元系統結合成 Web-based 應用系統，在此必須對系統的各種功能作交叉的測試以確保功能完整性，並且評估與調適影響系統執行的效率。

經上述五步驟依序地進行或於步驟二至步驟五之間執行模型化 (Modeling)，亦即至前階段加以調整或修正。最後，Web 網頁應用系統完成並可實際地部署於使用的 WWW 環境中。原則上，上述之方法主要著重於系統分析、系統設計、Web 網頁設計與 Web 雛型系統的建置四個階段，Web-based 應用系統與傳統應用系統設計的差異也就是導覽設計的部分，同時導覽模型是超媒體應用系統所特有的模型，這也是傳統物件式導向分析與設計方法所欠缺的。為了 Web-based 應用系統的建置時能充分展現對超媒體應用系統所特有的導覽模型的架構，將使用 UML 的類別圖並加入針對導覽所需的特性來加以擴增而成為能表達出導覽中的觀念與架構，以滿足 Web 應用系統中導覽模型建置的需求，此種類別圖稱之為 Web 應用延伸類別圖 (Class diagrams extension for Web application)。此種 Web 應用延伸類別圖與 UML 的類別圖相似，是以類別與類別之間的關聯來說明物件模型靜態架構並利用 UML 的 stereotype 的機制來加以擴充 (Larman 98)。Web 應用延伸類別圖中的圖示如圖 5 至圖 12，現針對其逐一說明：

元件 (Slices) 類別：元件類別 (圖 5 所示) 為此 Web 應用延伸類別圖中的組成基礎，然而 Web 應用系統的網頁可以由一個或一個以上的類別所構成。

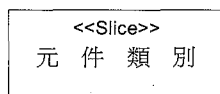


圖 5：元件 (Slices) 類別

節點(Node)類別：節點類別(圖 6 所示)主要是在表示一個具體的網頁物件，它可以由許多的元件類別組合而成或是單獨的存在於 Web-based 應用系統之中。

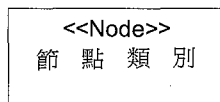


圖 6：節點 (Node) 類別

類別間存取 (Access) 關係：主要是說明 Web-based 應用系統的導覽模型，而類別間存取關係所使用到的圖示可分為三類：(1)節點類別間的關係、(2) 元件類別與節點類別間的關係、(3) 元件類別間的關係。其說明如下：

- (1) 節點類別間的關係是表示的是節點類別間導覽行為，主要是可以分為三項：條件式索引、條件式導覽與條件式索引導覽，分別說明如下：

條件式索引 (Conditional index) 如圖 7 所示：其主要是應用於節點物件間的聚合關係，有著這樣關係的兩個節點類別其主要是表示一個從屬的導覽關係如樹狀的分歧。

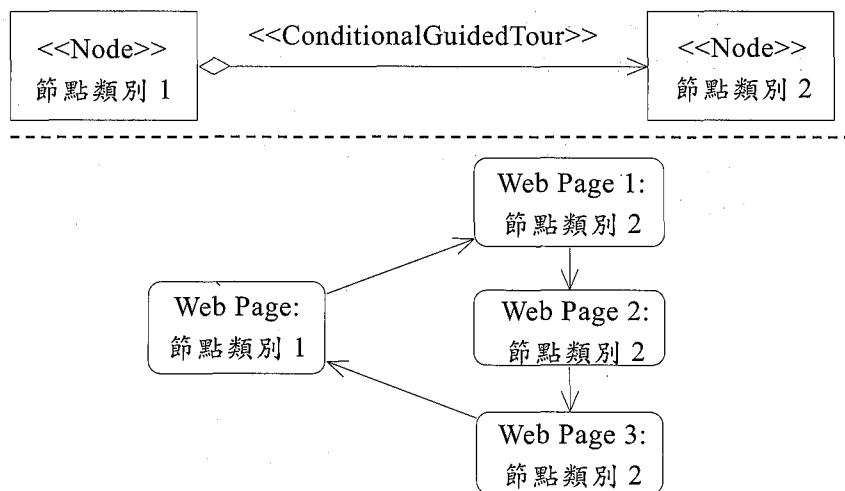


圖 7：條件式索引 (Conditional Index)關係

條件式導覽 (Conditional Guided Tour) 如圖 8 所示：主要是應用於節點物件間的聚合關係，有著這樣關係的兩個節點類別其主要是表示環狀的導覽關係。

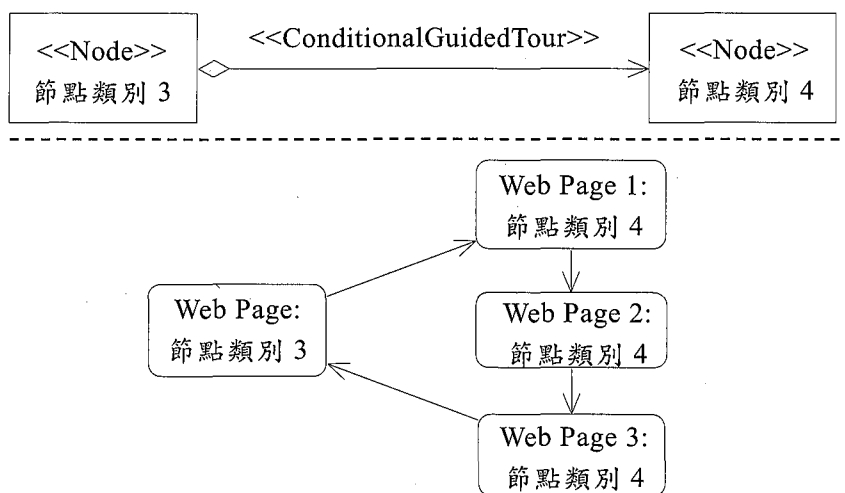


圖 8：條件式導覽 (Conditional Guided Tour) 關係

條件式索引導覽(Conditional Indexed Guided Tour) 如圖 9 所示：主要是應用於節點物件間的聚合關係，此種連結關係是結合上述兩種的導覽方式。

