以 XML 資料庫系統建置後設資料儲存庫

王祥安
中央研究院資訊所
台灣科技大學資訊工程系
sawang@iis.sinica.edu.tw

黃建中
台灣科技大學資訊工程系
M9215052@mail.ntust.edu.tw

林彥君
台灣科技大學資訊工程系
yclin@et.ntust.edu.tw

摘要

數位典藏系統的發展過程中，後設資料(metadata)規格的分析與製作，是很重要的一環。我們以 XML 文件儲存後設資料規格，並以原生型 XML 資料庫系統(native XML database system)儲存 XML 文件，實作出後設資料儲存庫系統，並實際將數位典藏國家型科技計畫中所產生的數份後設資料需求規格書存入系統中，提供使用者查詢各計畫的後設資料規格。系統亦提供讓使用者自行創造、儲存、分享後設資料規格之功能。本系統提供數位典藏後設資料規格的查詢、製作及儲存環境，期望未來進一步成為後設資料分析與製作的知識入口。

關鍵字

XML、資料庫系統、後設資料儲存庫、數位典藏

1. 前言

數位典藏是將重要的典藏文物數位化保存與管理。配合現今的網路環境，數位典藏文物資料可以方便地呈現與使用。在數位典藏系統的建置過程中，典藏品的後設資料(metadata)分析是很重要的一環。其分析結果會影響到整個典藏資料的完整性，並關係到整個典藏系統的建置、資料的著錄、檢索精確度、後續的資料交換與加值利用等。由於後設資料的重要性，在建立數位典藏系統過程中，後設資料的分析是在初期就必需被定義的。

製作一份後設資料規格書需花費大量的時間與人力。目前後設資料的分析與制定，多數由後設資料分析人員與典藏內容提供者共同討論資料需求，並比對過去相關計畫與國際標準的後設資料，最後再製作成後設資料規格書。在此分析、比對、製作的過程中，多以人工處理的方式，並沒有太多的軟體工具可以輔助。而產生的後設資料規格書，一般多以紙本或電子檔案的方式呈現，對於查詢與再利用上會受到限制。

XML 是由 World Wide Web Consortium (W3C)所制定。由於 XML 具有可自定標籤(tag)、可擴充性、易於程式處理及可彈性地改變文件結構與內容的特性，因此常被用來呈現結構複雜的資料。數位典藏領域的後設資料具有項目眾多、結構複雜、需求異動性大的特質。因此，XML 常被用來表示後設資料，如應用於數位典藏的 Dublin Core [5]及定義學習物件(learning object)的 IEEE Learning Object Metadata [6]等後設資料標準，皆訂出以 XML 呈現後設資料的規範。由於以 XML 呈現後設資料具有前述眾多優點[9]，因此，我們以 XML 來呈現後設資料規格。

我們的主要工作在提供一個後設資料儲存庫系統，儲存過去已發展的後設資料規格及國際的後設資料標準。它可以提供典藏單位在後設資料分析時的查詢與使用，並且快速地建立後設資料規格。預期可減少後設資料規格書之製作時間與人力，並提升整個數位典藏系統的建置速度。

本文第 2 節介紹不同類型的資料庫系統如何儲存 XML 文件。第 3 節說明後設資料儲存庫系統架構及如何以 XML 呈現後設資料規格。第 4 節說明軟體實作。第 5 節為結論與未來方向。

2. 儲存 XML 文件之資料庫

目前用來儲存 XML 文件的資料庫系統，大體上可以分為兩大類，一種是以關聯式資料庫系統儲存，另一種是以原生型 XML 資料庫系統(native XML database system)儲存，分別說明於下。

2.1. 以關聯式資料庫儲存

以關聯式資料庫系統儲存 XML 文件的方式可區分為兩種[8][10]。第一種是透過事先定義好的對應規則，將 XML 文件拆解成多個部份存放在不同的表格(table)。當要取回原本的 XML 文件時，則透過 SQL 查詢語言取得各表格的資料，再組合成 XML 文件。此方式的優點是可快速的處理結構固定的 XML 文件，並可直接套用在關聯式資料庫系統之上。但當 XML 文件結構很複雜時，此方法將產生大量的表格。若要將分散在各表格中的資料組合成完整的 XML 文件，必需透過複雜的資料結合(join)才能達成。此外，一旦對應規則訂定後，若因資料需求的改變而造成 XML 文件結構需要更動時，則可能需重新定義對應規則。因此，在處理不同結構的 XML 文件時較無彈性。

