

後設資料管理系統產生器

王祥安 郭令斌 林彥君

國立台灣科技大學資訊工程系

{D9215004, M9215034, y.lin}@mail.ntust.edu.tw

摘要

本研究發展一個後設資料(metadata)管理系統產生器,讓使用者可以輸入 HTML 檔案而建立後設資料管理系統。我們使用 Model-View-Controller 的觀念建立後設資料管理系統的程式,並以 XML 文件儲存後設資料。

關鍵詞：後設資料管理系統、HTML、Model-View-Controller、XML

1. 簡介

本研究以中央研究院資訊科學研究所數位典藏國家型科技計畫技術發展組(Digital Archive Architecture Laboratory, 簡稱 DAAL)所開發的數位典藏系統之架構基礎,實作一個後設資料管理系統產生器。DAAL 協助了多個典藏機構開發數位典藏系統,如中央研究院傅斯年圖書館之善本古籍系統、歷史語言所之漢代簡牘系統等,這些系統各自有一個後設資料管理系統以管理後設資料。由於各典藏系統的後設資料特性不同,後設資料也有很大的差異。因此,在建立一個新的數位典藏系統時,需要設計一組新的後設資料管理系統,以配合典藏單位的需求。

雖然不同後設資料管理系統的資料結構差異性大,但是處理資料的功能(如新增、修改、刪除、查詢與列表)與流程大部分是相同的。因此,在建立新的後設資料管理系統時,可以依循過去已發展的系統設計方式來實作。

DAAL 在建立後設資料管理系統之前,系統分析人員會先了解典藏單位的需求,然後製作軟體需求規格書與軟體設計規格書[2, 4],接著由視覺設計人員設計使用者介面,並製作成為 HTML 檔案。待確定介面的外觀、流程與資料欄位後,程式設計人員會根據軟體設計規格書將 HTML 檔案修改成 JavaServer Pages (JSP)網頁程式。HTML 檔案與 JSP 網頁程式兩者之差異在於前者只有靜態的介面外觀、流程與欄位等,無法存取資料庫的內容;而後者具有存取資料庫與動態產生網頁內容的能力。

因為 DAAL 製作的各個不同後設資料管理系統僅在後設資料的架構上不同,而功能與運作流程相似,所以我們整理了這些系統共同的功能與運作流程,設計一套後設資料管理系統產生器(以下簡稱產生器)。透過產生器可以減少程式設計人員將 HTML 檔案轉換成 JSP 網頁程式的工作,讓典藏單位可以透過此產生器,自行產生後設資料管理系統,供後設資料維護者(以下簡稱維護者)使用。

目前可以產生中文數位典藏後設資料管理系統的工具具有[7]:

- (1) Metalogy: 由國科會數位博物館專案計畫發展,可讓使用者輸入後設資料的 XML DTD 文件以產生後設資料管理系統。
- (2) ODAE: 由中央研究院資訊科學所發展,可讓使用者在系統上定義後設資料之結構,以產生後設資料管理系統。
- (3) DADT: 由中央研究院計算機中心發展,可讓使用者定義後設資料之資料庫儲存綱要(schema),以產生後設資料管理系統。

這些後設資料管理系統產生工具的共通特性,都是根據使用者對後設資料的定義,產生固定樣式的使用者介面之後設資料管理系統,無法讓使用者自訂介面。它們的資料皆以關聯式資料庫儲存,且後設資料管理系統被建立後即無法彈性修改後設資料的結構。我們的後設資料管理系統產生器可讓使用者輸入 HTML 檔案,以產生後設資料管理系統。至於資料,是以 XML 文件儲存。相較於其它系統,除了可以讓使用者自訂介面,讓產生的系統畫面可以符合使用者的規劃,並由於採用 XML 文件儲存資料,系統建立後仍可以彈性的修改後設資料的結構。

本論文的組織架構如下:第 2 節說明 DAAL 數位典藏系統的架構並說明使用 XML 文件與 Model-View-Controller (MVC)技術的目的。第 3 節介紹我們實作的後設資料管理系統產生器,包含架構、所使用的軟體與技術。第 4 節說明產生器的操作方式,以及用一個例子說明所建立的後設資料管理系統架構與功能。第 5 節說明產生後設資料管理系統時,程式運作的步驟及後設資料管理系統儲存資料的方式。第 6 節比較我們所產生的後設資料管理系統與 DAAL 的數位典藏系統的差異。第 7 節為結論,並提出未來可改進發展的方向。

2. 數位典藏系統與實作技術介紹

本節首先介紹 DAAL 數位典藏系統架構,其次說明以 XML 文件儲存後設資料的優點,以及利用 MVC 觀念開發應用程式的觀念。

2.1 DAAL 之數位典藏系統

DAAL 所開發的數位典藏系統之共同架構如圖 1 所示,簡介如下:

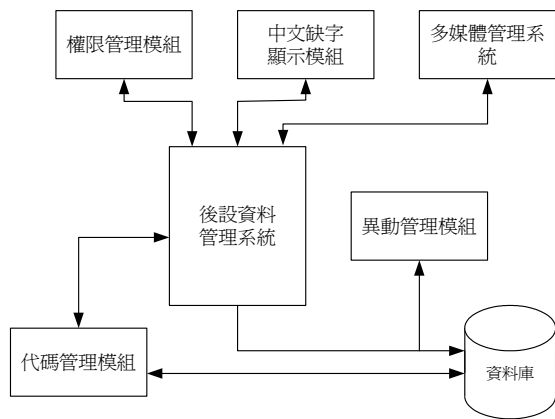


圖 1. DAAL 數位典藏系統架構

- (1) 後設資料管理系統：由 JSP 網頁程式、JavaBeans (一種提供特定功能且可重複使用的軟體元件 [10]) 所構成的系統，具有列表、新增、修改、刪除後設資料的功能。
- (2) 權限管理模組：管理使用者資料、登入權限與限制使用者所能執行的功能。
- (3) 代碼管理模組：代碼為表單欄位中下拉式選單所呈現的選項。代碼管理模組提供管理者管理代碼。
- (4) 異動管理模組：當使用者對後設資料進行新增、修改、刪除等動作時，異動管理模組會記錄哪一位使用者於何時對哪一筆後設資料進行何種處理 [3]。
- (5) 中文缺字顯示模組：在編輯後設資料的過程中若出現系統字元編碼沒有定義的文字(即缺字)時，將無法被正確的顯示，因此需要中文缺字顯示模組來處理，以正確顯示缺字 [5]。
- (6) 多媒體管理系統：可管理圖像、影音檔案，並提供編輯(如旋轉圖像)、轉換(如變換圖像的解析度)、安全保護(如加入浮水印)等功能。

DAAL 的數位典藏系統的組成方式，是以後設資料管理系統為中心，結合上述模組與多媒體管理系統所組成。其中，後設資料管理系統會因為各典藏單位的後設資料架構的不同而改變，而其它模組與多媒體管理系統則都是相同的。

後設資料管理系統的運作流程如下。首先，將資料庫中的後設資料以列表的方式顯示。接著，使用者可以選擇欲修改或刪除的後設資料，或新增後設資料。最後，透過後端資料處理程式將後設資料存入資料庫中。

由於 DAAL 數位典藏系統的架構中，僅後設資料管理系統會因後設資料架構的改變而有所不同，而後設資料管理系統的運作流程又有固定模式，因此本研究以此為基礎，發展只要使用者輸入後設資料架構，即可自動產生後設資料管理系統的產生器。

2.2 以 XML 文件儲存後設資料的優點

以 XML 文件儲存後設資料的方法優於以關聯式資料庫儲存 [1, 8]。其理由如下：

- (1) 不需要正規化後設資料結構。多數的後設資料

是樹狀結構，若以關聯式資料庫儲存，需將其正規化成數個具關聯性的資料綱要(schema)。XML 文件是樹狀結構，可以直接呈現後設資料結構，不需要進行資料正規化。

- (2) 易於儲存後設資料。在關聯式資料庫中，每一筆後設資料都需要儲存在多個與其結構相對應的表格中。若以 XML 文件儲存後設資料，則可以將後設資料儲存於一個 XML 文件中，在儲存上較關聯式資料庫容易。
- (3) 後設資料結構可彈性變動。若以關聯式資料庫儲存後設資料，當後設資料結構變動時，需要修改關聯式資料綱要；而以 XML 文件儲存後設資料時，可直接改變 XML 文件資料結構。

2.3 MVC 設計觀念

為了提升程式的模組化程度與再利用性，並降低更新與維護時的困難，以 JSP 開發大型或複雜的系統時，常會使用 MVC 的觀念設計程式 [6, 14]。MVC 是一種設計模式(design pattern)，被廣泛的用於軟體設計。MVC 將應用程式的架構區分成 model、view、controller 三個部分。其中 model 負責資料處理；view 是向使用者展示資訊(包含從 model 得到的資料)，與接收使用者的輸入；controller 會根據從 view 所接收的事件去呼叫 view 或 model 中相對應的功能 [9]。model、view、controller 之間的關係如圖 2。由於 model、view、controller 各部分的相依性低，所以應用程式的修改與更新較為容易，亦可以提高各部分的模組化程度與再利用性。

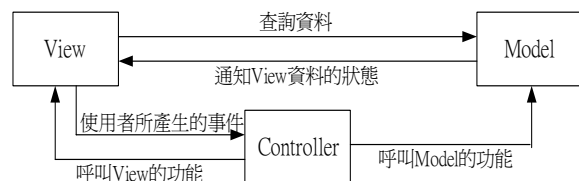


圖 2. MVC 架構圖

一般 JSP 網頁程式的架構可以分為 model 1、model 2 兩種 [6, 14, 15]。model 2 架構即為 MVC 架構；model 1 則讓資料顯示與流程控制由同一個 JSP 網頁程式處理。DAAL 的後設資料管理系統是以 model 1 架構設計，資料顯示與流程控制會互相影響，當資料顯示修改時，控制流程的程式也需要一起修改。因此，每當建立一個新的後設資料管理系統時，因為後設資料顯示不同，必需重新撰寫程式。如果後設資料管理系統是 MVC 架構，則建立新系統時只需修改顯示資料的部分，控制流程的程式可以重複使用，無須重新撰寫。因此，產生器以 MVC 架構建立後設資料管理系統，只需把介面與處理資料、運作流程的程式產生關連，可降低產生器建立後設資料管理系統的處理複雜度。