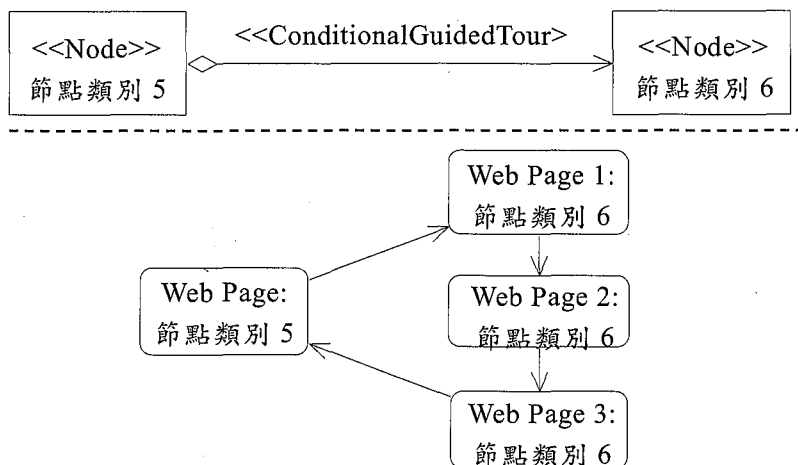


圖 9：條件式索引導覽 (Conditional Indexed Guided Tour)關係

- (2) 元件類別與節點類別間的關係主要是說明群組關係，如圖 10 所示：群組關係說明節點類別與元件類別間的聚合關係，也可實際地反應出單一的網頁物件是如何由元件類別組合而成。



圖 10：群組(Grouping)關係

- (3) 元件類別間的關係說明元件類別間的關聯性，主要分為兩種：單向連結與雙向連結。單向連結 (Unidirectional link) 如圖 11 所示：表示元件類別間有參考引用的關係存在，但是參考方向為單向時。

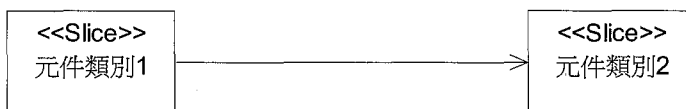


圖 11：單向連結(Unidirectional Link)關係

另外雙向連結 (Bi-directional link) 如圖 12 所示：雙向連結表示元件類別間有參考引用的關係，並且此參考關係是互相參考的關係。



圖 12：雙向連結(Bi-directional Link)關係

透過上述的圖示可將 Web-based 應用系統中導覽模型以 UML 類別圖的形式表現出來，利用既有 UML 工具的類別圖加以擴充以整合其應用系統模型所須的特殊屬性，例如圖 13 所示的訂購系統物件模型經由導覽設計後將以圖 13 所示的方式表現其導覽模型，圖 14 中訂購系統主要的網頁由訂單管理網頁所組成，其可依據使用者選定的超連結指向至採購明細網頁與產品明細網頁，而上述的兩個網頁均可分割成更小的元件類別所組成。