第二種方式是將 XML 文件以文字或二進位(binary)資料的格式存入關聯式資料庫系統表格的一個大型欄位(large object，LOB)中，在存取 XML 文件時只須對此欄位進行資料的輸入、輸出即可。此方式的優點是設計簡

單,不需複雜的表格拆解動作即可處理結構複雜的 XML 文件，適合應用於以整份 XML 文件作為輸入、輸出的應用程式，亦可直接套用在關聯式資料庫系統之上。而此方式的缺點是當僅需處理部份特定元素(element)之資料時，仍需輸出整份 XML 文件，再進行資料剖析(parsing)，才能正確地處理 XML 文件。針對此缺點，越來越多的關聯式資料庫系統加入 XML 文件的處理機制以改進此缺點。

2.2. 以原生型 XML 資料庫儲存

近年來興起一種專為儲存 XML 文件的原生型 XML 資料庫。它是以 XML 文件為基本儲存單位，儲存時不需將文件拆解而儲存於多個表格，或預先設定儲存欄位的資料型態。因此，可以容易的存取整份 XML 文件，並且不會改變其原本的資料結構與內容[2]。以 eXist 原生型 XML 資料庫系統為例[4]，除上述優點外，並提供 XQuery [12]、全文檢索(full-text searching)、自動索引(automatic indexing)、Java 程式語言整合(Java binding)等機制，可容易的進行程式的開發與應用。

由於數位典藏領域之後設資料相當複雜，且資料需求經常改變，因此，以整份 XML 文件為一個基本的儲存單位，對於更改後設資料規格的架構及內容有較高的彈性。而 XML 資料庫系統對 XML 文件的處理機制較完整,且有免費的 eXist 可以使用,所以我們以原生型 XML 資料庫系統儲存 XML 文件。

3. 後設資料儲存庫系統架構

我們的後設資料儲存庫系統架構(見圖 1),採用三層式架構，分別為 XML 資料庫系統、應用程式伺服器和使用者瀏覽器。XML 資料庫系統扮演著資料儲存庫的角色，主要是儲存過去所有計畫所採用的後設資料規格、國際的後設資料標準及使用者自行建立的後設資料規格。應用程式伺服器扮演著與 XML 資料庫系統及使用者瀏覽器的溝通角色；系統的主要功能即是在此應用程式伺服器上運作。使用者瀏覽器則是使用者的操作界面。透過三層式架構的設計，使用者無需特別安裝應用程式，即可以瀏覽器操作系統上所有的功能。

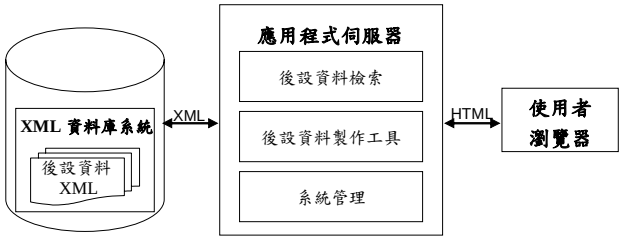

圖 1　後設資料儲存庫系統架構圖

本系統提供了三個主要功能：

(1) 後設資料檢索。本功能提供使用者以後設資料的項目名稱、英文名稱、資料型態、欄位大小等項目進行關鍵字的檢索，並以 XQuery 的語法查詢儲存於 XML 資料庫系統中的後設資料規格。

(2) 後設資料製作工具。本功能提供使用者以一般的瀏覽器，依自己的需求直接在線上建立全新的後設資料規格，或修改已存在於後設資料儲存庫系統中的後設資料規格。使用者完成製作的後設資料規格，可存入後設資料儲存庫系統中，亦可將此資料儲存在自己的電腦上，以提供後續後設資料的維護，及系統分析時之參考。

(3) 系統管理。將根據使用者的身分對其所能執行的功能加以控管；不同權限的使用者可使用的功能不同。透過此機制來防止未經授權的人進入系統，避免不當的使用、竄改或破壞資料。