3. 後設資料管理系統產生器

本節介紹我們的產生器的架構並說明開發過程所使用的軟體與技術。產生器建置於伺服器，使

用戶需透過瀏覽器來使用產生器所提供的功能，其架構如圖 3 所示，說明如下。

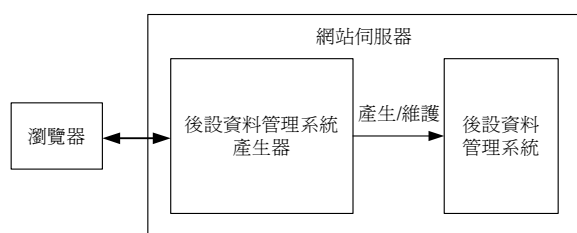


圖 3. 後設資料管理系統產生器架構

- (1) 瀏覽器。位於使用者端的程式(如 Firefox)，將使用者所發出的要求送至網站伺服器，並顯示從網站伺服器傳回的資訊。
- (2) 後設資料管理系統產生器。提供使用者可以自行建立並維護所需要的後設資料管理系統。
- (3) 後設資料管理系統。用來管理後設資料，包含後設資料維護、檔案管理、權限管理、異動記錄、代碼管理、表單欄位管理功能。

我們以 Resin 為應用伺服器軟體[16]，以 JSP 開發產生器。上傳檔案的功能以 JavaZOOM [11]所開發的 jClientUpload 製作。以 XML 格式儲存後設資料，存取 XML 文件是透過 SUN 所開發的 DOM API [12]與 SAX API [13]。

4. 建立後設資料管理系統

本節說明使用者操作產生器建立後設資料管理系統的過程，並用一個例子說明所產出的後設資料管理系統架構與功能。

4.1 產生後設資料管理系統的過程

使用者以產生器製作後設資料管理系統的操作過程，可分為以下兩部份：

1. 建立專案

當使用者登入產生器後，可在專案管理功能中新增專案。在輸入專案名稱與代號後，按下新增的按鈕即可新增專案。專案新增後會為後設資料管理系統自動產生『檔案管理』、『權限管理』、『代碼管理』、『異動記錄』、『表單欄位管理』等功能。新增專案畫面如圖 4。當一個新專案被建立時，產生器會為該專案建立一個以專案代號命名的目錄，然後在該目錄下，把『權限管理』的 JSP 網頁程式複製到 auth 目錄中，並將『異動記錄』、『代碼管理』、『表單欄位管理』的 JSP 網頁程式複製到 system_folder 目錄中。『檔案管理』是內建功能，不需額外產生 JSP 網頁程式。

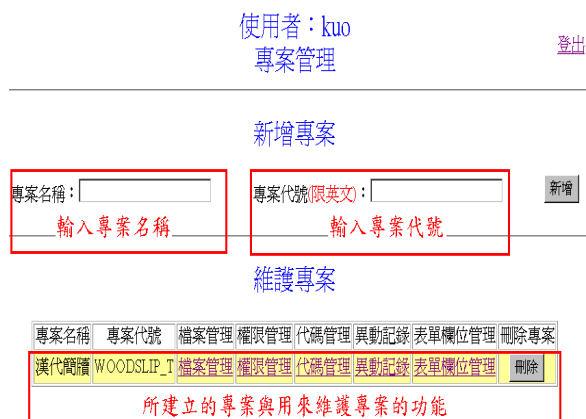


圖 4. 新增專案畫面

2. 建立『後設資料維護』

『後設資料維護』是後設資料管理系統的一個主要功能，建立『後設資料維護』的功能附屬於『檔案管理』功能下。以圖 4 為例，在點選『檔案管理』的超連結後，會顯示該專案的所有檔案與資料夾，並提供使用者『上傳檔案』、『刪除檔案』、『新增資料夾』、『刪除資料夾』、『建立功能』等功能，畫面如圖 5。使用者可以上傳檔案至目前的目錄中。

建立『後設資料維護』必須透過圖 5 中『建立功能』達成。使用『建立功能』前，使用者必須上傳 HTML 檔案，系統的建立則是透過分析 HTML 檔案並加以處理後完成。上傳的 HTML 檔案須符合產生器所規定的語法，內容需要包含用來描述畫面的資訊(如文字、顏色等)與描述後設資料的欄位(如主要名稱)。以基本資料著錄的 HTML 檔案為例，其畫面如圖 6。



圖 5. 「檔案管理」的畫面

基本資料著錄

開放檢索	-----	
識別資料	遺物登錄號	
	簡號	
品名	主要名稱	
所屬遺址	遺址識別碼	
	遺址代碼	
	遺址名稱	
詳細地點		
形制	分類	-----
合編資料	單件/合編	合編
	合編遺物編號	合編子項編號

