

# 數位典藏異動管理模組剖析與建議

林彥君

林泰安

王祥安

國立台灣科技大學資訊工程系 106 台北市基隆路四段四十三號

yclin@et.ntust.edu.tw

M9315029@mail.ntust.edu.tw

D9215004@mail.ntust.edu.tw

## 摘要

因應未來大量數位典藏系統建置的需求，我們剖析中央研究院資訊科學研究所數位典藏技術發展組所發展的異動管理模組，以減少新數位典藏單位的相同模組分析設計與開發所花費的時間和人力。剖析結果以 Unified Modeling Language 與 model-view-controller 架構來描述異動管理模組，讓讀者容易了解其功能與架構。最後也提出一些問題與建議，以供未來數位典藏系統開發人員參考。

## 1. 簡介

數位典藏系統以數位化的方式典藏珍貴文物、人文記錄等資料。在建構典藏資料的過程中，經常會對典藏資料進行新增、刪除或修改。若是缺乏詳細的異動記錄(log)作為檢驗工具，當系統發生錯誤時，將無法有效除錯和回復系統。因此需要異動管理模組負責儲存所有的異動記錄，以便追蹤典藏資料增刪與修改的過程。

本文根據中央研究院資訊科學研究所數位典藏技術發展組(DAAL)發展的異動管理模組原始程式和數位典藏系統核心元件及模組應用程式之介面說明[5]，找出可重複使用的元件或架構，提供給新數位典藏系統開發人員參考，以減少重複的分析與開發，並加速其建置。異動管理模組的剖析結果以 Unified Modeling Language (UML) [1,2,4]描述功能、架構、元件與運作流程，並用 model-view-controller (MVC) [3]架構將目前模組的外部顯示和內部實作分開以提高模組化。

本文架構如下：第 2 節介紹用來描述剖析結果的 UML 及 MVC 架構。第 3 節描述異動管理模組的功能與相關軟體架構。第 4 節詳細說明異動管理模組剖析

的結果。第 5 節提出在剖析過程中發現的問題與建議。最後一節為結論。

## 2. 表示剖析結果的相關技術簡介

### 2.1. UML

UML 是一個軟體塑模(modeling)的標準語言[2,4]，目前已經廣泛地在企業上和學術上使用。UML 不是包含詳細程序的程式設計語言，而是一種描述設計結果的語言，適用於各類型的系統。本文使用到的 UML 圖有：部署圖(deployment diagram)、使用案例圖(use case diagram)、循序圖(sequence diagram)、類別圖(class diagram)。以下將簡介各圖形的表示法。

#### 2.1.1. 部署圖

部署圖主要的目的是表現軟、硬體系統的實際配置情形，可以清楚了解整個系統的架構。部署圖包含節點(node)、元件(component)、節點或元件之間的關係，其中節點代表的是一種實體元素，通常是具有記憶體和運算能力的硬體，元件指的是程式、檔案、文件等軟體物件。

#### 2.1.2. 使用案例圖

使用案例圖描述系統功能與其使用者之間的關係，它可以幫助系統開發人員找出系統需求或是了解現有的系統功能。使用案例圖包含一組使用案例(use case)、行為者(actor)與兩者之間的關係。一個使用案例說明系統所提供的一項功能，而行為者就是跟系統互動的人或外部系統。

#### 2.1.3. 類別圖

類別圖用來描述不同的個體(人、事物和資料)與個體之間的關係；它描述了一個系統的靜態結構。類別圖包含一組類別與其間的關係。當針對典藏網頁進行剖析時，由於 UML 並沒有完全支援 Web application

\* 本研究接受國科會編號 NSC-93-2422-H-011-001 研究計畫經費補助

的表示法，因此參考 UML Web Application Extension [1] 來說明類別之間的關係，並加上模板(stereotype)來描述其關係，如圖 2 中的<<include>>即為模板。

類別圖中的類別是由三個部分所組成，由上而下依序為類別名稱、屬性(attribute)、操作方法(method)，如圖 1 所示。圖中用“<”和“>”括起來的部份代表可省略的內容，用來敘述存取權限、說明、屬性類型、初始值、傳入參數與回傳類型等相關資訊。屬性和操作方法前面可加上“+”、“-”、“#”分別代表 public、private、protected 存取權限。當剖析 JavaServer Pages (JSP)網頁時，本文將 JSP 網頁視為一個類別，其中用到的變數或物件視為屬性，而所呼叫的函式(function)則當作操作方法。