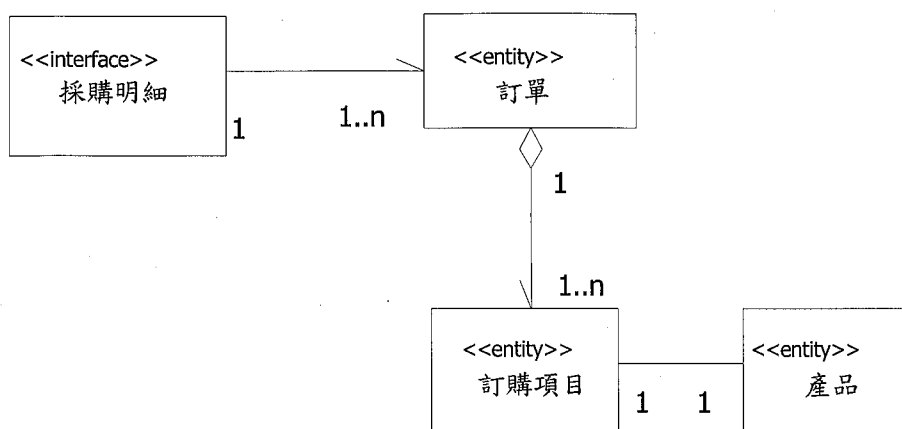


圖 13：訂購系統物件模型

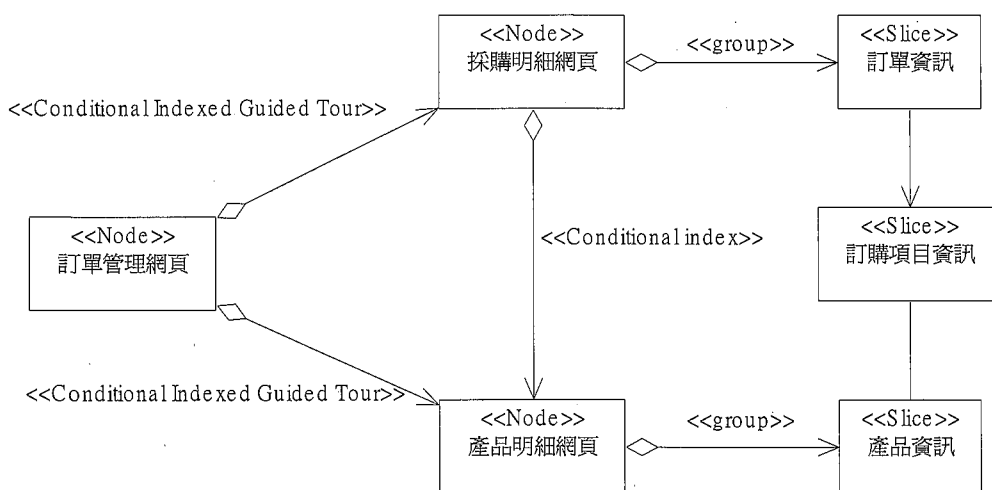


圖 14：訂購系統物件式導覽模型

本研究之設計方法主要的目標是針對 Web-based 應用系統使用物件式導向分析與設計技術能有效地建立導覽模型，但是系統開發中常會需要表示出豐富且複雜的企業規則，透過物件導向技術與搭配 MVC 方法可以快速且有效地將企業規則以物件模型表示，然而 Web 應用系統中物件的行為並不能直接的使用 Web 網頁來建置，此時將原有概念模型中所定義的物件中物件行為獨立抽離為單一的物件，而成為行為物件如 Command Pattern 的模型(Buschman 96; Gamma 95)，行為物件則可利用 Microsoft's .Net、JAVA 的 Servlet / JSP、CORBA 的技術來加以建置(Cho 97; Goeschka 01; Meyer 01)。對於導覽模型中所呈現出來的關聯性經由建置一 XML 應用程式架構可預先將條件式索引、條件式導覽與條件式索引導覽這三項關聯予以建置。導覽模型建置時由模型中元件與節點類別定義撰寫成 XML 之 DTD (Document Type Definition) 格式，而網頁物件則為此 DTD 所規範。再將網頁物件間關聯性繼承預先所建置的 XML 應用程式架構後可更有效地將模型中的網頁關聯性撰寫成 XML 格式，最後利用 DOM 物件模型將網頁物件與 Java 或相關技術設計出之企業行為物件予以整合以建置完成一離型 Web 應用系統。透過上述設計方法將有效地建立導覽模型並針對企業的行為物件予以妥善地分割後進而提升其應用系統之效率和建置更複雜的 Web-based 應用系統。

肆、Web-based 應用系統實例建置探討

本章中主要是依據上述的整合式物件導向方法論於 WWW 的環境上設計電力公司災害管理資訊系統，並實際探討此 Web-based 應用系統的相關議題以驗證此設計方法論之可行性與有效性。因電力公司災害管理資訊系統在災害發生期間需提供其公司內部災害應變中心與相關政府機關災害處理單位救災資訊，所以使用 Web 系統平台來建置 Web-based 應用系統可有效達成其災害資訊的通報與管理。此災害管理系統是必須滿足任務迫切性(Mission-Critical) 的設計、快速部署功能與開發時程縮短等三項需求，這些需求符合一般 Web-based 應用系統的特性(Reifer 00)，故本論文以此系統為實例建置探討具有其代表性。然而利用物件導向技術設計 Web-based 應用系統可提供模組性、再利用性、擴充性等優點。因此，使用物件導向技術對於災害管理資訊系統的開發將有較佳的效益。另外，實際系統的建置部分則使用 Microsoft .Net 技術，因為 Microsoft .Net 中的 ASP.NET 應用程式架構具有下列五項優點：(1) ASP.NET 中的 Web 控制元件機制提供了與非 Web 圖形介面類似的視覺導向開發方式；(2) Web 控制元件機制於伺服器端處理與瀏覽器的功能無關；(3) ASP.NET 能提供便利的用戶端狀態管理；(4) 透過 ADO.NET 使得 ASP.NET 能便捷的將資料庫內容提供於網頁內容中；(5) ASP.NET 直接使用 .Net 平台的物件模型、編譯器與執行機制(Runtime mechanisms)，使得傳統軟體開發與 Web-based 應用系統開發能整合至相同的平台。結合上述優點，將使本方法論所建構之物件模型能快速的建置於 Web 應用環境中，且重複使用既有 Microsoft .Net 技術所開發之軟體元件，達成跨程式設計語言、軟體功能分配性、易於維護與管理等優點。

Web 應用系統物件模型能有效地建立，須有兩項技術整合至設計方法中，分別為：(1) Web-based 應用系統物件模型演化技術—此技術主要提供系統模型設計時，設計人員可以依循類似的法則或經常使用的軟體樣式 (Pattern)，來調整或修正已設計出的 Web 應用系統物件模型，並視情況需要可依本方法論所提供之開發步驟，回溯前一步驟完成小幅的修改等，並能提供設計前期的模型有效性；(2) Web-based 應用系統物件模型度量 (Metrics) 技術—這對物件模型尤其重要，能將物件模型予以量化並計算預定的可度量性質，利用度量值可使設計人員更能有效地掌握系統模型特性，並能提供設計後期的模型有效性驗證。Web-based 應用系統物件模型演化可利用兩種技術，首先是 RMM 方法論中導覽設計技術。除了由小元件 (Slice) 所組成的下到上 (Bottom-Up) 的模型建立外，可針對設計出的模型按功能性逐一完成上到下 (Top-Down) 的查驗，以檢視物件的群組與階層並減少階層的深度以增進系統使用的便利性(Isakowitz 98)。在 Web 應用系統物件模型中也需要針對一對多 (One-Many) 的物件關係上再檢視條件式索引、條件式導覽與條件式索引導覽的使用並依照功能性調整所使用的物件關係以求更精進的 Web 應用系統物件模型；其次可使用超媒體的軟體樣式技術 (Buschman 96)。針對超媒體特性、導覽性設計特性以及使用介面設計這三方面進行，依照物件模型中相關的角色 (Roles) 予以調整或適度的修正原始設計的 Web-based 應用系統物件模型。而 Web 模型品質度量技術可以提供量化地方法檢視 Web-based 應用系統模型或 Web 系統建置。上述的兩種技術可於本研究設計步驟 2、3 與 4 中使用，以其提供模型設計前期與系統建置的有效性。

現針對電力公司災害管理資訊系統建置依前章所建議的 Web-based 應用系統開發流程進行說明。首先，進行 Web 應用系統需求收集步驟；此步驟主要是透過使用者的訪談並按使用者功能需求規劃，以完成災害管理資訊系統的使用個案圖如圖 15 所示。其主要系統功能分別為災害統計管理子系統、事故停電子系統、配電室搶修管理子系統、搶修人員派工管理子系統、搶修派工統計管理子系統等五項子系統，其各子系統功能間相依的關係如圖 16 所示。接下來將進行物件導向系統分析以完成系統相關之模型架構與流程。依據物件導向分析原理分析出三種物件類型，由實體 (Entity) 與介面 (Interface) 物件類型建立物件(網頁)與物件(網頁)間的實體物件關連模型。將實體物件關連模型依據 RMM 方法設計其導覽模型架構以完成 Web 應用延伸類別圖 (Class diagrams extension for Web application)，所產生延伸類別圖應用 MVC 功能分配法來確認各網頁已完成適當的資料與程序之分配。本實例中將電力用戶災害管理系統透過物件類別結構的實體物件關連模型設計以完成 Web 應用延伸類別圖，如例圖 17 所示，此圖中說明災害管理系統中災害案件管理之導覽設計模型的部分架構，其主要以案件資訊管理網頁(Node 類別)為主，其他竣工資訊、停電資訊、場室資訊、總搶修資訊與案件受理等網頁分別以條件式索引的導覽關係與案件資訊網頁連結，而圖中較複雜的網頁又可再進一的細分成元件類別(Slice 類別)，如圖中竣工案件條件、停電資訊統計與受災場室資訊等。於完成設計之後則進行 Web 網頁程式設計步驟，此步驟中利用 ASP.NET 技術建置網頁物件以落實 Web 應用延伸類別圖之導覽模型與企業行為物件，以達成具妥善軟體功能分配性之三層式電力公司災害管理雛型系統如例圖 18。電