圖 6. 基本資料著錄 HTML 檔案的畫面

使用者所上傳的 HTML 檔案需遵守產生器的下列規定：

- 使用單行文字方塊(textbox)、多行式文字方塊(textarea)、下拉式選單(option)來建立後設資料的欄位。產生器只支援這些類型的欄位。
- 使用「多值」、「連結」、「圖片」、「多階層選單」欄位型態時，必須使用下列的標籤(tag)宣告：
 - 多值欄位：`<multi_value ame="欄位名稱"></multi_value>`
 - 連結欄位：`<reference name="欄位名稱"></reference>`
 - 圖片欄位：`<picture name="欄位名稱"></picture>`
 - 多階層選單欄位：`<level_option name="欄位名稱"></level_option>`
- 其它規定：
 - 欄位屬性的值要用雙引號括住。例如：`<input name=aaa>`需改成`<input name="aaa">`。
 - 欄位宣告不能放在同一行。例如：`<input name="aaa"><input name="bbb">`需分成兩行撰寫。
 - 不要使用`<form>`標籤，產生器會自動產生。
 - 選擇鈕(radio)改以下拉式選單(select)表示。

當使用者上傳 HTML 檔案後，再按下如圖 5 中的『建立功能』按鈕，即進入建立『後設資料維護』的過程。首先需輸入『後設資料維護』的中、英文名稱，替所產生出的查詢、新增、修改與刪除功能的網頁程式命名。接著，產生器會分析該 HTML 檔案，並產生如圖 7 的表單欄位設定畫面。此畫面顯示該 HTML 檔案所包含的欄位名稱與型態，使用者需在此畫面中對於各欄位進行相關的設定。其設定方式與 4.2 節的『表單欄位管理』類似，請參考其說明。

當所有設定都完成後，再按下圖 7 中的確定按鈕，此時會把這些設定記錄於表單欄位定義的 XML 檔案中，並產生列表與查詢、新增、修改與刪除等功能的 JSP 網頁程式。其中新增、修改與刪除的畫面是根據所選擇的 HTML 檔案的畫面所決定，因此這些畫面與 HTML 檔案的畫面相似。

表單欄位定義設定畫面

是否於列表畫面顯示	選擇型連結名稱	欄位名稱	欄位型態	欄位限制
X	e	索引	索引	X
r	e	開放檢索	下拉式選單	X
r	e	遺物登錄號	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	簡號	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	主要名稱	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	遺址識別碼	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	遺址代碼	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	遺址名稱	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	詳細地點	單行文字方塊	*文字 字數限制: 0 (格式: 1-1-100-0(不限制)) e 數字 範圍限制: 0 (格式: 1-100-0(不限制)) c 日期 (格式: YYYYMM/DD)
r	e	分類	下拉式選單	X
r	X	合編遺物編號	連結	X
r	X	合編子項編號	連結	X
X	X	其他名稱	多值欄位	選擇
X	X	描述	多值欄位	選擇

圖 7. 表單欄位的設定畫面

4.2 後設資料管理系統

後設資料管理系統建置於伺服器端，使用者與維護者需透過瀏覽器來使用其功能，其架構參見圖 8。

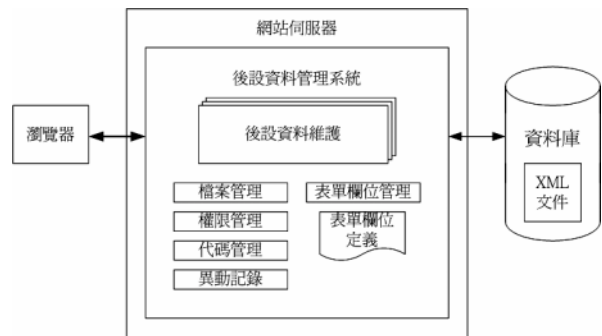


圖 8. 後設資料管理系統架構

產生器所建立的後設資料管理系統包括『後設資料維護』、『檔案管理』、『權限管理』、『代碼管理』、『異動記錄』、『表單欄位管理』功能。每一個後設資料管理系統各有其專屬的資料庫，存放記錄後設資料的 XML 文件，還有一個被稱為『表單欄位定義』的 XML 文件，它記錄後設資料管理系統的相關資訊。以下我們以產生器所產出的漢代簡牘後設資料管理系統為例，說明各個功能：

1. 後設資料維護

提供維護者查詢、列表、新增、修改、刪除等功能，讓維護者可以處理後設資料。操作方式如下：首先，維護者在登入系統後，可選擇要維護的後設資料項目，然後進入該後設資料的列表與查詢畫面。其列表畫面如圖 9，列出已存在的後設資料部分內容，維護者可以按下編號欄位中的超連結，以觀看其詳細內容，或修改、刪除該筆資料。

基本資料列表

新增資料 查詢資料				
總筆數：2筆				
每頁顯示 25 筆, 目前頁數/總頁數:1/1				
編號	開放檢索	遺物登錄號	簡號	主要名稱
Artifacts1	開放	H02103	071.003	漢簡
Artifacts2	開放	H02104	071.004	漢簡