在呈現後設資料的 XML 文件架構設計上，主要是參考中研院 MAAT 小組所製作的後設資料需求規格表之項目，以漢代簡牘後設資料需求規格書[1]為例，表 1 為需求欄位建置表部份內容，其中，英文名稱欄位用來說明後設資料各元素的英文名稱。資料型態欄位用來說明著錄資料時的資料型態，如 Varchar 表示該元素在典藏系統中是以文字型態的方式來輸入資料。大小欄位用來說明該元素在典藏系統資料庫中所需之最大空間，以字元為單位。

表 1　漢代簡牘需求欄位建置表部份內容

項目名稱		英文名稱		資料型態	大小
基本資料					
識別資料	遺物登錄號	Identifier	Registered Number	Varchar	30
	簡號		Pek. Coll.	Varchar	200

表 2 為需求欄位屬性表部份內容，其中，必填欄位標示「＊」，表示著錄資料時該元素之值不能為空值。屬性欄位標示「唯一」，表示該元素的值在資料庫中是唯一存在的。提供者欄位說明著錄該元素資料時是由典藏系統自動產生或由填表者自行填入。

表 2　漢代簡牘需求欄位屬性表部份內容

項目名稱		必填	多值		屬性	提供者
基本資料						
識別資料	遺物登錄號	＊			唯一	填表者
	簡號				唯一	填表者

表 3 為後設資料標準比對表部份內容，其目的是記錄各元素與國際標準(如 Dublin Core)的對應情況，如「遺物登錄號」是對應於 Dublin Core 所規範的 Identifier。

表 3　漢代簡牘後設資料標準比對表部份內容

項目名稱		Dublin Core Elements
基本資料		
識別資料	遺物登錄號	Identifier
	簡號	Identifier

我們將上述三個表格的內容以 XML 呈現，如圖 2 所示。在主題的 element 中，包含一個 name 的屬性(attribute)，其值為「漢代簡牘」，它用來描述後設資料規格的計畫名稱。接著根據前述的表格內容，以 element 代表後設資料規格中的元素(如識別資料)，並以 attribute 表示各元素之描述項目(如英文名稱、資料型態等)。

```
<?xml version="1.0" encoding="BIG5" ?>
- <主題 name="漢代簡牘">
  - <基本資料>
    - <識別資料 英文名稱="Identifier" 必填="*" 多值="  ">
        <遺物登錄號 英文名稱="Registered Number" 資料型態="Varchar" 大小="30" 必填="  "
          多值="  " 屬性="唯一" 提供者="填表者" DC-Element="Identifier">  </遺物登錄號>
        <簡號 英文名稱="Pek. Coll." 資料型態="Varchar" 大小="200" 必填="  " 多值="  " 屬性
          ="唯一" 提供者="填表者" DC-Element="Identifier">  </簡號>
      </識別資料>
    </基本資料>
  </主題>
```

圖 2　以 XML 呈現漢代簡牘後設資料規格

此外，我們參考將 MAAT 小組對 Dublin Core [5]、CDWA [3]等國際上著名的後設資料標準之翻譯，將其內容以 XML 格式呈現。由於 Dublin Core 僅有規範英文名稱的資訊，因此，如圖 3 所示，在呈現 Dublin Core 基本元素的 XML 文件中，僅有英文名稱的 attribute，而沒有圖 2 中資料型態、大小、多值等 attribute。這樣的設計可避免 XML 文件中包含不必要的內容描述，以減少 XML 文件之儲存空間。若未來 element 的 attribute 項目增加，系統亦可彈性的擴充，並可避免改變其它已存在於系統中的 XML 文件。

```
<?xml version="1.0" encoding="big5" ?>
- <計畫名稱 name="DC">
  - <DC元素>
      <標題 英文名稱="Title" />
      <著作者 英文名稱="Creator" />
      <主題-關鍵字 英文名稱="Subject and Keywords" />
      <描述 英文名稱="Description" />
      <出版者 英文名稱="Publisher" />
      <貢獻者 英文名稱="Contributor" />
      <日期 英文名稱="Date" />
      <資料類型 英文名稱="Resource Type" />
      <格式 英文名稱="Format" />
      <資料識別 英文名稱="Resource Identifier" />
      <來源 英文名稱="Source" />
      <語言 英文名稱="Language" />
      <關連 英文名稱="Relation" />
      <範圍 英文名稱="Coverage" />
      <管理權 英文名稱="Rights Management" />
    </DC元素>
  </計畫名稱>
```

