

整合式物件導向技術於 Web 應用系統設計之研究

Integrated Object-Oriented Approach to Modeling the Web Application's Design

楊欣哲

東吳大學資訊科學系副教授
e-mail : sjyang@cis.scu.edu.tw

高國峰

東吳大學資訊科學系碩士生
e-mail : nms8905@cis.scu.edu.tw

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 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 應用系統整合資料與行為之基礎。由於 Web 應用系統具備:(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特性,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 應用系統的開發與擴充性是相當重要的。有鑑於此,本研究將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,使用 UML(Unified Modeling Language)工具加以延伸擴充以整合 RMM 方法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 應用系統,而達成有效地建立 Web 應用系統之目標。利用此種系統設計技術其後經用 XML 語言與 Java 語言來建置一 Web 應用系統,可有效地建置物件系統模型並驗證此系統設計方法的可行性與實用性。總之,採用此整合式物件導向技術於 WWW 架構下來建置 Web 應用系統預期將可帶來六大特色: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。

關鍵詞: 物件導向、關係管理結構法、快速雛型法、Web、MVC

Abstract

Due to the recent advances in World Wide Web technology, how an organization to efficiently implement Web applications under WWW environment is becoming a primary issue. 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 Applications. Since all information on the WWW are presented through Web pages, and Web pages are inter-connected, it is essential to clearly identify the connectivity of the Web pages in order to develop Web applications efficiently. On this basis, we can integrate OOA/OOD (Object-Oriented Analysis and Design) with RMM (Relationship Management Methodology) analysis via utilizing UML (Unified Modeling Language) and 3-tier MVC (Model-View-Controller) and Prototyping to design the system for Web applications. Based on that design issues, we can build a practical Web application through the use of XML and JAVA to identify the feasibility and usability. Consequently, any Web applications built upon this integrated object-oriented approach can have six anticipated merits: modularity, reusability, maintainability, effective distribution of data and functionality, security, and good performance.

Keywords: Object-Oriented, RMM, Prototyping, Web, MVC

一、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。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 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 應用系統也稱為超媒體應用系統(hypermedia application) [10]。現今網頁開發技術的標準已由 HTML 演化成為 XML(Extensible Markup Language)[1],這兩種技術都是由 W3C(World Wide Web Consortium)所制

定。XML 與 HTML 主要的差異在於 XML 沒有固定的標籤集(tag set)[23]，可允許使用者透過 XML-Data[19] 可自行的定義 XML 文件結構，再透過 XSL(Extensible Stylesheet Language)[4] 可使 XML 能更有彈性地展現文件的內容。以上的技術使得網頁能以更生動活潑的方式呈現出超媒體的特質，但是要設計 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 應用系統的超媒體特性，這也是傳統應用系統開發方法無法適用於 Web 應用系統的原因[11]。針對 Web 應用系統的超媒體特性，在系統設計上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) [14]，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) [20]，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 [11] 模型與使用 UML 工具來建構 Web 應用系統的 Jim Conallen [4, 5] 設計模型。

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 應用系統開發時因具備有三種的觀點來說明所建置的系統模型 [22] 分別是：(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) [18] 工具已是軟體業界公認的物件模型圖示標準，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，所以 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[8]。因此本篇論文將提出一套整合系統設計方法以開發 Web 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) 以有效地設計出 Web 應用系統導覽模型與使用 UML 工具，並加以延伸擴充。總之，RMM 分析出的導

覽模型能使用 UML 工具加以表示，並且於設計時配合三層次 MVC (Model-View-Controller)與快速離型法(Prototyping)等方法，以達到有效地建立 Web 應用系統的導覽模型和網頁架構。XML 技術可以由使用者自行定義或擴充，透過 XML 中 DOM、DTD(Document Type Definition)、XLink 與 XSL(Extensible Stylesheet Language)等相關技術可以預先建置一 XML 應用程式架構(Framework)。因而使 Web 應用系統建置時能更方便地將物件系統模型轉化成 XML 文件，並將複雜的企業邏輯以 Java 語言工具建置以結合 XML 文件，由此可整合現有或傳統的應用程式系統[12] 進而產生其離型系統。最後，將使用此方法以設計一 Web 應用系統實例，來說明並驗證此設計方法的可行性、有效性與實用性。