選擇頁面：
1

圖 9. 後設資料的列表畫面

維護者可在圖 9 的畫面中選擇『查詢資料』的按鈕以進入圖 10 的查詢畫面。維護者可輸入關鍵字查詢後設資料，查詢的結果以列表的方式顯示。

開放檢索	不限
遺物登錄號	
簡號	
主要名稱	
遺址識別碼	
遺址代碼	
遺址名稱	
詳細地點	
分類	不限
合編遺物編號	
合編子項編號	
多值欄位"描述"	
描述	不限
多值欄位"其他名稱"	
類型	
名稱	
備註	

查詢 回上一頁

圖 10. 後設資料的查詢畫面

在列表的畫面點選後設資料編號的超連結後，即進入如圖 11 所示的修改後設資料的畫面。此畫面列出後設資料的詳細資訊，維護者可以修改後設資料的各個欄位或刪除此筆後設資料。

若維護者想新增一筆後設資料，可在圖 9 的列表畫面中，選擇『新增資料』的按鈕以進入新增資料畫面。『新增資料』的畫面與圖 11 的修改後設資料的畫面相似，其差異僅在於新增資料畫面中，欄位皆為空白，而圖 11 則有過去已輸入的資料。

後設資料的輸入與呈現需求，除了一般的文字欄位之外，還包括多值處理、外部資料參照、圖片資料、多階層式資料結構。後設資料管理系統提供四種特殊欄位來處理上述需求：

- (1) 多值欄位。在某欄位中所要描述的內容個數，可能是超過一個以上。如圖 11 中的「形制」欄位，內容包括「刻齒」與「繫繩圓孔」。

基本資料著錄

開放檢索	開放
識別資料	遺物登錄號 H02103
	簡號 071.003
	主要名稱 漢簡
品名	新增
	類型/名稱 備註
	刪除 修改 通稱/居延漢簡
所屬遺址	遺址識別碼 HA08
	遺址代碼 A8
	遺址名稱 破城子
詳細地點	
形制	分類 兩行
	新增
	描述
	刪除 修改 刻齒
	刪除 修改 繫繩圓孔
合編資料	單件/合編 合編
	合編遺物編號 合編子項編號
	新增超連結 新增超連結
	Aggregation1 刪除 Artifacts1 刪除 Artifacts2 刪除

修改資料 刪除資料 重設資料

返回列表與查詢畫面

圖 11. 後設資料修改的畫面

- (2) 連結欄位。此欄位可輸入超連結的位置，超連結的目標可以是一個網頁或一個物件，如圖 11 的「合編資料」欄位以後設資料的修改畫面為超連結的目標。

- (3) 圖片欄位。提供維護者上傳圖片及顯示圖片。
- (4) 多階層選單欄位。當一個欄位的值是以選單的方式輸入後，會影響到下一個欄位的內容（即選單），且欄位與欄位間具有階層關係時，則使用此欄位設計。

2. 檔案管理

讓使用者上傳自己撰寫的網頁或多媒體檔案，產生器可依網頁內容建立其所需要的後設資料管理系統。『檔案管理』也提供使用者管理所上傳的檔案。

3. 權限管理

為了防止未被授權的使用者，後設資料管理系統有兩種保護方式：(1)要求維護者輸入帳號及密碼登入系統以避免非法使用。(2)管制已登入的維護者可以使用的功能，以避免不當操作。例如：某維護者只能使用列表與查詢的功能，不能使用新增、修改與刪除的功能。

4. 異動記錄

當後設資料的內容變動時，系統會自動記錄哪一位維護者，在何時異動了哪一筆後設資料，及異動的方式（如刪除）。維護者可在『異動記錄』中檢視這些資料。

5. 代碼管理

在編輯後設資料的內容時，以下拉式選單供維護者選擇內容，以快速且正確方式輸入。下拉式選單的內容即為『代碼』。代碼管理可讓維護者新增、修改、刪除代碼的內容。

6. 表單欄位管理

可用來管理列表、新增、修改畫面中所呈現的表單欄位，其畫面如圖 12。它具有以下三種功能：

- (1) 設定列表畫面中所要顯示的欄位。當一份後設

資料有很多欄位時，若把所有欄位都同時顯示在如圖 9 的列表畫面上，會使畫面混亂而不易閱讀。透過此功能，可設定要在列表畫面上顯示哪些欄位的資料。

- (2) 設定超連結文字。當維護者利用連結欄位的功能建立超連結時，可以將超連結的目標設定為任一筆已存在的後設資料的修改畫面的網址，這時超連結文字將由系統自動產生，預設值為該後設資料的編號(例如 Artifacts1)，使用者可以在此設定後設資料中某一項欄位的內容為超連結文字。
- (3) 設定欄位輸入後設資料時所需遵守的限制。例如可設定某些欄位只能輸入數字，或是限制輸入字串的長度等，以減少輸入錯誤。如果欄位為圖片欄位，則可設定圖片顯示的長與寬。