本文所使用的類別圖關係有：相依(dependency)、結合(association)與一般化(generalization)關係。相依關係表示一個類別使用到其他類別的情形。我們使用到的相依關係包含 include 和 import，其表示方法以 include 為例，如圖 2 所示。在 JSP 中，include 代表將指定的類別程式碼加入成為程式的一部份；而 import 代表將指定的類別或套件(package)匯入(import)作為參考，因此可以使用所 import 的類別變數或函式。

結合關係說明兩個類別如何相互連接，在此兩個類別是同等地位的。我們使用到的結合關係有 link、submit、redirect，三者的功能相似，其表示方法以 link 為例，如圖 3 所示。link 和 submit 是一種 HTML 語法，link 代表連結到指定的網頁，submit 除了連結的功能外，還會伴隨著表單(form)資料的傳送。而 redirect 是一個 JSP 函式，只要執行就會重新導向指定的網頁。結合關係可以在連接線兩端註明 1(恰一)、0..1(零或一)、0..\*(零到多)、1..\*(一到多)或 m..n(m 到 n)來表現其對應的個數。一般化關係是父類別和子類別之間的關係，通常是用來表現繼承的關係。當一個類別繼承一個套件時，其表示方法如圖 4 所示。

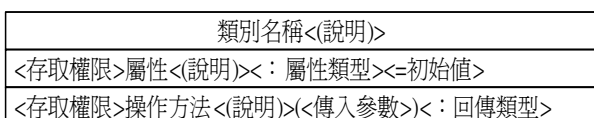


圖 1. 類別圖

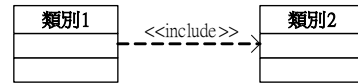


圖 2. 類別 1 include 類別 2 的表示法

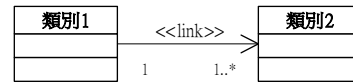


圖 3. 類別 1 link 類別 2 的表示法



圖 4. 一個類別繼承一個套件的表示法

#### 2.1.4. 循序圖

循序圖用來描述物件之間的互動情形，它強調時間上的先後順序。循序圖通常是系統的一個使用案例或是其部分的詳細流程，其中包含一組物件個體(instances)與其之間的訊息傳遞，物件個體可能是系統、類別或單位等物件。循序圖有兩個象限：垂直方向代表物件的生命線(lifeline)，描述事件發生的順序；水平方向則描述物件個體間傳遞訊息的過程。

### 2.2. MVC 架構

MVC 是一種設計模式(design pattern) [3]，它將軟體系統分成 model、view、controller 三個部分，以分開外部顯示跟內部運作的設計工作，並提高模組化和重複使用性。其中 model 包含系統的核心功能、邏輯與狀態。view 是整個系統的外觀呈現，當 model 的狀態改變時應該要通知 view 變更外觀，以維持跟 model 狀態的一致性。controller 反應使用者的輸入，呼叫 model 中相對應的功能，處理結果可能會改變 model 的狀態，且可以選擇所要呈現的 view，之間的關係如圖 5 所示。

### 3. 異動管理模組簡介

在說明異動管理模組的剖析結果之前，本節先介紹其模組功能及相關軟體架構。

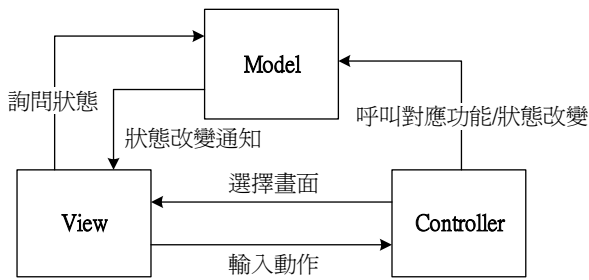


圖 5. MVC 架構圖

### 3.1. 模組功能

異動管理模組的功能分為兩個部分：寫入異動相關記錄和資料異動記錄檢索。當使用者透過數位典藏系統新增、刪除或修改典藏資料時，異動管理模組會將異動記錄(異動日期、使用者名稱等)寫入到資料庫中。若系統發生錯誤，異動記錄可以用來追蹤資料建構過程，進而除錯與回復系統，以確保資料的安全性。當需要查詢異動記錄時，異動管理模組以網頁的方式提供使用者檢索功能，以作為內部工作的統計、審核與管考資料。