二、Web 應用系統設計原理

本節將概要地介紹 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，經由物件導向方法可以設計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應用程式中的物件模型上加入資料庫物件(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之後，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。

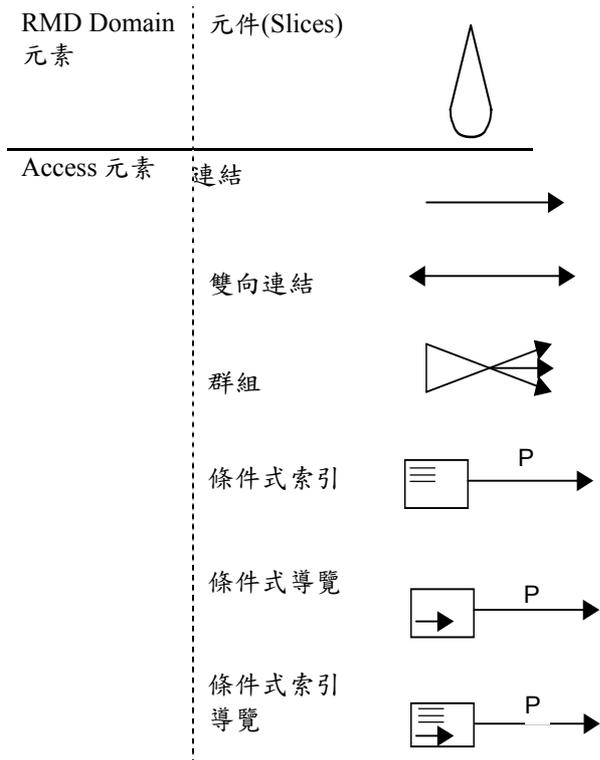


圖 1. RMM 方法論物件導覽模型基本圖示

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 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 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 方法使設計人員可以有一清晰且明確的方法已建立 Web 應用系統中的導覽模型。