力公司災害管理系統建置期間，可依使用者實際功能需求以循環和漸增方式來修正與改善災害管理系統功能。當災害管理雛型系統模型設計完成與雛型系統建置完成時，可使用模型品質度量技術對雛型系統進行軟體品質的相關度量，以評估模型品質與系統建置品質並作為模型修正與改善的重要依據，相關之 Web-based 應用系統品質度量技術與實例探討將於下一章中說明。最後，由物件導向導覽設計方法與品質測量技術的整合，可使物件導向技術中所強調之循環與漸進式的設計方法有清楚且易於遵循的設計與改善目標，進而於 WWW 作業環境上建置具優良設計架構與品質測量的電力公司災害管理資訊系統。

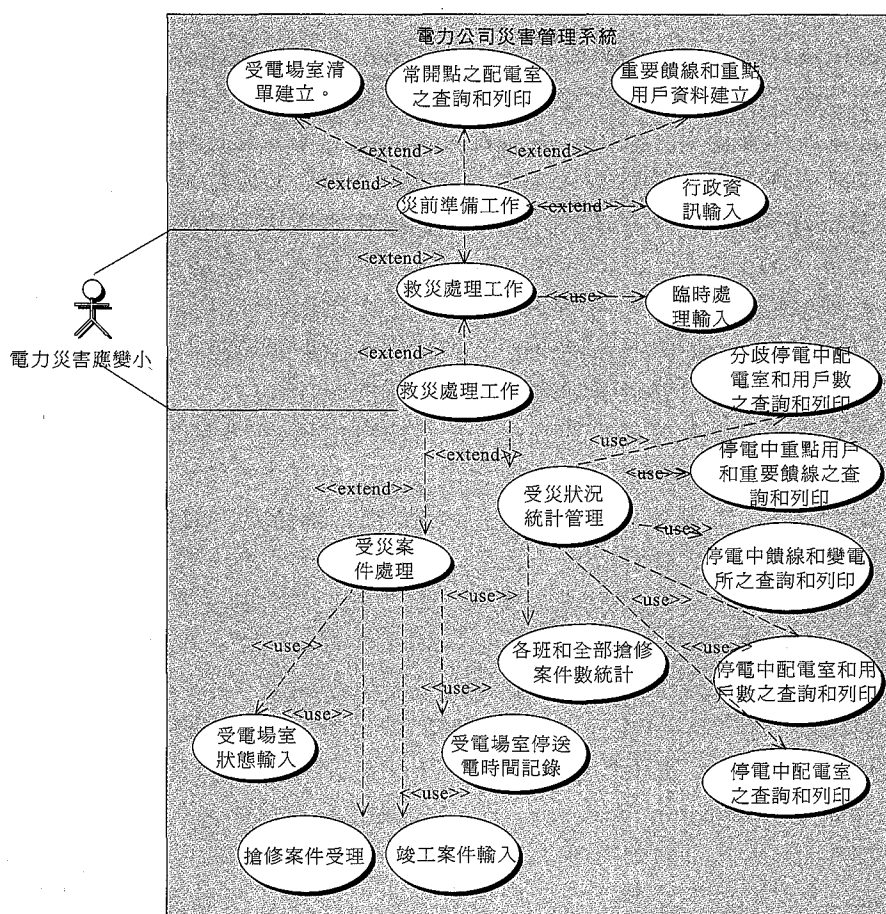


圖 15：災害管理資訊系統 Use Case 功能結構圖

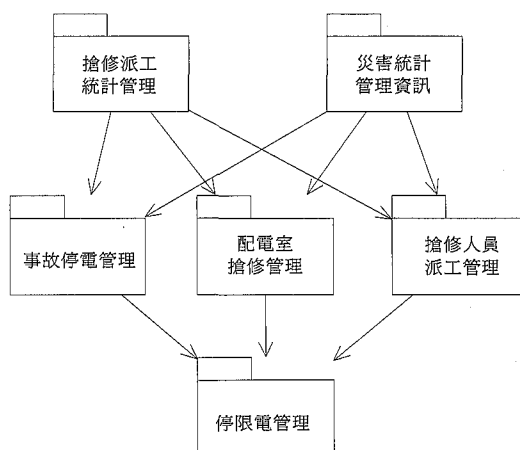


圖 16：災害管理各子系統功能相依圖

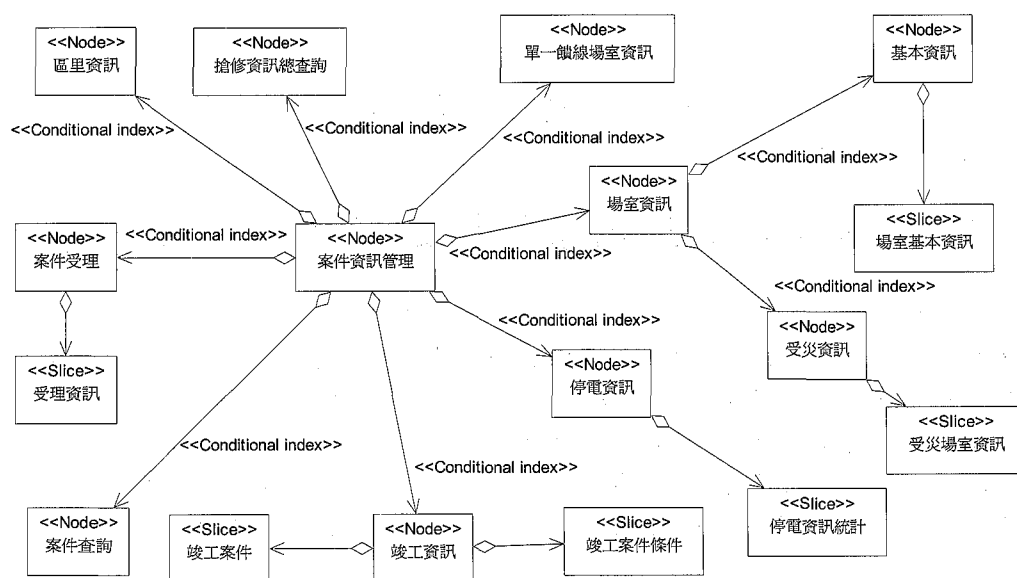


圖 17：災害管理資訊系統導覽類別圖

統物件模型合併評估以利於系統移植或協同合作。而本研究使用階層式的物件品質模型 (QMOOD : Quality Model for Object-Oriented Design) 為基礎來建立 Web-based 應用系統品質度量模型(Bansiya 02)，以評量物件導向設計方法所建置之模型品質，其以四個階層的品質屬性度量模型如圖 19 來建立測量的品質屬性與物件導向 Web-based 應用系統設計元件間的量化關係。因本研究所設計之 Web-based 應用系統物件模型已將 Web 平台的超媒體特性予以物件化，故可使用此技術建立 QMOOD 於 Web-based 應用系統之物件品質度量模型。此度量模型建立初期需先定義出欲度量的品質屬性，然而基本的 Web-based 應用系統物件模型品質屬性如表 1 所示，其屬性定義是依據 ISO 9126 定義並配合 Web-base 應用系統物件模型之系統特性所擬訂出的物件模型品質屬性此也為品質模型的第一階層，因本文中之方法論主要採用物件導向設計方法，而 ISO 9126 是廣泛所接受的物件導向模型品質指標，故以此標準所評量的指標將具有重要的參考性；第二層的設計特性主要由審查模型的內部與外部結構、關聯性(超連結特性)、Web 元件的功能性、Web 