### 3.2. 系統架構

數位典藏系統是以三層式(3-tier)架構所建構的，包含用戶端(client)、伺服器端(server)與資料庫系統。異動管理模組是數位典藏系統的一個模組，以 Java 與 JSP 開發，其部署圖如圖 6 所示。圖中描述了異動管理模組如何部署到實際的硬體環境，以用戶端為例，User、Client 和 Browser 分別代表節點名稱、節點類型與元件名稱。此外，節點之間的連線(connection)以直線顯示，而元件之間的相依關係以虛線加上方向來表示。

用戶端若要透過瀏覽器使用異動管理模組的功能，必須經由 TCP/IP 通訊協定先向網站或應用系統伺服器(Web/Application server)提出請求，以顯示畫面來

表 1. Record\_Log 資料欄位內容

| Name        | Type         | Nullable | Comments |
|-------------|--------------|----------|----------|
| ID          | numeric(5)   | N        | 自動編號     |
| RECORD_NO   | varchar(50)  | Y        | 記錄識別碼    |
| RECORD_TYPE | varchar(50)  | Y        | 記錄類型     |
| PROCESS     | varchar(50)  | N        | 執行程序     |
| USER_NAME   | varchar(50)  | N        | 使用者名稱    |
| GROUP_NAME  | varchar(200) | Y        | 群組名稱     |
| LOG_DATE    | Date         | N        | 異動日期     |
| REMARK      | varchar(200) | Y        | 備註       |

進行操作。操作異動管理模組時，若需要用到資料庫中的資料，網站或應用系統伺服器會透過 JDBC (Java Database Connectivity)向資料庫系統提出請求，以進行寫入或查詢的動作。

## 4. 剖析結果

本文根據 DAAL 所開發的異動管理模組程式和數位典藏系統核心元件及模組應用程式之介面說明，以傅斯年圖書館藏善本古籍數位典藏系統網站為例[6]，剖析異動管理模組。首先介紹異動記錄資料庫內容，再以 UML、MVC 架構來描述剖析結果，分別說明於以下各節。

### 4.1. 異動記錄資料庫

異動記錄資料庫中只有一個 Record\_Log 表格，而 Record\_Log 的資料欄位有：ID(自動編號)、RECORD\_NO(記錄識別碼)、RECORD\_TYPE(記錄類型)、PROCESS(執行程序)、USER\_NAME(使用者名稱)、GROUP\_NAME(群組名稱)、LOG\_DATE(異動日期)、REMARK(備註)，其中 ID 為主鍵(primary key)，用來識別不同的異動記錄。資料欄位內容顯示於表 1。

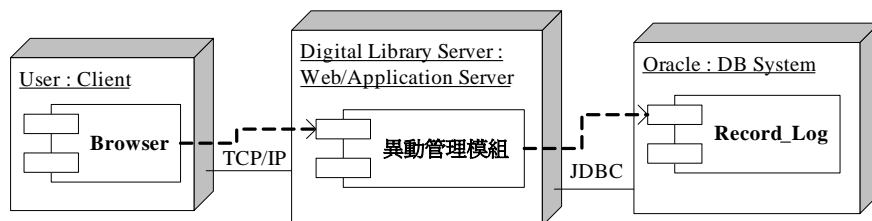


圖 6. 異動管理模組的部署圖

## 4.2. 用 UML 表示

本節使用 UML 中的使用案例圖、類別圖、循序圖來描述異動管理模組的功能、架構、元件與運作流程。

### 4.2.1. 使用案例圖

正如第 3 節所述，異動管理模組有兩個使用案例：寫入異動相關記錄和資料異動記錄檢索。畫出的異動管理模組使用案例圖如圖 7 所示，其中系統管理者、資料著錄人員與異動記錄資料庫都是行為者，而整個系統的範圍會以框線來界定。