圖 3　以 XML 呈現 Dublin Core 的基本元素

4. 軟體實作

我們的系統，在 XML 資料庫系統端，採用的是 eXist [4]。在應用程式伺服器端，採用的是 Apache Tomcat。系統程式採用 Java SDK 1.4.2_03 及 JavaServer Pages (JSP)程式語言開發，資料庫查詢語言採用 XQuery。在使用者端採用 Microsoft Internet Explore 及 Mozilla Firefox 進行測試。在作業系統上皆採用 Microsoft Windows XP 作業系統。

後設資料儲存庫系統中所儲存的 XML 文件，並不適合於使用者直接閱讀，因此需將其轉換成 HTML 之表格型式呈現，以利於使用者了解。負責將後設資料 XML 內容轉換成 HTML 的處理程式，是以 Java API for XML Processing (JAXP)之 DOM API [7]撰寫。此程式會計算出 XML 文件中每一個 element 的階層數及子 element 個數，並逐一抓出各 element 的名稱及 attribute 的值以產生 HTML。以圖 2 的漢代簡牘後設資料規格 XML 文件為例，透過此程式會轉換成如圖 4 的表格。

項目名稱		英文名稱	資料型態	大小	必填	多值	屬性	提供者
基本資料								
識別資料	遺物登錄號	Identifier	Registered Number	Varchar	30	*		唯一	填表者
	簡號		Pek. Coll.	Varchar	200			唯一	填表者