類別的屬性與方法、以及 Web 類別定義，此設計特性在 QMOOD 中彙整如表 2 中的第一欄；而第三層的設計測量即為根據前一層特性定義在實際的物件模型中可量化的數值如表 2 的第二、三欄所示(Bansiya 02)；最後一層為實際設計出的 Web 應用系統中的元件，可以是實際的程式碼或其他物件模型的表示方式。在第三階層設計度量值也直接影響及反映出第一階層設計模型品質屬性，其影響的關係如表 3 所示(Bansiya 02)；此表中正的加權值表示對品質屬性有正面的影響，而負的加權值則表示對此品質屬性的影響為負面的；經由表 3 品質屬性與設計度量值的關係，單一品質屬性以加權比例反映出個別設計度量值的相關重要性，因此可使所有品質屬性的計算值有相同的範圍，本論文中對於品質屬性的計算值採用 0 到+1 或 0 到-1 的範圍，正值表示正面影響品質屬性的程度、負值表示負面影響品質屬性的程度而 0 則表示對品質屬性沒有影響。依據前述原則設定模型參數時，初期過程對於正面影響的加權值使用+1 或+0.5，且負面影響使用-1 或-0.5。經由初步的權重設定之後，再逐項檢視單一品質屬性並且調整影響比例的權重值使新的權重值總合為±1，完成之後的品質計算公式如表 4。使用這樣的加權影響設定主要是因可是簡單與直接的應用在經由品質屬性與設計屬性的關係定義。故利用此度量模型可將 Web-based 應用系統物件模型予以量化，再經由度量公式可以計算此應用系統物件模型的品質度量更進而驗證設計後期的模型有效性。

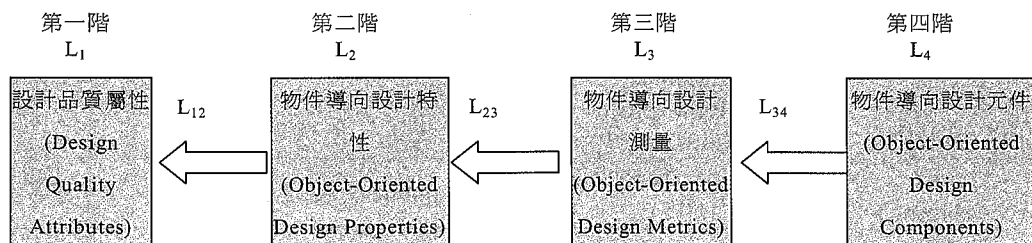


圖 19：QMOOD 的品質屬性階層模型

表 1：評估之品質屬性

品質屬性 (QUALITY ATTRIBUTE)	屬性定義 (DEFINITIONS)
重複使用性(Reusability)	反應網頁應用系統設計模型可重複應用於新的應用系統上
彈性 (Flexibility)	網頁應用系統設計模型允許變動設計模型的能力(功能需求相同的情形下)
可理解性 (Understandability)	網頁應用系統模型能夠易於學習與理解
功能性 (Functionality)	網頁應用系統模型中類別所被負予的責任
延展性 (Extendibility)	網頁應用系統模型可以接受新需求的能力
有效性 (Effectiveness)	網頁應用系統模型能夠達成功能與行為的能力