(二)、Web 應用系統使用物件式導向設計方法

Web 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

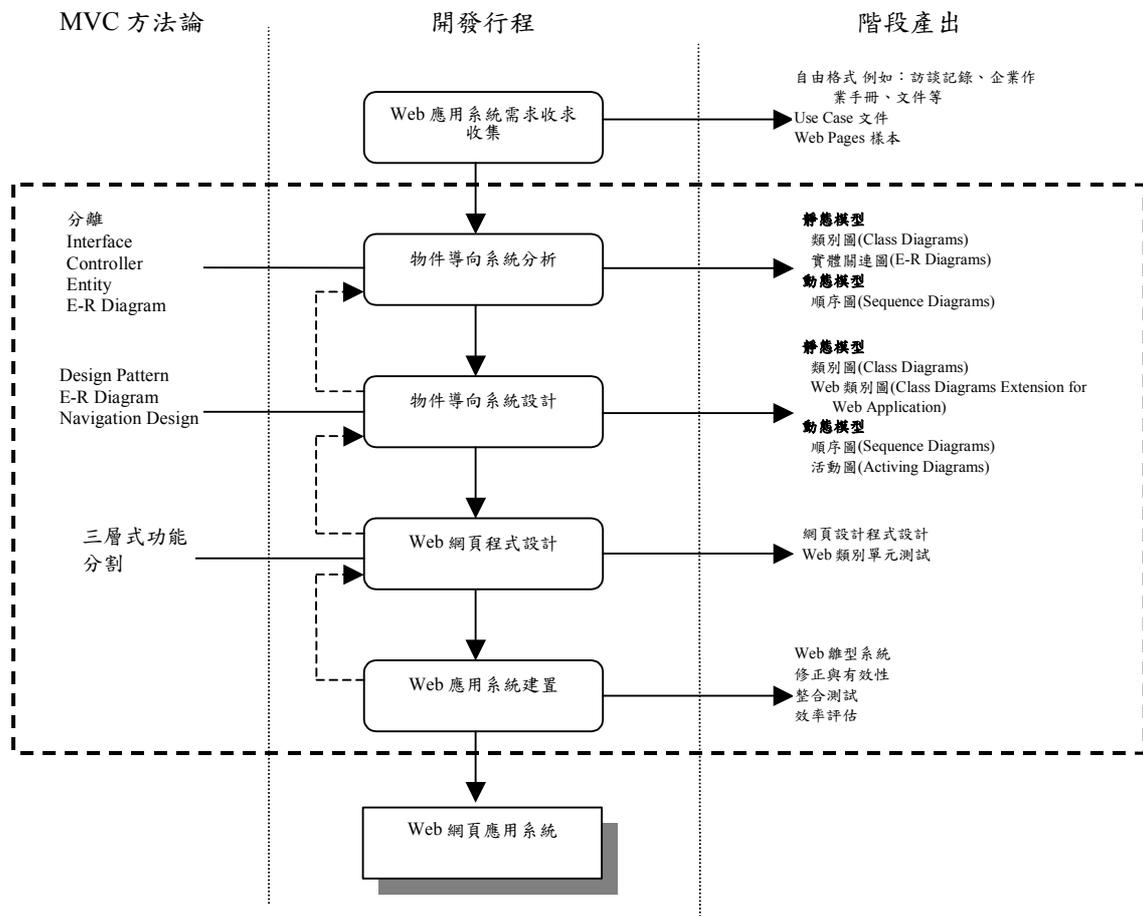


圖 2. 物件導向 Web 應用系統開發流程

統(不論是不是 Web 應用系統)皆是以概念模型為其之核心，它反映了應用系統中物件的關係與行為。因為 Web 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，它主要在定義一個概念模型的 View，如同 MVC 模型中對於 View 定義，這也是 Web 應用系統系統設計最重要的部分。最後是介面模型，主要在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的，然而在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 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 應用系統開發時可利用現有物件導向方法之優點。但是 Web 應用系統與傳統應用系統設計主要的差異是導覽模型的設計，對於 Web 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。下節將先介紹此整合式物件導向方法並探討離型系統與說明模型改善方式，最後將使用此方法建置一 Web 應用系統實例以進行實証研究。

三、整合式物件導向 Web 應用系統方法

綜觀前節對於 Web 應用系統的設計原理，本文所提出之整合式物件導向 Web 應用系統方法是將超媒體之特性結合物件導向開發技術。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 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 應用系統開發程序可分為五個步驟其流程如圖二所示，現將各步驟說明如下：

步驟一：Web 應用系統功能需求收集。此階段為 Web 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來作為需求收集與建置 Use Case [15] 模型的基礎，由此階段建構 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。此階段中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 (Use Case) 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 (Entity) 物件、控制 (Control) 物件與介面 (Interface) 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。

步驟三：物件導向系統設計。此階段主要是由系統分析階段的結果來進行系統的設計工作，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。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設計，此導覽模型將由 Web 應用延伸類別圖(Class diagrams extension for Web application)的方式表現。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，並使用 UML 工具以有效地完成系統物件設計模型。

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 應用系統的離型系統。首先，測試各整合後之單元的功能性是否完備，並驗證 Web 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)和漸進方式來增強離型系統，並可使用物件導向測試方法來測試此 Web 離型系統各子單元[17]。最後，各個子單元系統結合成 Web 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