圖 4　將 XML 轉換成表格呈現

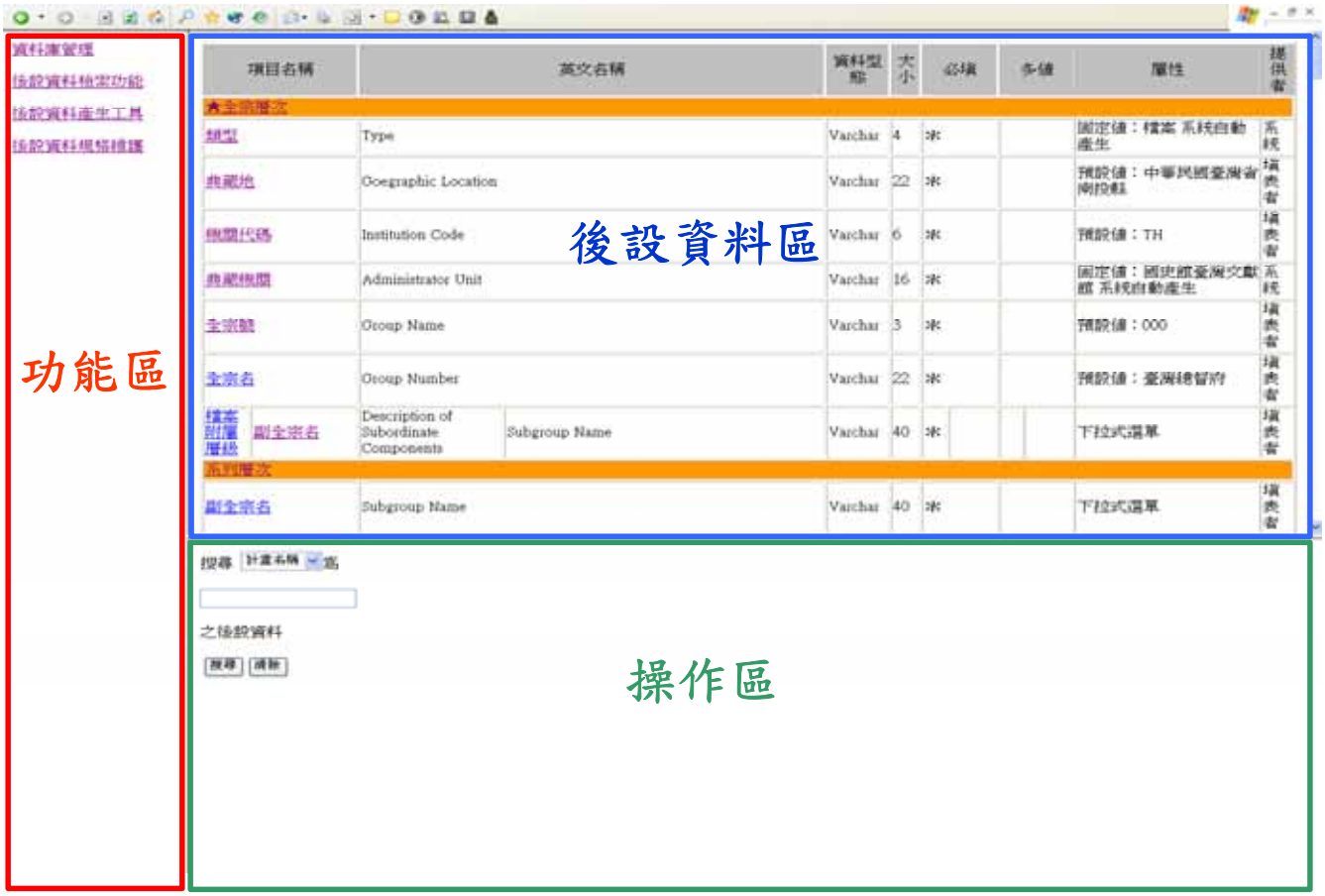

圖 5　後設資料儲存庫系統畫面

後設資料儲存庫系統畫面如圖 5 所示，主要可分成三個區域。在系統畫面的左方是功能區，其顯示使用者可操作的系統功能，如後設資料檢索功能等，使用者可透過點選各功能的連結來進行操作。功能區的右上方為後設資料區，用來呈現檢索功能所找到的後設資料規格，或呈現後設資料製作工具所產生的後設資料規格。功能區的右下方為操作區，此區是讓使用者操作系統各功能時輸入應提供的資訊，將根據不同功能呈現不同的處理介面。例如使用者點選功能區的後設資料檢索功能後，在操作區會出現要求使用者輸入關鍵字的介面，以檢索系統中的後設資料規格。

後設資料儲存庫系統之初始資料來源，是以人工方式將中研院 MAAT 小組所產生的後設資料需求規格書之部份內容及國際上著名之後設資料標準，如 Dublin Core、CDWA 等，將其轉換成 XML 格式匯入後設資料儲存庫系統中，以測試系統之處理能力。

我們的後設資料製作工具是後設資料儲存庫系統的核心，它可以用來製作新的後設資料規格或是維護已存在於系統中的後設資料規格，其運作流程如圖 6 所示。使用者登入系統後，需先選擇要製作新的或修改舊的後設資料規格，若要製作新的後設資料規格，使用者須先輸入計畫名稱、元素名稱等內容來產生根元素(基本元素)。若要維護存在於系統中的後設資料規格，須先從系統中選取被授權操作的後設資料規格。接著需選取目前所要操作的後設資料元素；如圖 7 所示，所選取的元素為遺物登錄號，在該元素上會以「★」表示，並會出現新增、修改、刪除之按鈕。使用者可新增此元素的子元素，或修改、刪除該元素之資料。以修改為例，按下修改按鈕後，在操作區將出現如圖 8 之操作介面，使用者可修改各欄位的內容。而新增元素的方式有兩種：

1. 使用者可透過輸入新元素的項目名稱、英文名稱等內容來產生新元素。
2. 使用者查詢已存在於系統中的後設資料規格，在操作區將出現如圖 9 之檢索結果，使用者可透過點選「加入」按鈕來產生新元素。