表 2：評估的度量值

設計特性	度量值 (METRIC)	說明 (DESCRIPTION)
設計尺度 (Design Size)	Design Size in Classes (DSC)	於設計中所有類別的合計
階層性 (Hierarchies)	Number of Hierarchies (NOH)	在設計中類別階層的合計
抽象性 (Abstraction)	Average Number of Ancestors (ANA)	在繼承結構中根類別到所有類別隨著路徑的 類別數量平均值
封裝性 (Encapsulation)	Data Access Metric (DAM)	在類別中屬性宣告為 private 或 protected 所 佔的比例 (範圍由 0 至 1)
耦合性 (Coupling)	Direct Class Coupling (DCC)	類別直接關聯到另一個類別差異數的合計。 包含類別中屬性宣告與方法參數
內聚性 (Cohesion)	Cohesion Among Method of Class (CAM)	基於方法的參數列表計算類別方法上的相關 性。在類別中所有參數型態的最大獨立集合 中方法參數交集的總合。接近 1.0 最好
複合性 (Composition)	Measure of Aggregation (MOA)	資料宣告為型態為使用者定義類別的總數
繼承性 (Inheritance)	Measure of Functional Abstraction (MFA)	類別繼承方法佔所有方法的比例
多形性 (Polymorphism)	Number of Polymorphic Methods (NOP)	具有明確多形行為方法的合計 (如 C++ 中宣告為 virtual)
訊息性 (Messaging)	Class Interface Size (CIS)	宣告為 public 方法數量的合計
複雜度 (Complexity)	Number of Methods (NOM)	所有方法的合計

表 3：品質屬性與模型度量值的關聯性

	重複 使用性	彈性	可理解性	功能性	延展性	有效性
設計規模	+		-	+		
階層性			-	+		
抽象性					+	+
封裝性		+	+			+
耦合性	-	-	-		-	
內聚性	+		+	+		
複合性		+				+
繼承性					+	+
多形性		+	-	+	+	+
訊息性	+			+		
複雜度			-			

表 4：評估的度量計算公式

品質屬性 (QUALITY ATTRIBUTE)	索引值計算公式 (INDEX COMPUTATION EQUATION)
重複使用性 (Reusability)	$-0.25*DCC + 0.25*CAM + 0.5*CIS + 0.5*DSC$
彈性 (Flexibility)	$0.25*DAM - 0.25*DCC + 0.5*MOA + 0.5*NOP$
可理解性 (Understandability)	$-0.33*ANA + 0.33*DAM - 0.33*DCC + 0.33*CAM - 0.33*NOP - 0.33*NOM - 0.33*DSC$
功能性 (Functionality)	$0.12*CAM + 0.22*NOP + 0.22*CIS + 0.22*DSC + 0.22*NOH$
延展性 (Extendibility)	$0.5*ANA - 0.5*DCC + 0.5*MFA + 0.5*NOP$
有效性 (Effectiveness)	$0.2*ANA + 0.2*DAM + 0.2*MOA + 0.2*MFA + 0.2*NOP$

由前述可知 QMOOD 的物件品質評量模型是依據實際設計的 Web 元件逐層推演至最高層的物件品質屬性，這樣的評量方法因為不涉及人員技術熟練度或人為管理上的因素，所以非常適合使用此技術實際建置自動化度量工具以提供快速的模型品質評量試算，此可以提供設計人員進行 Web-based 應用系統設計、建置以及循環式的系統開發時，能快速有效的模型品質度量值供其參考或審視模型有效性。使用 QMOOD 技術的自動化度量工具其運作之演算法如表 5 所示，此演算法輸入為設計模型的元件資料而經由計算之後輸出此模型的品質屬性，而處理的過程簡單的分為三個步驟(1)讀取設計模型並建構內部所使用的模型結構；(2)逐一針對品質度量值的項目讀取內部模型

結構後累計其數值；(3)依據品質屬性計算公式與品質度量值核算出此模型的品質屬性，另本研究中也實際建置此度量工具以供 Web-based 應用系統品質評估之用。

表 5：自動化評量工具演算法

```

procedure Auto_Quality_Attributes(in components : { component },
    out quality : array[Reusability .. Effectiveness] of attributevalue)
begin
    /* 利用系統設計元件建構設計模型 */
    /* 設計模型為模型元件所構成之集合，元件包含有類別、物件屬性、物件方法與物件關聯性 */
    designmodels : { object model element}
    /* 經由讀取元件相關資訊建構設計模型 */
    designmodels := InitializeObjectModel(components)

    /* 由設計模型中累計出品質度量值 */
    /* 品質度量值為儲存度量結果之資料結構 */
    qualitymetrics : array[DSC .. NOM] of metricvalue
    /* 讀取設計模型中的每個元件累計至品質度量值中 */
    for each element in designmodels do
    begin
        /* 針對單一設計模型元件異動品質度量值 */
        QualityMetricsAccumulator(qualitymetrics, element)
    end

    /* 由品質度量值核算出品質屬性 */
    /* 品質屬性值為儲存屬性結果之資料結構 */
    qualityattributes : array[Reusability .. Effectiveness] of attributevalue
    /* 品質屬性核算公式含有相關品質度量值與核算度量值之係數 */
    qualityformulaset : { qualityformula }
    for each attribute from Reusability to Effectiveness do
    begin
        /* 依照公式定義核算出屬性值 */
        qualityattributes[attribute] :=
            ComputeFormula(qualityformulaset(attribute), qualitymetrics)
    end
    quality := qualityattributes
end
end

```

二、Web-based 應用系統評估結果分析

We-based 應用系統物件模型建置完成時可使用度量模型將物件模型予以量化評估，並可藉由本研究所自行開發的自動化的度量工具來快速地計算出單一模型的品質度量值與累計比較模型版本演化時的差異比較值。在自動化度量工具上使用 Rational 公司 XDE 設計工具之物件模型檔來進行品質度量計算的來源資料，因為其結合 Microsoft Visual Studio.Net 開發工具中，可提供系統開發人員於系統設計階段或於系統建置階段均可使用其系統進行物件模型建置之工作，而本研究將使用其物件模型進行模型品質的度量評估。本研究以前章所述之 Web-based 應用系統模型為實例進行績效評估，以相同的系統需求功能建置應用系統，並使用物件導向設計方法的循環式開發程序，而系統的開發人員為相同的一組開發小組，第一次設計是由原設計人員利用其各自既有的開發經驗加以討論並系統設計與實作(此為 V 1.0 版)，第二個設計模型是依循本研究所提出之物件導向 Web-based 應用系統開發方法中的設計原則加以改善