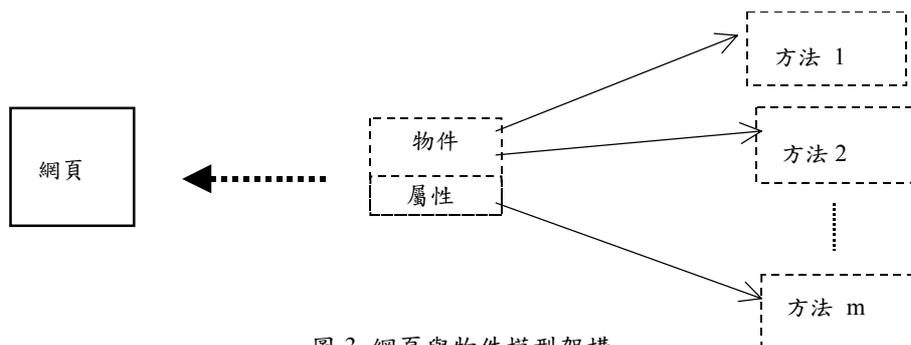


圖 3. 網頁與物件模型架構

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 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(Hypermedia)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 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 [18]。現針對 Web 應用延伸類別圖中的圖示逐一說明：

元件(Slices)類別：元件類別(圖 4 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 4. 元件(Slices)類別

節點(Node)類別：節點類別(圖 5 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 應用系統之中。

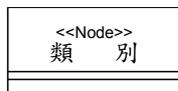


圖 5. 節點(Node)類別

類別間存取 (Access) 關係：主要是說明 Web 應用系統的導覽模型，而類別間存取關係所使用的圖示可分為三類：(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 6 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。

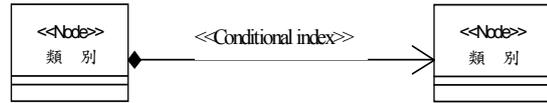


圖 6. 條件式索引(Conditional index)關係

條件式導覽(Conditional Guided Tour) 如圖 7 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 7. 條件式導覽(Conditional Guided Tour) 關係

條件式索引導覽(Conditional Indexed Guided Tour) 如圖 8 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 8. 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 9 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 9. 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 10 所示：表示元件類別間有參考引用的關係存在，但是參考方向是固定的時候。

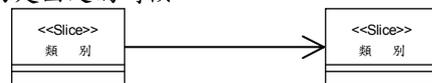


圖 10. 單向連結(Unidirectional link)關係

雙向連結(Bi-directional link) 如圖 11 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。

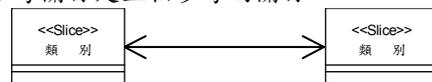


圖 11. 雙向連結(Bi-directional link)關係

透過上述的圖示將 Web 應用系統中導覽模型以 UML 類別圖的型式表現出來，既有 UML 工具的類別圖可以整合 Web 應用系統模型的特殊屬性加以擴充，例如圖 12 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。

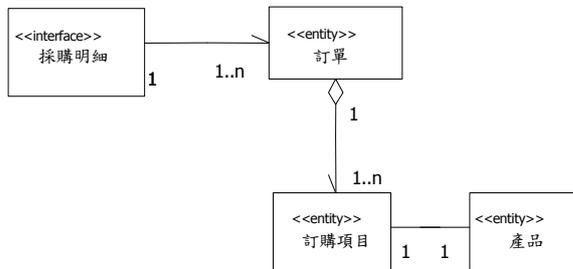


圖 12. 訂購系統物件模型

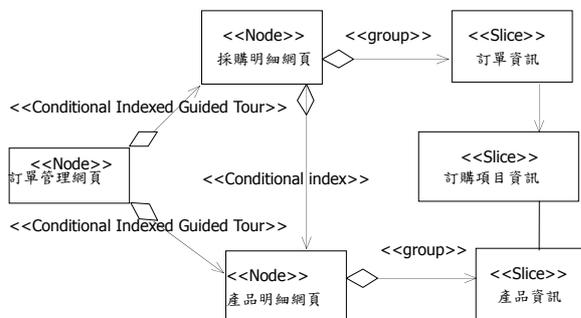


圖 13. 訂購系統物件式導覽模型

本研究之設計方法主要的目標是針對 Web 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[2, 7] 的模型，行為物件則可利用 JAVA 的 Servlet / JSP 或 CORBA[3, 13] 的技術來加以建置。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 格式，而網頁物件則為此 DTD 所規範之 XML 格式。再將網頁物件間關聯性繼承預先所建置之 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 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 Web 應用系統之效率和建置更複雜的 Web 應用系統。

