

以內容、版面和軟體元件建構數位博物館

俞旭昇，國立暨南國際大學資訊管理學系

王懿德，國立暨南國際大學資訊管理學系

摘要

數位博物館是將一般博物館所應具有的展示、收藏、教育、研究等功能以數位化的方式呈現。由於數位博物館具有多樣化的目標與功能，必須由許多不同專業的人員在共同的平台上一起完成，因此如何建構多角色合作式的資訊系統，讓參與者突破數位技術的障礙，便成為重要的研究問題。本研究將內容與版面元件化，並利用與 Web 平台無關的軟體元件呈現內容與版面結合後的展示效果，使得內容和版面提供者不需具備數位化技術能力，就可以共同合作完成數位博物館。本研究之內容元件與詮釋資料(Metadata)均存放於資料庫，可儲存多元素材和不同的註錄規格，達成數位博物館多元化的收藏與研究功能。我們也提出了十種軟體元件，每個元件再透過不同的動態網頁實作技術，可在不限定 Web 平台下，有效支援數位博物館的展示與教育需求。本系統以 Java 語言實作內容與版面管理系統，完成軟體元件的 ASP 實作部份，並已實際應用於大埔里的人文與自然數位博物館與一般企業的網站。

關鍵字：數位博物館(Digital Museum)、軟體元件(Software Component)、動態網頁(Dynamic Web Page)、內容管理(Content Management)

1. 緒論

國際博物館協會定義「博物館係一個非營利的常設機構，為服務社會及促進社會發展而為大眾開發，為研究、教育與娛樂為目的，從事收藏、維護、研究、傳播，並展覽有關人類及其環境之具體證物。」[2]。博物館具有保存人類文化、推廣知識、教育民眾、提供研究等重大功能，所以世界各國無不建立屬於自己的博物館，對人類文化的保存做出貢獻。

林育如[4]則定義數位博物館為：「一般博物館所應具有的展示、收藏、教育、研究等功能在數位博物館中皆以數位化的方式呈現」。隨著資訊科技的進步、網際網路的發達，數位博物館的內容可以透過網際網路傳播到全世界，大眾使用者

不受到時間、空間的限制隨時可以進入數位博物館中觀看，加速知識傳播，內容提供者也獲得知識分享的空間。

SCORM(Sharable Content Object Reference Model)是一個以 XML 為發展基礎的數位學習標準，此標準中定義了「Content Aggregation Model」及「Run-time Environment」兩個架構，前者為制定學習教材元件的標準，後者以 Javascript 溝通教學平台與教材[3]，簡單地來說，透過遵循共通的元件制定標準，可以達成網路學習及教材共享、交換的目的。

數位博物館計畫大體以國家型計畫為主，例如：美國回憶錄計畫(American Memory)、日本次世代數位典藏系統研究與發展專案及全球數位博物館計畫、國科會數位博物館計畫等，其重點為資訊的整合與標準的制定或運用，這些計畫多以國際或整個國家的典藏資料為著眼點來進行數位博物館的規劃[7]；在標準的制定及運用上，都柏林核心集(Dublin Core)、視覺資源核心類目(VRA Core Categories for Visual Resources)、CIDOC 資訊類目(CIDOC Information Categories)等皆為目前數位博物館計畫常用的幾項標準[8]。