其設計模型，進而演化成為（此為 V 2.0 版）再加以實作，以驗證本論文之設計方法論的有效性。經過品質度量方法進行系統模型品質評量之後，個別系統模型之實際設計度量值與模型品質屬性指標如表 6 所示，此表中依據前述加權值設定的原則可知品質屬性與指標值大小成正比關係。另因為兩次設計之災害管理系統是在相同的機能需求，且由相同的開發人員進行建置，故具有可比較的基準。在此，本論文使用正規化值（第二版度量值除以第一版度量值）取代實際度量值與品質屬性表如表 7 及表 8 所示，以便於比較與分析計算結果。

表 6：災害管理資訊系統實際度量值

度量值 (METRICS)	災害管理系統 V 1.0	災害管理系統 V 2.0
設計規模 (DSC)	76	97
階層性 (NOH)	5	5
抽象性 (ANA)	1	1
封裝性 (DAM)	0.845	0.862
耦合性 (DCC)	749	774
內聚性 (CAM)	0.692	0.672
複合性 (MOA)	0	0
繼承性 (MFA)	0	0
多形性 (NOP)	613	684
訊息性 (CIS)	334	466
複雜度 (NOM)	729	889

表 7：正規化後之災害管理資訊系統設計模型度量值

度量值 (METRICS)	災害管理系統 V 1.0	災害管理系統 V 2.0
設計規模 (DSC)	1	1.28
階層性 (NOH)	1	1
抽象性 (ANA)	1	1
封裝性 (DAM)	1	1.02
耦合性 (DCC)	1	1.03
內聚性 (CAM)	1	0.97
複合性 (MOA)	1	1
繼承性 (MFA)	1	1
多形性 (NOP)	1	1.12
訊息性 (CIS)	1	1.4
複雜度 (NOM)	1	1.22

表 8：災害管理資訊系統品質屬性值

品質屬性 (QUALITY ATTRIBUTE)	災害管理系統 V 1.0	災害管理系統 V 2.0
重複使用性 (Reusability)	1	1.33
彈性 (Flexibility)	1	1.06
可理解性 (Understandability)	-0.99	-1.21
功能性 (Functionality)	1	1.17
延展性 (Extendibility)	1	1.05
有效性 (Effectiveness)	1	1.03

經由表 5 發現於 Web 應用系統設計時，因為使用 Microsoft .Net 技術之故，所有網頁均是繼承由單一物件，所以使用元件分割對於物件階層不會造成影響，而評量資料來源使用應用系統以物件模型設計工具進行反向工程後所產生之物件模型。因為物件模型設計工具中並不能判斷物件模型的語意，所以對於聚合性與繼承性就不能自動地計算出來。雖然如此，仍可透過正規化值（如表 7）計算兩系統之品質改善程度（如表 8），既有 Web 應用系統架構使用物件導向 Web 系統設計方法之後，經由實驗結果可知針對重複使用性與功能性兩項品質屬性，是具有較大的改善幅度，而對於彈性、延展性與有效性三項品質屬性亦提供相對地改善但其幅度較小，然而對於可理解性此項品質屬性卻會成反向的成長。主要是因為隨著物件技術上抽象化、封裝性以及關係管理結構法中元件的分割方法，會增加整個模型中類別關聯的程度，這樣的現象雖然對於其他的品質屬性有正面的提升，但是卻造成整個模型比之前的較難理解，所以應強化系統開發人員的技術教育訓練與豐富的開發文件內容將不難克服此項問題。然而本論文中以相同的程式設計小組為實驗對象，在進行第二次系統設計進行時人員可能已經累積前一次的經驗而使系統模型的品質獲得改善，雖然本論文所提之 Web-based 應用系統設計方法論是以具有系統化的程序進行第二次的系統模型設計，但未能實驗計算出僅設計方法論對於模型的影響，不過仍然對於 Web-based 應用系統設計方法論提供重要的參考價值。最後本論文之設計方法論在整合品質評量技術並實際度量 Web 應用系統設計模型後可以有效地達成下列三項重要成效：(1)利用關係管理結構法以結構化的方法建立網頁物件系統模型，可提供設計人員以具有系統化的設計程序有效地設計出網頁導覽模型，以表現出 Web-based 應用系統中較複雜的超媒體特性；(2)結合物件導向方法與 MVC 法可利於傳統應用系統移轉至 Intranet 架構下建置 Web-based 應用系統並且兼容傳統物件模型；(3)應用 Web 品質模型測量技術可於設計階段與雛型系統建置階段以量化的技術來評估 Web-based 應用系統系統開發的品質，以提供設計人員有效地品質評量之系統管理方式。

陸、結論

目前單純使用物件導向技術來設計 Web-based 應用系統之架構模型仍嫌有不足，主要是因為 Web-based 應用系統是以網頁與網頁間所組合或聯結而成的。因此，物件導向技術應用於 Web-based 應用系統時首要之務就是能將 Web 的超媒體 (Hypermedia) 特性亦即網頁與網頁間的超連結 (Hyperlink) 關係以物件的模型方式表現出來。而關係管理結構法則對超媒體關係模型化 (Hypermedia Process Model) 提供了結構化的關聯性架構，亦同時表示出網頁與網頁之間複雜的關聯性，但是卻難反應重複使用以及資料與程序之有效地分配等特性。然而目前的物件導向技術則對重複使用於資料和程序之有效分配等應用程式特性提供了有效的解決方法，所以除利用關係管理結構法來設計 Web-based 應用系統的導覽模型外，並應整合 MVC 方法將模型與行為物件加以分割後，便於使用 Microsoft .Net 中 ASP.NET 與 ADO.NET 技術來建置複雜的企業邏輯規則以及雛型系統以提昇系統效能。再利用 QMOOD 品質測量方法於系統設計與系統建置階段驗證其特性。總之，在整合 Web 網頁間關聯架構與物件導向技術來設計 Web 應用系統，預期將達成模組性、再利用性、可維護性、資料與程序之有效分配、安全性以及良好的績效等特性，因而縮短 Web-based 應用系統的開發週期和降低其開發成本進而大幅提高了系統之維護性以及品質。