四、Web 應用系統實例探討

本節中主要是依據上述的整合式物件導向方法於 WWW 的環境上設計電力公司停限電管理系統，並實際探討此 Web 應用系統之相關議題以驗證此設計方法之可行性與有效性。因電力公司停限電系統常需要提供一般電力用戶停限電資訊，所以適合使用 Web 系統平台來建立。利用物件導向技術設計 Web 應用系統可提供模組性、再利用性、擴充性等優點。因此，使用物件導向技術概念對於停限電資訊系統的開發有極大的幫助。另外，實際系統的建置部分則可使用 Java 技術，因為 Java 具有跨平台、軟體功能分配性、可攜性與易於維護等優點。

現針對電力公司停限電管理系統建置依上述所建議的 Web 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，以完成停限電管理系統的使用個案圖如圖 14 所示。其主要系統功能分別為工作停電子系統、事故停電子系統、限電管理子系統、停限電月報子系統、電力用戶停限電資訊子系統等五項子系統，其各子系統功能間相依的關係如圖 15 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。依據傳統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。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)，所產生延伸類別圖再應用 MVC 功能分配法來確認各網頁是否已完成適當的資料與程序之分配，如此即可完成系統的物件設計模型。本實例中將電力用戶停限電資訊子系統透過物件類別結構間的實體物件關連模型設計之後完成 Web 應用延伸類別圖如例圖 16 所示。完成設計之後，則進行 Web 網頁程式設計步驟，此步驟中先使用 XML 技術建置網頁物件再利用 Java 技術建置電力公司之企業行為物件並在 WWW 的環境上建置用戶停限電資訊子系統之離型系統。以建置離型系統為基礎，依使用者實際功能需求以重複(iterative)和漸增方式來修正與增加停限電管理離型系統功能。離型系統完成後，將依系統功能需求與物件導向測試方法對離型系統進行功能測試，以確保功能完整性並評估系統執行效率。最

後，可於 WWW 作業環境上建置較完整的電力公司停限電管理系統。

經使用本研究之設計方法實例驗證後，證實本設計方法有兩項重要成效：(1)利用關係管理結構法可以有效地將 Web 應用系統中複雜的超媒體特性以系統結構式設計出網頁導覽模型，降低因超媒體特性所造成的系統複雜度；(2)結合物件導向方法與 MVC 法可使傳統應用系統能快速地移轉至 Intranet 架構下建置 Web 應用系統並且兼容傳統物件模型，以有效地達成模組性、再利用性與可維護性及良好的績效。