表單欄位定義設定畫面

是否於列表畫面顯示	選擇欄位名稱	欄位名稱	欄位型態	欄位限制
<input checked="" type="checkbox"/>	e	索引	索引	X
<input type="checkbox"/>	e	開啟檢查	下拉式選單	X
<input checked="" type="checkbox"/>	e	遺物登錄號	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input checked="" type="checkbox"/>	e	編號	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input checked="" type="checkbox"/>	e	主要名稱	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input checked="" type="checkbox"/>	e	地址識別碼	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input type="checkbox"/>	e	地址代碼	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input type="checkbox"/>	e	地址名稱	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input type="checkbox"/>	e	詳細地點	單行文字方塊	* 文字 字數限制: F (格式: 1-100-0(不限制)) * 數字 範圍限制: F (格式: 1-100-0(不限制)) * 日期 範圍限制: F (格式: YYYYMM/DD)
<input type="checkbox"/>	e	分類	下拉式選單	X
<input checked="" type="checkbox"/>	X	含編號登錄號	連結	X
<input checked="" type="checkbox"/>	X	含電子項編號	連結	X

功能1 功能2 功能3

圖 12. 表單欄位管理的畫面

以上設定的內容，都會被記錄於表單欄位定義的 XML 檔案，包含各『後設資料維護』的中文名稱與英文代號、所使用的代碼，以及各欄位的名稱、類型、欄位限制、是否在列表畫面出現、是否為超連結欄位等。

5. 產生器的程式運作方式

本節說明產生器產生後設資料管理系統的程式運作步驟，與後設資料管理系統儲存後設資料的方式。5.1 節說明如何將 MVC 的觀念運用於後設資料管理系統的產生。5.2 節說明分析使用者上傳的 HTML 檔案到產生後設資料管理系統的 JSP 網頁程式之過程。5.3 節說明後設資料管理系統儲存資料的方式。

5.1 運用 MVC 於後設資料管理系統的產生

產生器在建立『後設資料維護』功能時，會產生管理後設資料功能的 JSP 網頁程式。其作法是參考 DAAL 數位典藏系統的設計方式，並以 MVC 的

觀念設計 JSP 網頁程式的架構。MVC 的觀念把每一個 JSP 網頁程式分成後端的資料處理(model)、後設資料顯示(view)、流程控制(controller)三個部份。資料處理負責存取資料庫中的後設資料。後設資料顯示則將資訊顯示至瀏覽器，並接收來自瀏覽器的訊息。流程控制則根據接收的訊息執行相關的動作，包含動態改變頁面的內容，或是呼叫其它功能的 JSP 網頁程式，其架構如圖 13。其中後端資料處理、流程控制可以設計成與其處理的後設資料無關，因此可以將其模組化以便重複使用。因為後設資料的不同會造成外觀上的不同、儲存資訊的不同，所以會使得程式產出時有所差異，但其影響範圍僅在後設資料顯示部份，因此可以提升 JSP 網頁程式的模組化程度，以及再利用性。

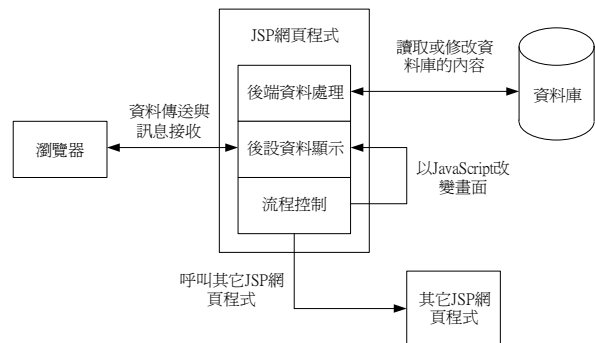


圖 13. JSP 網頁程式的架構

5.2 HTML 檔案轉換成 JSP 網頁程式

當使用者執行『後設資料維護』功能時，產生器會分析 HTML 檔案的內容，並根據此內容產生兩個 JSP 暫存檔，其中一個是用於後設資料新增畫面的顯示，另一個則用於後設資料修改畫面的顯示。

產生 JSP 暫存檔時，會把 HTML 檔案中各種不同類型的欄位，插入特定的函式(function)，或是由相對應的函式取代。以圖 6 的 HTML 檔案畫面為例，遺物登錄號旁的單行文字方塊在 HTML 檔案中的內容為

在欄位轉換完畢後，產生器會在 JSP 暫存檔中加入表單<form>標籤，其中會包含型態為隱藏(hidden)的單行文字方塊與型態為按鈕(button)的功能按鈕，如圖 14 的第 3 部分。其主要目的是讓系統可依內容判斷存取 XML 文件的方式，並決定按下功能按鈕後所要執行的功能。

```

<jsp:include page="/WOODSLIP_T/auth/AccessControl.jsp" > 1
<jsp:param name="chkName" value="Artifacts_add" />
</jsp:include>

<%@ include file="/WOODSLIP_T/system_folder/add_form_prefix.jsp" %> 2

<form name=edit_table action="" method="post">
<input name="not_reset" type="hidden" value="not_reset">
<input name="editing_xml_name" type="hidden" value="<%= editing_xml_name %>">
<input name="process" type="hidden" value="<%= process %>">
<input name="root_or_branch" type="hidden" value="root">
<input name="metadata_index" type="hidden" value="<%= metadata_index %>">
<input name="next_page" type="hidden" value="/WOODSLIP_T/Artifacts_list.jsp">
. . . . .
<tr>
<td width="33%" rowspan="2">識別資料</td>
<td width="33%">遺物登錄號</td>
<td width="34%">
<input type="text" name="遺物登錄號" / value="<%=get_text("遺物登錄號")%>" ></td>
</tr>
. . . . .
<p align="center">
<input name="send_out" type="button" value="新增資料" onClick="javascript:form_submit()">
<input name="clear" type="button" value="清除資料" onClick="javascript:clear_data()">
</p>
</form>
<%@ include file="/WOODSLIP_T/system_folder/add_form_postfix.jsp" %> 2