由於異動記錄是作為內部工作之用，因此只有擁有權限的使用者可以使用資料異動記錄檢索這項功能。寫入異動相關記錄的功能並不是由一般使用者直接使用，而是當資料著錄人員對典藏資料進行新增、修改或删除時，才會由系統執行寫入異動相關記錄的功能。圖 7 的典藏資料著錄系統是指 metadata 管理系統中進行新增、修改或删除典藏資料時所用到的網頁。此外，執行寫入異動相關記錄和資料異動記錄檢索功能時，都會到異動記錄資料庫寫入或查詢資料。

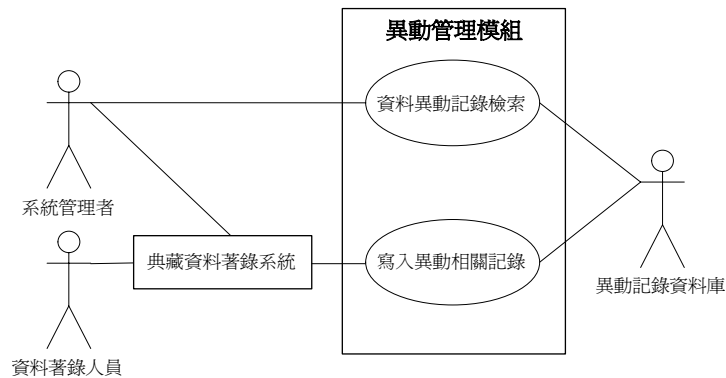


圖 7. 異動管理模組的使用案例圖

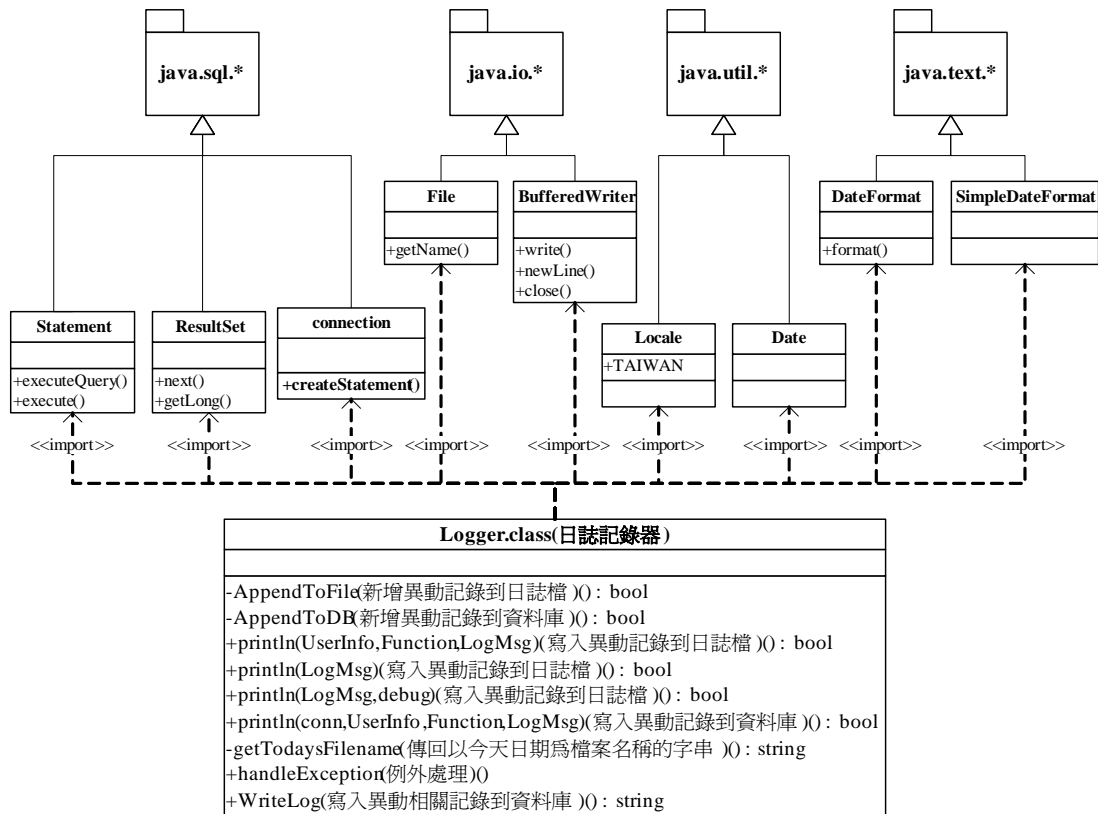


圖 8. 日誌記錄器的類別圖

#### 4.2.2. 類別圖

根據異動管理模組的程式，將類別(包含異動管理模組中的類別以及其所使用的套件與類別)與類別之間的關係找出，並畫出類別圖。所畫出的日誌記錄器、資料異動記錄檢索類別圖分別顯示於圖 8 和圖 9。