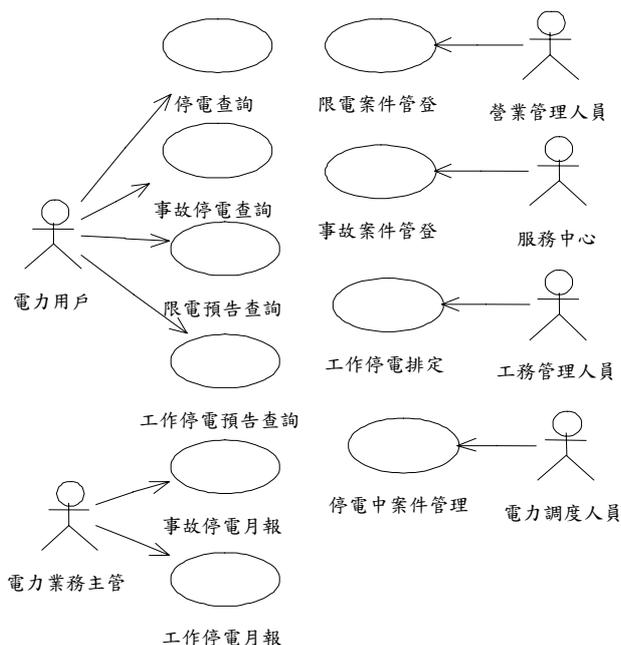


圖 14. 電力運轉管理系統 Use Case 功能結構圖

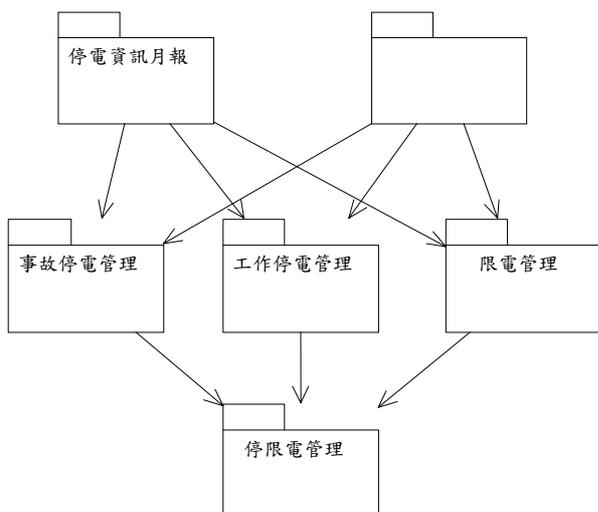


圖 15. 電力運轉管理各子系統功能相依圖

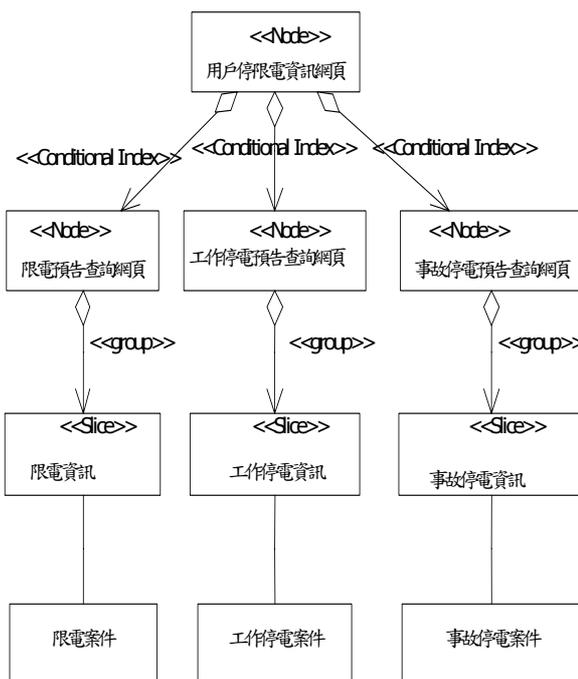


圖 16. 電力用戶停限電資訊子系統 導覽類別圖

五、結論

目前單純使用物件導向技術來設計 Web 應用系統之架構模型仍嫌有不足，主要是因為 Web 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 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 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 JAVA 或 CORBA 的技術來設計複雜的企業邏輯規則以建置離型系統以提昇系統效能。預先將網頁物件間關聯性利用 XML 技術建立 XML 應用程式架構後，對於本研究所提出之物件導覽模型將可利用物件導向繼承技術將網頁導覽的關係經由繼承此應用程式架後能有效地建立。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 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。

利用整合式物件導向技術來設計 Web 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 應用系統之外，未來可將 XML 技術[7] 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。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。原則上，利用本研究提出的設計方法所產生之物件模型架構應可適用於更多相關的 Web 物件技術例如 Microsoft's .Net 或 J2EE 技術。