```

圖 14. JSP 暫存檔修改的各部分

接著，產生器會再加入兩個引入指令(include directive)，如圖 14 的第 2 部分。一個語法為<%@ include file="檔案的路徑" %>，置於 JSP 暫存檔前端，用於資料處理，負責把指定的後設資料從 XML 文件中取出。另一個引入指令置於 JSP 暫存檔的最後端，用於流程控制，負責控制各個功能按鈕執行後，所對應的 JSP 網頁處理程式。而被引入的 JSP 網頁程式是固定的程式模組，不同的 JSP 暫存檔均可適用。

最後，於 JSP 暫存檔的最前端會再加入<jsp:include>動作元素(action element)，並搭配<jsp:param>指定被引入的 JSP 網頁程式之參數，用於引入檢查維護者權限的 JSP 網頁程式，如圖 14 的第 1 部分。

當使用者完成『後設資料維護』後，產生器會將把兩個 JSP 暫存檔存成負責新增功能的 XXX_Add.jsp、負責修改功能的 XXX_Modify.jsp，並產生與負責後設資料列表與查詢的 XXX_List.jsp ("XXX"是 metadata 表單的英文名稱)。

5.3 後設資料管理系統儲存資料的方式

本系統使用 XML 文件儲存後設資料，並以 DOM API 與 SAX API 存取 XML 文件的內容。在新增或修改後設資料時，系統會根據『表單欄位定義』中對於各欄位的定義，將欄位的內容存成指定格式的 XML 文件，如圖 15 所示。XML 文件的根元素(element)名稱由系統產生，如 Artifacts1。根元素中包含兩種子元素，multi 元素記錄多值欄位的資訊，其它欄位的資訊記錄在 single 元素中，每一個元素都會有 name 屬性，用來記錄內容是來自哪個欄位。當讀取 XML 文件時，程式根據元素名稱與

name 屬性來取得相關的內容。

以 XML 文件儲存後設資料，若後設資料結構變動，過去已建立的 XML 文件內容不需改變，可彈性改變後設資料的結構。

```

- <Artifacts1>
+ <multi name="描述">
- <multi name="其他名稱">
- <item>
  <single name="類型">通稱</single>
  <single name="名稱">居延漢簡</single>
  <single name="備註" />
</item>
</multi>
<single name="開放檢索">2</single>
<single name="遺物登錄號">H02103</single>
<single name="簡號">071.003</single>
<single name="主要名稱">漢簡</single>
<single name="遺址識別碼">HA08</single>
<single name="遺址代碼">A8</single>
<single name="遺址名稱">破城子</single>
<single name="詳細地點" />
<single name="分類">4</single>
<single name="合編遺物編號">&%
&/WOODSLIP_T/Aggregation/Aggregation_modify.jsp?
metadata_index=Aggregation1#%#
#Aggregation1#%#Aggregation</single>
<single name="合編子項編號">&%
&/WOODSLIP_T/Artifacts/Artifacts_modify.jsp?
metadata_index=Artifacts1#%#Artifacts1#%#
#Artifacts&%
&/WOODSLIP_T/Artifacts/Artifacts_modify.jsp?
metadata_index=Artifacts2#%#Artifacts2#%#
#Artifacts</single>
</Artifacts1>

```

圖 15. XML 文件的內容

6. 與 DAAL 數位典藏系統的比較

本產生器所產生的後設資料管理系統與 DAAL 數位典藏系統的差異如下：

1. 產生器所產生的後設資料管理系統是參照 DAAL 數位典藏系統的架構(如圖 1)所建立的，後設資料維護、權限管理、代碼管理、異

動記錄的功能是相同的，但沒有中文缺字顯示、多媒體管理的功能，因為這些並不是每一個數位典藏系統都會需要的功能。而產生器增加檔案管理、表單欄位管理等功能，可以讓使用者自行維護他們的後設資料管理系統。