利用整合式物件導向技術來設計 Web-based 應用系統之物件(網頁)導覽模型，除可快速且有效地將傳統應用系統移轉至 Web-based 應用系統之外，未來可將 XML 技術以統合企業內資訊格式與資訊交換內容並利用建置 XML 應用程式架構而達成更大的延展性與重複使用性並與其他 Web 程式設計技術有更完善的整合(Carlson 01)。此外，所設計出 Web 應用系統物件模型可利用 MVC 法進行功能分配，利用此種方法可以再整合 XML 中的 XSLT (Extensible Stylesheet Language Transactions) 技術可使 Web 應用系統適合於行動通訊的計算環境與一般 Intranet 的計算環境。QMOOD 品質度量方法上可以再納入多媒體特性的品質參考加權值以及多媒體元件度量期能適合於既有 Web 應用系統品質度量，如此可提供未來 Web-based 應用系統開發建置之重要依據。此外因 QMOOD 方法中的度量值可直接由系統的模型或原始碼中取得，故此方法可開發自動化評量工具應用系統，將可擴展使其他相關模型工具或以原始碼來進行的品質度量程序自動化。原則上，利用本研究提出的方法論所產生之物件模型架構也可適用於更多相關的 Web 物件技術例如 J2EE 技術，並能有效地計算其品質屬性。

誌謝

本論文係國科會研究計畫(計畫編號: NSC 89-2218-E-031-004)之部分內容，研究期間感謝國科會經費補助得以順利完成；另本論文承兩位匿名審查委員惠賜寶貴意見並悉心指正，特此致謝。

參考文獻

1. Bansiya, J. and Davis, C.G. "A Hierarchical Model for Object-Oriented Design Quality Assessment," *IEEE Transactions on Software Engineering* (28:1) 2002, pp: 4-17.
2. Bray, T., Paoli, J. and Sperberg-McQueen, C. M. "Extensible Markup Language (XML) 1.0," W3C Recommendation, *World Wide Web Consortium* 1998, <http://www.w3.org/TR/REC-xml>.
3. Buschman, F., Meunier R., Rohnert, H., Sommerland, P. and Stal, M. "Pattern-Oriented Software Architecture, A System of Patterns," John Wiley & Sons Inc. U.S.A., 1996.
4. Cho, E.S., Kim, S.D., Rhew, S.Y., Lee, S.D. and Kim, C.K. "Object-Oriented Web Application Architectures and Development Strategies," In 4th *Asia-Pacific Software Engineering Conference and International Computer Science Conference (APSEC '97 / ICSC '97)*, Clear Water Bay, HONG KONG, Dec. 2, 1997.
5. Clark, J. and Deach, S. "Extensible Stylesheet Language(XSL)," W3C Working Draft, *World Wide Web Consortium*, 1998, <http://www.w3.org/TR/WD-xsl>.
6. Conallen, J. "Building Web Application with UML," Addison-Wesley Inc. Reading MA, U.S.A., 2000.
7. Carlson, D. "Modeling XML Application with UML," Addison-Wesley Inc., Reading MA, U.S.A., 2001.
8. Decker, S., Melnik, S., Harmelen, F. V., Fensel, D., Klein, M., Broekstra, J., Erdmann, M. and Horrocks, I. "The Semantic Web: The Roles of XML and RDF," *IEEE Internet Computing* (4:5) 2000, pp: 63-74.
9. Dromey, G.R. "Cornering the Chimera," *IEEE Software* (13:1) 1996, pp: 33-43.
10. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley Inc., Reading MA, U.S.A., 1995.
11. Garzotto, F., Schwabe, D. and Paolini, P. "HDM- A Model Based Approach to Hypermedia Application Design," *ACM Transactions on Information Systems* (11:1) 1993, pp: 1-26.
12. Gellersen, Hans-W. and Gaedke, M. "Object-Oriented Web Application Development," *IEEE Internet Computing* (3:1) 1999, pp: 60-68.
13. Goeschka, K. M. and Schranz, M. W. "Client and Legacy Integration in Object-Oriented Web Engineering," *IEEE MultiMedia* (8:1) 2001, pp: 32-41.
14. Isakowitz, T., Stohr, E. and Balasubramaniam, P. "RMM: A Methodology for Structured Hypermedia Design," *Communications of the ACM Journal* (38:8) 1995, pp: 34-48.
15. Isakowitz, T., Kamis, A. and Koufaris, M. "Reconciling Top-Down and Bottom-Up Design Approaches in RMM," *The DATA BASE for Advances in Information Systems*

- (29:4) 1998, pp: 58-67.
16. Jacobson, I. "Object-Oriented Software Engineering: A Use Case Driven Approach," Addison-Wesley Inc. Reading MA, U.S.A., 1992.
 17. Kung, D., Liu, C.H. and Hsia, P. "An Object-Oriented Web Test Model for Testing Web Applications," In *Proc. of IEEE 24th Annual International Computer Software and Applications Conference (COMPSAC2000)*, Taipei, Taiwan, Oct. 25, 2000.
 18. Larman, C. "Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design," Prentice Hall Inc., NJ U.S.A., 1998.
 19. Layman, A. "XML-Data," W3C Note, *World Wide Consortium*, 1998, <http://www.w3.org/TR/1998/NOTE-XML-data>.
 20. Gaedke, M., Gellersen, H.W., Schmidt, A., Stegemüller, U. and Kurr, W. "Object-oriented Web Engineering for Large-scale Web Service Management," In *32nd Annual Hawaii International Conference on System Sciences (HICSS '32)*, Maui, Hawaii, USA, Jan. 5. 1999.
 21. Medes, E., Mosley, N. and Counsell, S. "Web Metrics-Estimating Design and Authoring Effort," *IEEE MultiMedia* (8:1) 2001, pp: 50-57.
 22. Meyer, B. ".NET Is Coming," *IEEE Computer* (34:8) 2001, pp: 92-97.
 23. Reifer, D. J. "Web Development: Estimating Quick-to-Market Software," *IEEE Software* (17:6) 2000, pp: 57-64.
 24. Rossi, G., Schwabe, D. and Garrido, A. "Pattern Systems for Hypermedia", In *Proc. of PLOP'97* - University of Illinois, Monticello, USA, 1997.