誌謝：本論文承蒙國科會研究計畫經費補助得以順利完成(計畫編號: NSC 89-2218-E-031-004)，特申謝意。

六、參考文獻

- [1] T. Bray, J. Paoli, and C.M. Sperberg-McQueen, eds., "Extensible Markup Language (XML) 1.0," W3C Recommendation, World Wide Web Consortium, 1998; <http://www.w3.org/TR/REC-xml>.
- [2] F. Buschman, R. Meunier, H. Rohnert, P. Sommerland and M. Stal, "Pattern-Oriented Software Architecture, A System of Patterns," John Wiley & Sons Inc. U.S.A., 1996.
- [3] Eun Sook Cho, Soo Dong Kim, Sung Yul Rhew, Sang Deuk Lee and Chang Kap Kim, "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—Dec. 5, 1997.
- [4] J. Clark and S. Deach, eds., "Extensible Stylesheet Language(XSL)," W3C Working Draft, World Wide Web Consortium, 1998, <http://www.w3.org/TR/WD-xsl>.
- [5] Jim Conallen, "Building Web Application with UML," Addison-Wesley Inc. Reading MA, U.S.A., 2000.
- [6] Jim Conallen, "Modeling Web Application with UML," White Paper Conallen Inc, June 1998, <http://www.conallen.com/whitepapers/webapps/ModelingWebApplications.htm>.
- [7] David Carlson, "Modeling XML Application with UML," Addison-Wesley Inc., Reading MA, U.S.A., 2001.
- [8] S. Decker, S. Melnik, F. V. Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann and I. Horrocks, "The Semantic Web : The Roles of XML and RDF," *IEEE Internet Computing*, Vol. 4, No.5, Sep./Otc., 2000.
- [9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides., "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley Inc., Reading MA, U.S.A., 1995.
- [10] F. Garzotto, D. Schwabe and P. Paolini, "HDM-A Model Based Approach to Hypermedia Application Design," *ACM Transaction on Information Systems*, Vol. 11, No. 1, Jan. 1993, pp. 1-26.
- [11] Hans-W. Gellersen and Martin Gaedke, "Object-Oriented Web Application Development," *IEEE Internet Computing*, Vol. 3, No. 1, Jan./Feb. 1999, pp. 60-68.
- [12] Karl M. Goeschka and Markus W. Schranz, "Client and Legacy Integration in Object-Oriented Web Engineering," *IEEE MultiMedia*, Vol. 8, No. 1, 2001.
- [13] T. Isakowitz and E. Stohr; P. Balasubramaniam, "RMM: A Methodology for Structured Hypermedia Design," *Communications of the ACM Journal*, Vol. 38, No. 8, Aug. 1995, pp. 34-48.
- [14] T. Isakowitz, A. Kamis, and M. Koufaris., "Reconciling Top-Down and Bottom-Up Design Approaches in RMM," *The DATA BASE for Advances in Information Systems*, Vol. 29, No. 4, pp. 58-67.
- [15] Ivar Jacobson, "Object-Oriented Software Engineering: A Use Case Driven Approach," Addison-Wesley Inc. Reading MA, U.S.A., 1992.
- [16] R. E. Johnson and B. Foote, "Design Reusable Class," *Journal of Object-Oriented Programming*, Vol. 1, No. 5, June/July 1988, pp. 22-35.
- [17] David Kung, Chien-hung Liu and Pei Hsia, "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, October 25 -- October 29, 2000.
- [18] C. Larman, "Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design," Prentice Hall Inc., NJ U.S.A., 1998
- [19] A. Layman et al., "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. -- Jan. 8. 1999.
- [21] E. F. Mohamed and D. C. Schmidt, “Object-Oriented Application Frameworks,” *Communications of the ACM Journal*, Vol. 40, No. 10, October 1997, pp. 57-66.
- [22] Daniel Schwabe and Gustavo Rossi, “An Object Oriented Approach to Web-Based Application Design,” *Theory and Practice of Object Systems*, Vol. 4, No. 4, 1998, pp. 207-225.
- [23] Lauren Wood, “Programming the Web : The W3C DOM Specification”, *IEEE Internet Computing*, Vol. 3, No. 1, Jan./Feb. 1999, pp.48-54.