2. 產生器所產生的後設資料管理系統將後設資料以 XML 文件儲存，而 DAAL 數位典藏系統是以關聯式資料庫儲存後設資料。以 XML 文件儲存資料比關聯式資料庫儲存資料容易修改資料的結構。
3. DAAL 數位典藏系統是程式設計人員以人工的方式所製作的，而產生器則是將人工撰寫後設資料管理系統程式的工作，以自動化程式取代。我們對於 DAAL 的數位典藏系統的不足之處所做的改進，說明如下：
 1. 我們將每一個後設資料處理功能(即列表、查詢、新增、修改、刪除)的 JSP 網頁程式分成後端資料處理、後設資料顯示、流程控制三個部分，增加了程式的再利用性及維護性。
 2. 使用者可利用產生器自行建立所需要的後設資料管理系統，後設資料管理系統也提供使用者自行調整的功能，包含選擇列表項目、設定欄位等。
 3. 使用者可以調整後設資料綱要，不需要程式設計人員修改相關的程式。已儲存的後設資料 XML 文件亦無需變動。

7. 結論與後續研究

我們分析 DAAL 數位典藏系統的架構，並將分析的結果結合 MVC 與 XML 的技術，實作後設資料管理系統產生器，將程式設計者所做的大部份工作以自動化取代，以提升後設資料管理系統的建置速度。MVC 的觀念將後設資料管理系統中的維護後設資料功能，分成後端資料處理、後設資料顯示、流程控制三個部分，藉此降低程式的耦合度，並提升程式的模組化程度，增進程式的再利用性。在後設資料的儲存上，以 XML 文件儲存後設資料，其優點是當後設資料綱要變動時，可以彈性的處理；XML 文件的內容，不需調整即可繼續使用，讓使用者可容易的修改後設資料綱要。

在實作後設資料管理系統產生器的過程中，我們發現了一些可以進一步研究的議題：

1. 目前已有 MVC 為基礎的架構(architecture)，如 Apache Struts 等，若將這些架構運用至產生器所建立的後設資料管理系統，預期可以提高程式的再利用性與減少程式修改的困難度。
2. 未來可以加入更多有助於使用者建立系統的功能，如引導使用者操作的工具，以簡化後設資料管理系統的建立。
3. 本研究是以使用者所提供的 HTML 檔案作為基礎，建立後設資料管理系統。未來可以加入特定 HTML 樣版(template)，或協助使用者產生 HTML 檔案，讓使用者不需具備 HTML 檔案的撰寫能力亦可使用產生器。

參考文獻

- [1] 林宗德, "以 XML 為基礎之多版本後設資料管理—儲存、查詢與 Web 介面之動態產生", 國立雲林科技大學碩士論文, 雲林縣, 台灣, 2004.
- [2] 林彥君, 吳俊融, 王祥安, "數位典藏軟體設計規格書之建議標準: 五份軟體設計規格書之剖析", Proc. 第一屆軟體工程研討會, 台北市, 台灣, 2005, pp. 336-340.
- [3] 林彥君, 林泰安, 王祥安, "數位典藏異動管理模組剖析與建議", Proc. 第四屆數位典藏技術研討會, 台北市, 台灣, 2005, pp. 21-28.
- [4] 林彥君, 黃建中, 王祥安, "數位典藏軟體需求規格書之初期建議: 六份軟體需求規格書之剖析", Proc. 第一屆軟體工程研討會, 台北市, 台灣, 2005, pp. 330-335.
- [5] 黃國倫, 蕭人豪, 李家豪, 陳心渝, "數位典藏系統缺字處理及應用", Proc. 第三屆數位典藏技術研討會, 台北市, 台灣, 2004, pp. 79-85.
- [6] 趙晨希, "用 Struts 建立 MVC 應用的介紹", 2002. <http://www-128.ibm.com/developerworks/tw/library/l-struts-mvc/>
- [7] 數位典藏國家型科技計畫後設資料工作組, "後設資料系統工具", http://www.sinica.edu.tw/~metadata/tool/tool_chi.htm#metadatatool
- [8] C. Berkley, M. Jones, J. Bojilova, and D. Higgins, "Managing scientific metadata," IEEE Internet Computing, vol. 5, no. 5, pp. 59-68, 2001.
- [9] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, *Pattern-Oriented Software Architecture: A System of Patterns, Volume 1*. West Sussex: Wiley, 1996.
- [10] R. Englander, *Developing JAVA Beans*. Sebastopol: O'Reilly, 1997.
- [11] JavaZOOM, "jClientUpload Pro full package." <http://www.javazoom.net/applets/jclientupload/jclientupload.html>
- [12] Sun Microsystems, "org.w3c.dom (Java 2 Platform SE v1.4.2)." <http://java.sun.com/j2se/1.4.2/docs/api/org/w3c/dom/package-summary.html>
- [13] Sun Microsystems, "org.xml.sax (Java 2 Platform SE v1.4.2)." <http://java.sun.com/j2se/1.4.2/docs/api/org/xml/sax/package-summary.html>
- [14] G. Seshadri, "Understanding JavaServer Pages Model 2 architecture." <http://www.javaworld.com/javaworld/jw-12-1999/jw-12-ssj-jspmvc.html>
- [15] I. Singh, B. Stearns, M. Johnson, and E. Team, *Designing Enterprise Applications with the J2EE Platform, 2nd ed.*, Boston: Addison-Wesley, 2002.
- [16] Caucho Technology, "Caucho Resin: Fast, Open-Source Application Server." <http://www.caucho.com/index.xtp>