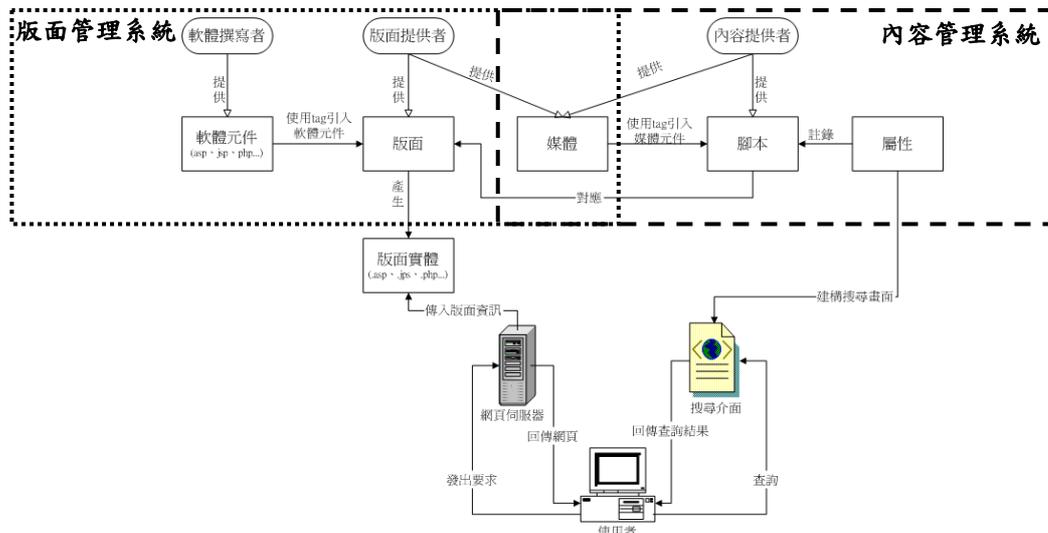
蘭嶼多樣化數位博物館主要針對多媒體呈現方式進行探討[5][6]。其中主要以樣板方式(Template-based)或區塊式超媒體文件模型(Block-based Hypermedia Document Model)製作文件，並以 XML 格式儲存，再配合 XSL 樣式而達到多重呈現的目的。

然而以 XML 文件為資料儲存的方式，如何有效搜尋與管理館藏為一大問題，目前雖有 XQL 標準，但相關實作尚處於發展階段，能否充分支援數位博物館為數龐大的館藏資料，仍需更多的研究與調查。其次以 XSL 樣式作為內容呈現的方法，使得 XSL 必須符合特定 XML 的 DTD，造成在呈現上缺乏彈性。再者，XSL 的產生需要具有程式撰寫能力的人支援，對於一般的館藏管理人員而言，系統使用的進入障礙顯然較高。

本研究則試圖從不同的觀點，提出一套內容與呈現分離且可彈性支援資料搜尋的系統架構，並突破參與者在數位技術上的障礙，不受伺服器平台語言的限制，使得數位知識得以更有效率的方式散佈與推廣。

2. 系統架構

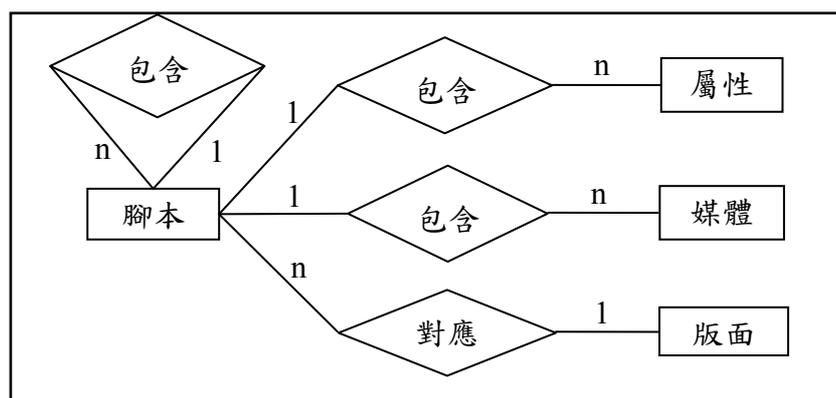
我們將數位內容分割為內容、版面和軟體三大類元件，分別由不同專長的人員製作，整體架構如圖一。