當透過典藏資料著錄系統新增、修改或刪除典藏資料時，會呼叫封裝在 edu.sinica.iis.cscl.util 之下的

Logger.class (日誌記錄器)來寫入異動相關記錄。如要新增一筆影像檔資料時，典藏資料著錄系統寫入影像檔資料到資料庫的同時，也會呼叫 edu.sinica.iis.cscl.util.Logger.WriteLog() 來寫入異動記錄。圖 8 的 Logger.class import 了 java.sql.\*、java.io.\*、java.util.\*、java.text.\*這四個套件，有使用到的類別、屬性和操作方法皆列在相對應的套件底下。

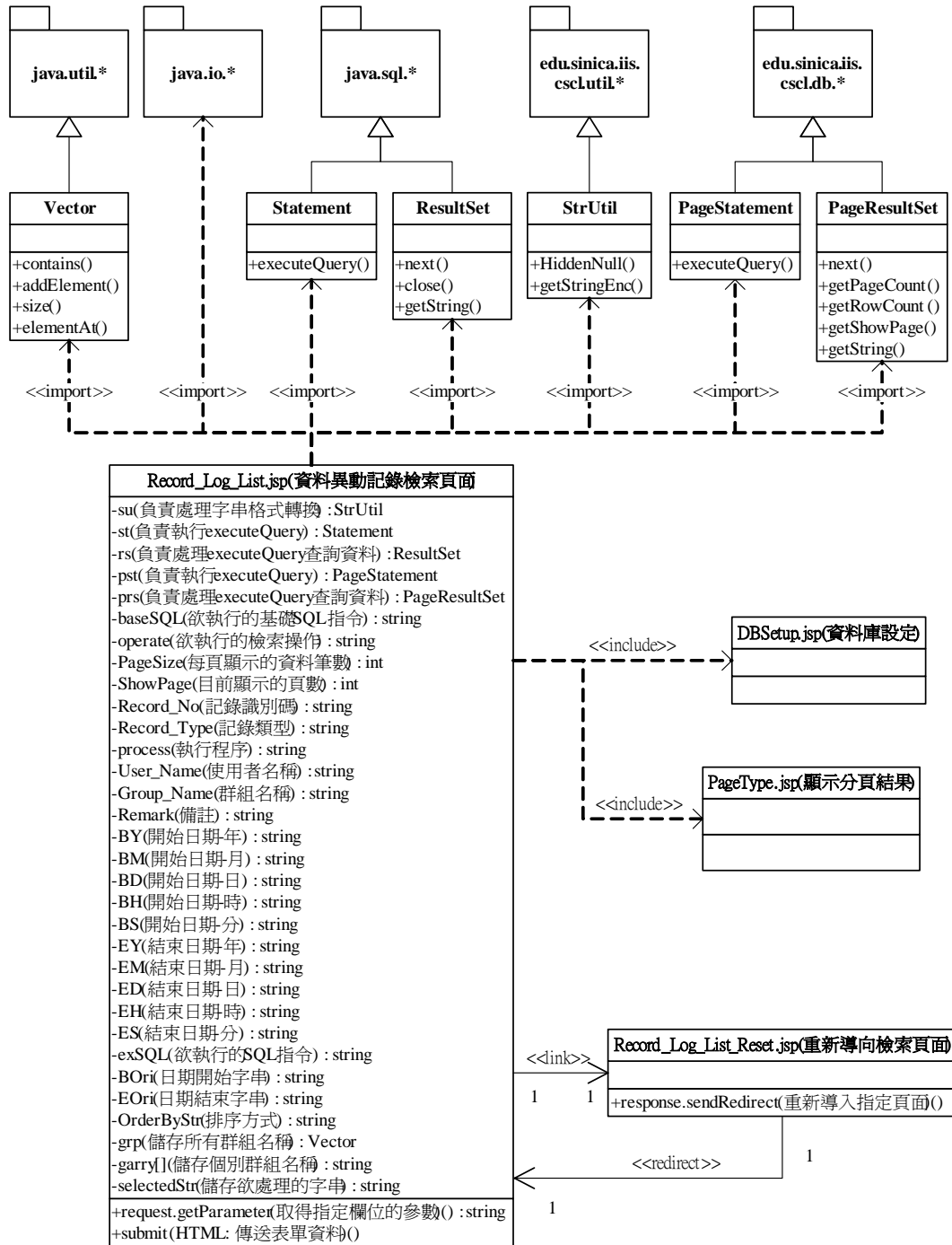


圖 9. 資料異動記錄檢索的類別圖

圖 9 的 Record\_Log\_List.jsp (資料異動記錄檢索頁面)主要是提供異動記錄檢索功能，透過網頁輸入欲查詢的條件，即可查詢到異動記錄。而 Record\_Log\_List\_Reset.jsp 的功用是當使用者想要回到初始查詢畫面時，只要點選網頁上的“清除”按鈕就會連結到 Record\_Log\_List\_Reset.jsp，之後再重新導向 Record\_Log\_List.jsp，以顯示初始查詢畫面。Record\_Log\_List.jsp import 了 java.util.\*、java.io.\*、java.sql.\*、edu.sinica.iis.cscl.util.\*、edu.sinica.iis.cscl.db.\* 這五個套件，有使用到的類別、屬性和操作方法皆列在相對應的套件底下。此外，Record\_Log\_List.jsp 還 include 了 DBSetup.jsp 和 PageType.jsp，其中

DBSetup.jsp 的功能是設定資料庫連結，而 PageType.jsp 功能則是顯示查詢的分頁結果。

#### 4.2.3. 循序圖

圖 7 的寫入異動相關記錄和資料異動記錄檢索使用案例所對應的循序圖分別顯示於圖 10 和圖 11。圖 10 中的典藏資料著錄系統、Logger.class、Record\_Log 都是物件個體；人(如圖 11 中的系統管理者)以外的物件個體皆以矩形表示。圖中生命線以垂直的虛線表示，而傳遞與回傳的訊息分別以實線和虛線來代表。此外，若要描述傳遞的訊息在什麼情形下會被送出，可以在傳遞的訊息加上“[條件]”來說明。