圖一 數位博物館系統架構

2.1. 內容管理系統

內容管理系統的資料模型，主要採用陳鴻嘉[9]所提出來的改良式 Web 超文字資料模式。該文將網頁資料分成階層式的文件、頁面與節點，並透過結構性鏈結、參考性鏈結與引入鏈結，將文件內的頁面、文件間的頁面以及頁面與節點連接起來，以方便編輯者或瀏覽者檢索 Web。我們考量文件觀念在瀏覽器上並沒有支援，因此將文件以及結構性連結刪除，並加入屬性概念，以容納不同標準的詮釋資料。我們以如圖二的關聯式資料庫儲存此資料模型。腳本可以包含其他的腳本，被包含的腳本稱為子腳本。版面及腳本間的關係留待 2.2 節中敘述。



圖二 數位博物館資料庫架構 E-R Model

2.2. 版面管理系統

在 WWW 的領域中，所謂版面指的是由各種 html 標籤、CSS(Cascading Style Sheets)組成的頁面。但靜態的 html 頁面並無法有效地呈現內容管理系統中所輸

入的腳本及媒體資料，必須利用描述語言(script language)撰寫動態網頁與資料庫互動，抓取館藏資料後再呈現於大眾眼前。由於撰寫描述語言需要相當的技術背景，對著重在畫面呈現的版面提供者而言，存在著相當的使用障礙。有鑑於此，本研究中將呈現腳本及媒體資料的這段邏輯，包裝成軟體元件，版面提供者只需引用適當的元件即可達到呈現館藏的目的。這樣的系統設計方式，不僅可以降低數位博物館的參與者在數位技術上的使用障礙，更可以達到內容與呈現完全分離的目標。版面提供者可先利用專業的網頁編輯工具設計版面，再嵌入適當的軟體元件後，儲存至資料庫。

2.3. 軟體元件

在一般的網頁開發流程中，版面設計者往往需根據內容提供者所提供的內容來製作呈現的版面，一旦內容有所更動，版面也需隨之修改，這意味版面與內容兩者之間具有高度的耦合性。為了達到內容與呈現分離的目標，本研究提出軟體元件的概念，將文字與媒體的排版，拆解為一個個的元件，當版面設計者希望以某種排版方式顯示文字性或多媒體的資料時，只需將適當的元件插入欲顯示的位置，完全不需要知道文字或媒體本身的資訊。

軟體元件有一定的介面和多個不同動態網頁的實作。版面設計者在提取元件置入版面時，元件是以標籤(tag)的形式存在，一旦經過存檔的步驟後，系統會自動抓取元件所對應的不同實作，而形成不同的版面實體。版面實體除了 html 標籤外，還可透過伺服器端的動態網頁程式語言與資料庫互動並抓取資料，將典藏內容呈現在大眾眼前。此架構讓數位內容的呈現不再受限於某種網頁伺服器，只要軟體元件提供不同的實作，再搭配適當的網頁伺服器系統即可運作。

本研究根據 Jun Han[1]所提出的通用軟體元件模型而將軟體元件的結構定義為下列幾層：

1. 參數層：參數指的是可由元件的使用者自行調整的部分，透過修改參數層中的參數，可以讓最後執行的結果更為符合使用者的需求。
2. 描述層：包含元件的名稱、作者、使用的語言及適用的網站伺服器，以方便版面設計者引用。
3. 程式碼層：軟體元件中實際運作的程式碼。

依據使用者訪談與對網頁的觀察，我們實作了十種如表一的軟體元件。在表

二的軟體元件範例中，描述層中 codeid 是元件名稱，author 是元件開發者的名字，language 則是說明本元件使用的程式語言，webServer 則是該語言適用的網頁伺服器；參數層中則包含八項參數，可依據需要修改影像高度、影像寬度等參數，使媒體能以最合適的大小呈現在畫面上；程式碼層包含了將媒體以直列方式呈現的邏輯，並將上述參數層的參數放入 html 標籤中適當的位置，讓畫面的呈現可以完全符合使用者的設定。

表一 系統中動態元件列表

元件名稱	說明
show_script_content	顯示此腳本內容
show_script_title	顯示腳本標題
list_childmedia_row_ml	列出子媒體，直列，並附有圖標題，圖及標題皆為超連結，點下超連結可觀看該圖的詳細資料
list_childscript_row	列出子腳本連結
show_content_with_media	同時顯示腳本與圖片內容，圖片嵌在內文中，圖片擺放方式為一左一右或一右一左
simple_search	顯示文字方塊與按鈕，使用者可以鍵入關鍵字搜尋資料庫中的資料
show_script_content_with_hyperlink	列出子腳本連結，子腳本嵌在內文中
show_media_content	同時顯示腳本與圖片，擺放方式為一張圖片配一段文字
show_childmedia_download	顯示子媒體，並可下載至瀏覽者的電腦中
show_childmedia_album	顯示子媒體，並以相簿集的方式呈現

表二 軟體元件結構

```

<!--列出子媒體_直列，圖片形式，媒體形成連結
<div:code codeid="list_childmedia_row_ml" author="guocy" language="vbscript" webServer="iis 5.0"
  td_width="100%" '儲存格寬度
  td_height="216" '儲存格高度
  td_background="624.gif" '儲存格背景圖片
  img_border="0" '影像框線
  img_height="134" '影像高度
  img_width="178" '影像高度
  img_alt="按圖有詳細說明(放大圖)" 'mouse 文字
  popurl="14" '圖片說明版面編號
-->
<%
  query="select mid, extension, title from ..."
  rs.Open query, DBConnection, adOpenStatic
  do until rs.eof
    media_title=rs("title")
  %>
<tr>
<td>
<a href="#" onclick="pop('<%=media_title%>', '<%=media_src%>')" target=_self><img class=imageAlign
src='../media/<%=media_src%>' width=<%=img_width%> border="0"></a>
  
```

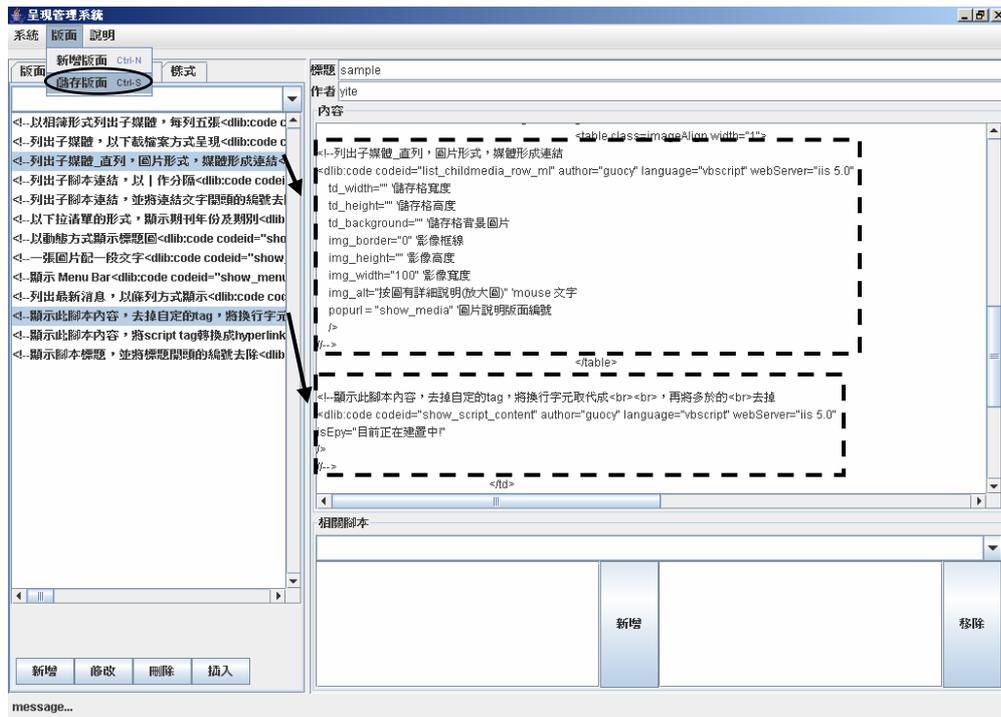
以下省略

The diagram illustrates the structure of a software component, divided into three distinct layers:

- 描述層 (Description Layer):** This layer is represented by a dashed-line box and contains the HTML structure for the component, including the opening and closing tags for the `div` and `td` elements, as well as the `img` tag with its attributes.
- 參數層 (Parameter Layer):** This layer is represented by a solid-line box and contains the parameter values for the component, such as `td_width="100%"`, `td_height="216"`, `td_background="624.gif"`, `img_border="0"`, `img_height="134"`, `img_width="178"`, `img_alt="按圖有詳細說明(放大圖)"`, and `popurl="14"`.
- 程式碼層 (Code Layer):** This layer is represented by a solid-line box and contains the code that generates the content, including the SQL query, the database connection, and the loop that iterates through the results to populate the `media_title` variable.

2.4. 版面與軟體元件之結合

以圖三名稱為 sample 版面為例，在箭頭所示的位置分別插入 list_childmedia_row_ml 及 show_script_content 兩元件，選擇「儲存版面」之後，該版面即產生如表三的版面實體 sample.asp 檔，以此版面實體顯示某腳本的結果如圖四。



圖三 插入軟體元件

表三 sample.asp 程式碼部分節錄

```

.....
<tr>
  <td align="left" bgcolor="#F8F7F2">
    <table class=imageAlign width="1">
<!--列出子媒體_直列，圖片形式，媒體形成連結 //-->
<%
  td_width="" '儲存格寬度
  td_height="" '儲存格高度
  .....
%>
.....

<!--顯示此腳本內容，去掉自定的 tag，將換行字元取代成<br><br>，再將多於的<br>去掉 //-->
<%
sid = Request("script")

Set rs = Server.CreateObject("ADODB.Recordset")
query = "select sid, title, content from script where sid = " & sid
rs.Open query, DBConnection, adOpenStatic

content = rs("content")
%>
.....
</tr>
.....

```



圖四 sample.asp 的執行結果

2.5. 後端的資料與前端使用者的互動

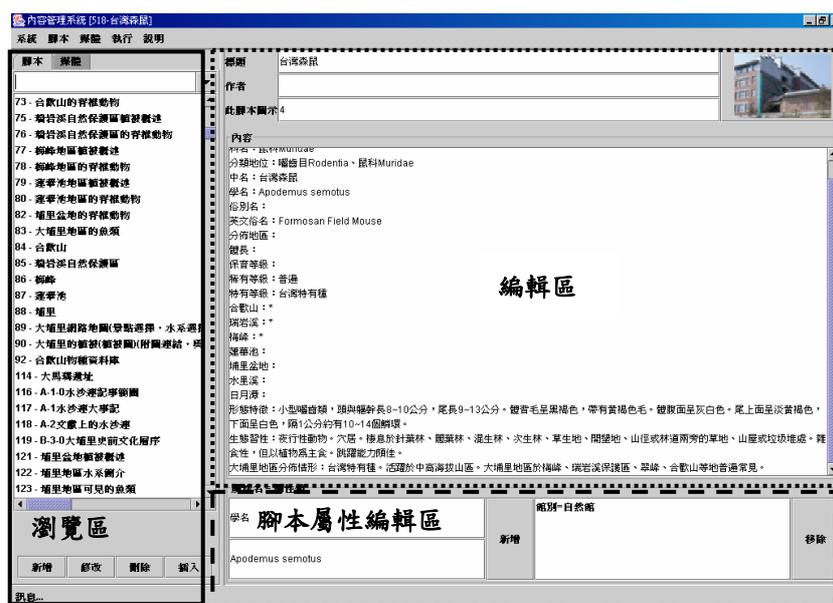
當使用者發出觀看某一個腳本內容的請求時，實際上是呼叫某個版面實體，並給予適當的參數，由版面實體到資料庫中抓取相關資料並呈現在使用者眼前。本研究之架構還可以提供客製化的搜尋機制，以方便使用者查詢館藏。由於數位博物館的館藏量相當龐大，如何提供精準且有效率的查詢方式是一項重要課題。2.1 節裡的屬性可以為腳本增加不限個數的 metadata，這些 metadata 可以進一步用在搜尋機制上，提供更精準的搜尋結果。舉例來說，當內容提供者新增某一物種，此時在屬性名的設定上可以設為「學名」，屬性值的部分則填入該物種的名稱。在前端的搜尋頁面上，我們可以設計出以學名來搜尋的介面，因此在設定完該物種的屬性後，前端頁面即可立即顯示出該物種的學名，若使用者點選該學名，便可查詢到新增的物種資料。