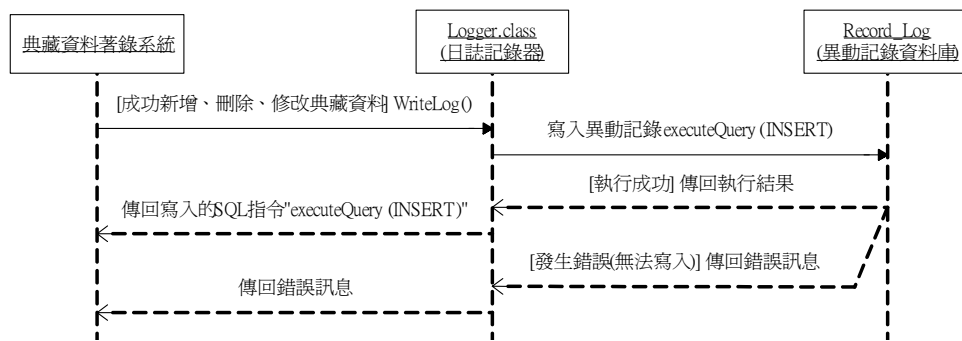


圖 10. 寫入異動相關記錄的循序圖

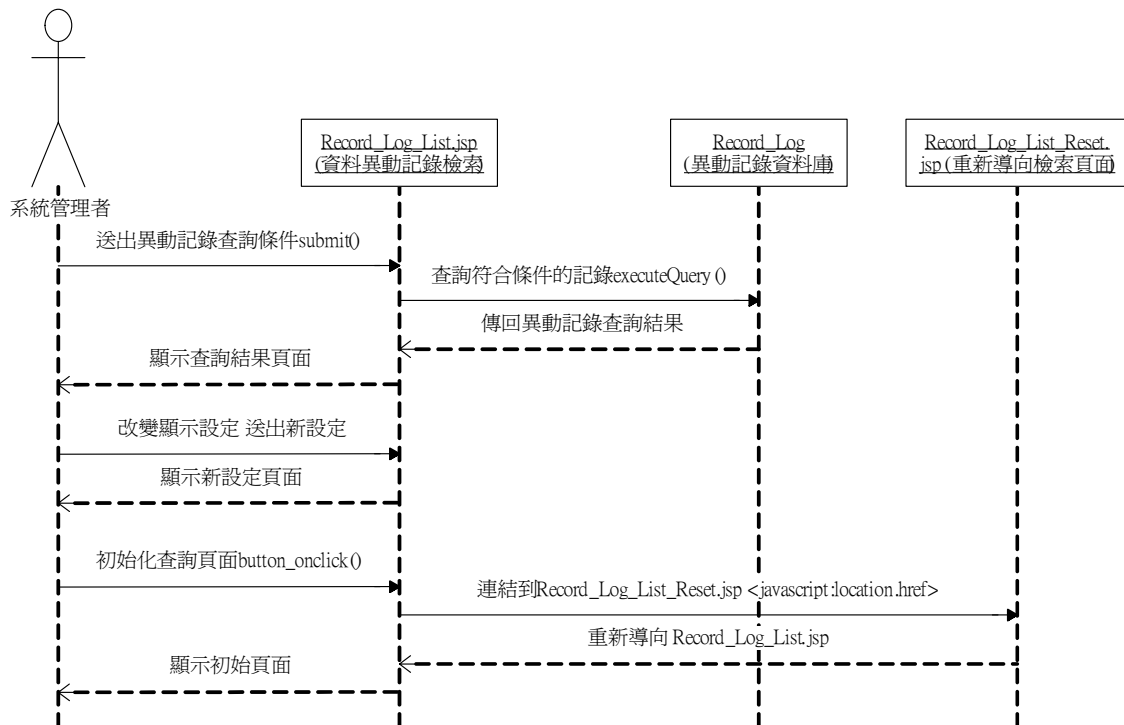


圖 11. 資料異動記錄檢索的循序圖

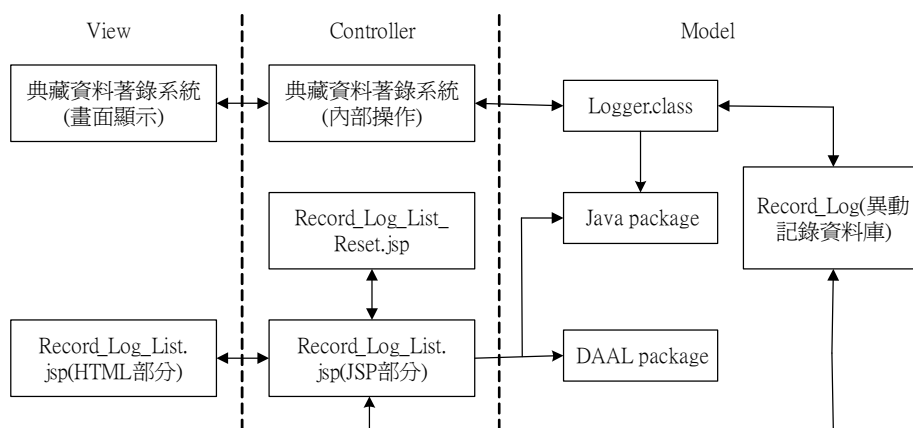


圖 12. 異動管理模組的 MVC 架構圖

循序圖是由圖形左上方的驅動者(driver)開始引發一連串的事件，在圖 10 之中，當資料著錄人員透過典藏資料著錄系統新增、修改或删除資料時，典藏資料著錄系統會呼叫 Logger.class 的 WriteLog()函式，接著將相關異動記錄寫入到 Record\_Log 中；若執行寫入成功，則會回傳寫入的 SQL 指令，反之則回傳錯誤訊息。

圖 11 的資料異動記錄檢索功能可再細分為：查詢異動記錄、改變顯示設定與初始化查詢頁面。查詢異動記錄需要輸入欲查詢的條件，再按下“查詢”按鈕，以送出表單到資料庫中查詢，並傳回結果。改變顯示設定需選擇每頁顯示筆數、排序條件或是顯示頁數，網頁就會顯示新設定的頁面。初始化查詢頁面則要點選“清除”按鈕，之後會重新導向初始頁面。

### 4.3. 用 MVC 表示

為了將網頁設計人員和程式設計人員的工作清楚區別出來，以提高模組化的特性以及加速開發程序，我們利用 MVC 架構將異動管理模組所使用的元件歸屬於 model、view、controller 三個模組，顯示於圖 12。

在 Record\_Log\_List.jsp 中包含了 HTML 和 JSP 兩種內容，在圖 12 中分別屬於網頁顯示和內部實作設計。圖中的典藏資料著錄系統並不屬於異動管理模組，但為了配合 Logger.class 的運作，所以在圖中加入典藏資料著錄系統，以說明兩者之間的關係。此外，圖中的 Java package 是指 Java 核心的套件，而 DAAL package 代表 DAAL 小組所開發的套件，包含數位典藏系統中共用、重要的元件(如資料庫存取、字串處理)。

## 5. 問題與建議

我們最後提出幾點剖析過程中發現的問題和建議，以供後續開發人員參考。

(1) 程式中有許多多餘的部分，降低了程式的可讀性。多餘的部分包含沒有作用的變數與測試用的程式碼，例如在 Record\_Log\_List.jsp 程式碼中的 NPG、LPG、FPG、PPG、checkReport(由於沒有作用所以不列在類別圖的屬性中)等變數。此外同樣在 Record\_Log\_List.jsp 中的一段程式碼：

```
// out.print(exSQL+"<br>");
```

這只是為了檢查執行的 SQL 指令是否正確，但程式完成後卻沒有將這段程式碼拿掉，造成閱讀上的困擾。因此，應刪除程式中多餘的部分，再調整程式碼編排上的一致性，以提升程式的可讀性。

- (2) 程式中缺乏詳細的註解，造成閱讀上的困難。建議加入足夠的註解，使後續開發人員能夠清楚瞭解程式內容，並可供未來修改與測試之用。
- (3) Record\_Log\_List\_Reset.jsp 的功能只是重新導回初始頁面，可以和 Record\_Log\_List.jsp 整合為一個 JSP，以降低程式的複雜性。
- (4) 整個異動管理模組網頁顯示和內部實作的部分過於緊密結合，造成設計人員無法專注於各自的設計工作。建議利用 MVC 架構來開發系統，以分開網頁設計和程式設計，進而提升模組化的程度。
- (5) 所有數位典藏系統中的異動管理模組皆有相似的元素和架構。為避免重複的分析與開發，異動管理模組的 JSP 中若有共通的部分，可以整合為一

個類別，以供所有數位典藏系統使用。由於異動管理模組資料庫的架構在所有數位典藏系統都應相同，建議以系統化的方式產生資料庫架構，以便於其他數位典藏系統重複使用。

- (6) 為建構未來大量的數位典藏系統，建議透過自動化的方式來產生異動管理模組，以加速新系統建置的速度。

## 6. 結論

本文透過 UML 及 MVC 架構來描述數位典藏系統的異動管理模組。UML 圖形可以清楚地表達模組的功能、架構、元件與運作流程；MVC 架構將模組的顯示、控制和內部處理分開以提高模組化。剖析結果有助於找出可重複使用的元件或架構，以提供給各領域的數位典藏單位參考，希望能減少重複的分析與開發，以加速新系統的建置，並提高數位典藏系統的品質。

## 參考資料

- [1] J. Conallen, *Building Web Applications with UML*, 2nd ed., Addison-Wesley, Boston, MA, 2002.
- [2] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley, Boston, MA, 1999.
- [3] *Model-View-Controller*,  
<http://java.sun.com/blueprints/patterns/MVC.html>.
- [4] *UML 2.0 Superstructure Specification*,  
<http://www.omg.org/cgi-bin/doc?ptc/2004-10-02>.
- [5] 中央研究院數位典藏技術發展組，數位典藏系統核心元件及模組，  
<http://ndweb.iis.sinica.edu.tw/api/index.html>.
- [6] 中央研究院數位典藏技術發展組，傅斯年圖書館藏善本古籍數位典藏系統，  
<http://140.109.18.243/rarebook/>.