3. 系統實作

本研究在內容和版面管理系統採用三層式架構(3-tier)，整個前端系統是以 Java 寫成，透過 Java Application Server 與資料庫連結。此外我們採用 Java Web Start[10]的技術，以利這些人員從網頁上直接啟動 Java Swing 介面。資料庫部分採用微軟的 SQL Server 2000。

3.1. 內容管理系統之實作

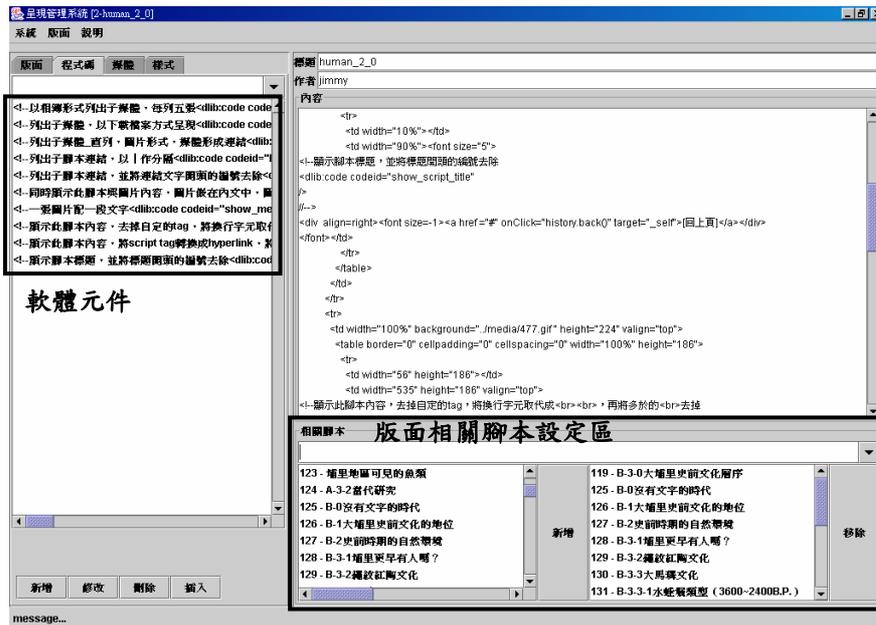
主畫面分為左邊的「瀏覽區」和右邊的「編輯區」，如圖五。「瀏覽區」能瀏覽資料庫所存放的資料，如。「編輯區」則讓使用者編輯此腳本內容。內容提供者可在腳本屬性編輯區新增、修改、移除此腳本的屬性。屬性由一組屬性名及屬性值組合而成，腳本的屬性沒有個數限制。



圖五 內容管理系統主畫面

3.2. 版面管理系統之實作

主畫面與內容管理系統相同，也分為瀏覽區與編輯區，如圖六。瀏覽區提供了讓版面編輯者瀏覽的功能，此區可以新增、查詢、修改及刪除版面。版面編輯時會用到系統中的軟體元件，可藉由瀏覽區插入的功能，將元件置入適當的位置，存檔時會解析這些元件，將之轉換為對應的動態網頁。版面相關腳本設定區可以設定那些腳本是利用此一版面呈現，完成設定後會將腳本與版面的對應關係存入資料庫。



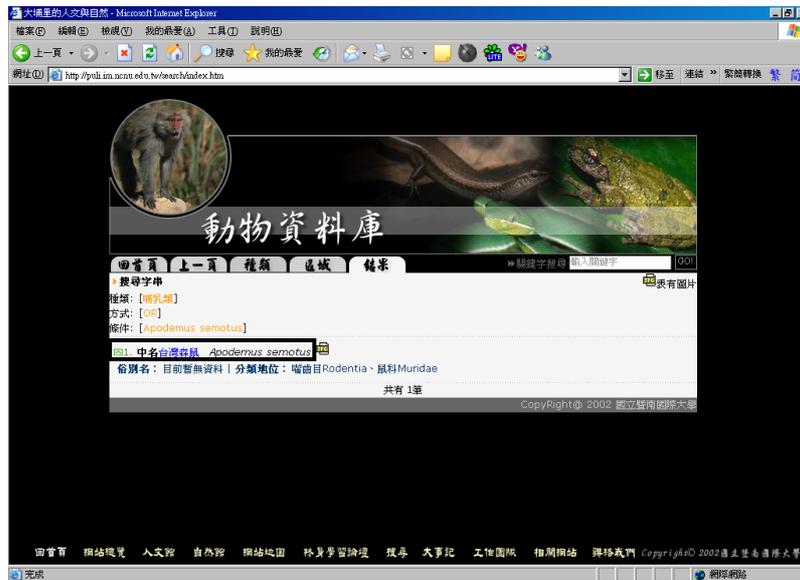
圖六 版面管理系統主畫面

3.3. 網頁搜尋機制之實作

以下的例子說明，內容提供者欲新增某一物種，而該物種之學名為 *Apodemus semotus*，此時在屬性名的設定上可以設為「學名」，屬性值的部分則填入「*Apodemus semotus*」，如圖五。在前端的搜尋頁面上，我們可以設計出如圖七以學名來搜尋的介面，在設定完圖五的物種屬性後，圖七的查詢選項即可立即顯示出該物種的學名。最後的搜尋結果如圖八。



圖七 以學名為查詢條件的搜尋畫面



圖八 以學名為條件的搜尋結果

4. 實例應用

本章節中將列出兩個實際案例，說明本研究所提的系統架構應用在真實網站上的成果。

4.1. 大埔里的人文與自然數位博物館

大埔里的人文與自然數位博物館為國科會數位典藏計畫之一，該計畫是由國立暨南國際大學成人與繼續教育研究所及資訊管理學系、中央研究院史語所及國立自然科學博物館合作。自 2002 年計畫開始至今日為止，目前內容管理系統中共有 907 篇腳本、783 個媒體檔，版面管理系統則有 51 種版面、使用到 7 種軟體元件。實際執行成果，可至 <http://puli.im.ncnu.edu.tw> 參觀。

4.2. 埔里基督教醫院全球資訊網

埔里基督教醫院全球資訊網為該醫院資訊室與國立暨南國際大學資訊管理學系合作建置。該計畫自 2004 年 10 月開始至今，目前仍持續進行網站內容的建置，因此在腳本及媒體數量方面尚無法預估，不過現今版面管理系統中已有 23 種版面，其中共使用到 10 種軟體元件。欲參觀該網站請蒞臨下述網址：<http://www.pch.org.tw>。

在本計畫中，國立暨南國際大學資訊管理學系擔任的角色為軟體元件的開

發、系統導入以及版面設計的工作，雖然本計畫至今仍在持續進行中，但是系統建置與導入方面的工作已經完成，總計從計畫起始至系統導入完成約需一個半月的時間，而埔基方面的人員從接觸新系統到完全上手的時間，若以每週 5 小時計算，估計約需 4 週共 20 小時。整體來看，本研究所提出的系統架構對於沒有技術背景的人來說在學習與使用上，並沒有太多的技術進入障礙，此點相當符合本研究當初的設計精神，未來應可繼續在各領域的網站內容建置上加以推廣與應用。

5. 結論與未來研究方向

綜觀來看，在我們的系統架構下所設計出來的數位博物館，具有以下貢獻：

1. 腳本、版面、媒體元件可重覆使用。
2. 資料輸入的彈性大幅提升。透過屬性名、屬性值的設定，內容提供者能不限個數為數位博物館的素材增加 metadata，不必受限於資訊人員所設計的資料庫欄位。
3. 軟體元件的設計理念，可以讓版面設計者更快速地製作版面，同時藉由豐富元件庫中的元件，可以讓版面呈現更多元，不必擔心版面流於同一樣式。
4. 數位典藏內容的呈現不受特定網頁伺服器的限制，網頁的呈現可隨時以不同的描述語言搭配適當的網頁伺服器來運作。
5. 本系統運用三層式架構，讓內容提供者或版面設計者不論身在何處，只要有網路就可以使用系統完成任務，大幅提升了系統的可達性。

本研究中對於數位博物館的內容管理機制提出一套解決方案，然而在整體系統架構上仍有未盡完善的地方，可作為未來改進或繼續研究的方向：

1. 本研究中未來應加入權限控管的部份，讓系統中各種素材的編輯、查詢及引用受到更嚴謹的保護，同時也可避免具機密性的文件在未授權的情況下在網路上散佈。
2. 本研究在軟體元件部份未來仍可持續擴充，繼續開發更多元的排版組合，讓畫面的呈現更為豐富。
3. 本研究所提的系統架構目前較難針對 html 中的某些元素加以拆解而獨立成一個軟體元件，例如要在網頁中以表格的形式呈現資料，就無法利用軟體元

件的方式來達成，因為在 html 的結構中，表格(<table>)是由多重的列(<tr>)與欄(<td>)組合而成，有時表格中又包含多個表格，由於結構複雜，很難結構化後形成一個元件。因此未來可以針對此類較複雜的 html 結構提出一套解決方案。

6. 參考文獻

1. Jun Han, A Comprehensive Interface Definition Framework for Software Components, Software Engineering Conference, 1998, pp.110-117
2. 王嵩山，過去與未來·數位博物館中的人類學空間，2000，台北：稻香出版社
3. 王學誠、涂文祥、游文淮、陳俊杉、謝尚賢，『導入網路教學共享機制之探討與實做』，國立台灣大學台大工程學刊，2002年6月，第八十五期，pp.59-68
4. 國科會數位博物館專案推廣教育計畫小組，數位博物館任遨遊，科學月刊，1999，第二十九卷，第五期
5. 張惠閔，以 XML 為基礎之數位博物館資料呈現設計及網頁特效管理系統，國立暨南國際大學資訊管理研究所碩士論文，2001
6. 林金龍，區塊式數位博物館超媒體 XML 展示文件及管理系統設計，國立暨南國際大學資訊管理研究所碩士論文，2001
7. 陳百薰，以 XML 為基礎之數位博物館展示資料模型與內容管理，國立暨南國際大學資訊管理研究所碩士論文，2001
8. 陳亞寧、陳淑君，『Metadata 在數位博物館之發展與分析』，圖書館學與資訊科學，2001年10月，第二十七卷，第二期，pp.52-66
9. 陳鴻嘉、許秉瑜、陳彥良，『WWW 資料庫設計之研究』，資訊管理學報，1999年1月，第五卷，第二期，pp.33-54
10. Java Web Start Technology: <http://java.sun.com/products/javawebstart/>