當後設資料規格建置完成後，使用者可將此資料儲存至後設資料儲存庫系統中，以供後續之利用。

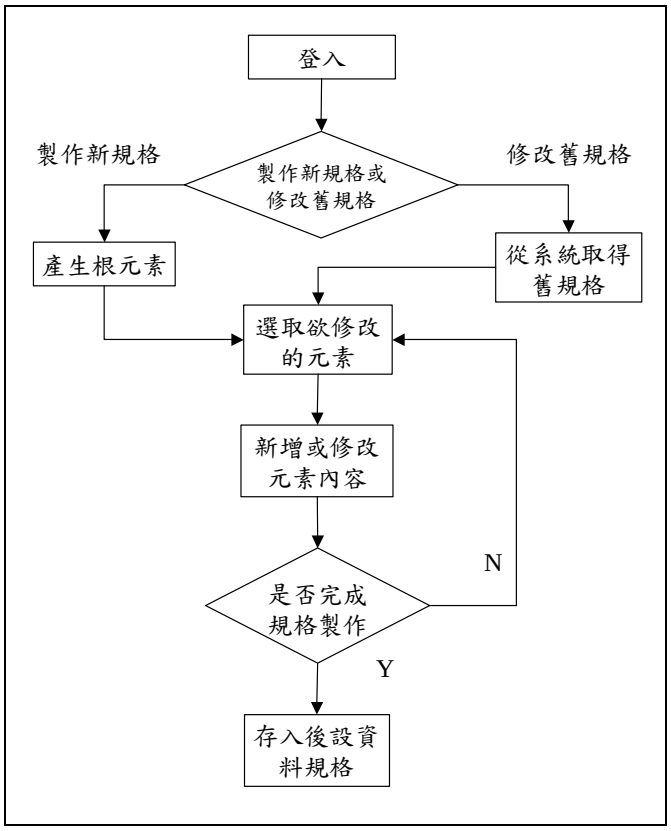

圖 6　後設資料製作工具運作流程

圖 7　後設資料建置畫面

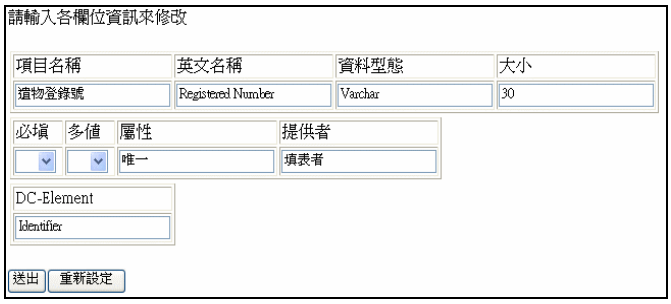

圖 8　修改元素內容之操作介面

圖 9　以檢索結果新增元素之操作介面

5. 結論與未來發展

數位典藏系統的建置過程中，後設資料規格的分析與製作很重要卻費時費力。我們提出了一個以 XML 文件儲存後設資料規格之方式，並使用原生型 XML 資料庫系統做為儲存庫以儲存 XML 文件。我們已將多個數位典藏計畫的後設資料規格及國際的後設資料規範，以人工方式轉換成 XML 文件，並儲存於後設資料儲存庫系統中，可提供使用者檢索不同計畫的後設資料，透過本系統之後設資料製作工具，讓一般使用者或後設資料分析人員，在系統的協助下，建立一個新的後設資料規格。

在本研究的進行過程中，發現了一些未來值得繼續探討的方向：

(1) XML 呈現方式的轉換。目前將 XML 文件呈現在瀏覽器上，雖然可直接顯示，但仍不易於人之閱讀，因此需要將其轉換成 HTML，才能在瀏覽器上完整的呈現表格、顏色、圖型等，然而將 XML 文件轉換成 HTML 的處理過程，仍然需要繁複的程式來進行。目前 W3C 已提出 XForms 1.0 recommendation，它用 XML 來指示文件之呈現方式[11]。透過支援 XForms 的瀏覽器，可直接呈現出文件格式。雖然 XForms 目前仍未被所有的瀏覽器支援，但已有越來越多的軟體廠商陸續的加入支援 XForms 的行列。未來若 XForms 正式成為標準，將可大幅降低呈現 XML 資料的繁複工作。

(2) 以 XML 為基礎之數位典藏系統。目前數位典藏系統大多是以關聯式資料庫系統為儲存實體。在此架構下，當後設資料需求改變時，資料結構及系統程式都需要隨之變更，耗費大量的人力與時間維護系統。若將典藏資料以 XML 文件方式儲存，將可增加資料結構的彈性，對於後續資料的利用，亦有較佳的再利用性。

(3) 後設資料建議機制。目前後設資料規格的製作，是以人工方式逐一比對國際標準的或相同領域所發展的後設資料規格，以制定出自己的後設資料規格。未來，當後設資料儲存庫系統存放大量的後設資料

規格後，可加入智慧分析之能力，自動分析後設資料規格之各元素所具有的特性及關聯性，並提出後設資料元素之建議，提供使用者製作後設資料規格之參考。系統可根據使用者回應之資訊，不斷提升建議系統之智慧能力。

(4) 後設資料儲存庫效能評估。目前系統中尚未儲存大量的後設資料規格，無法測出整個系統的效能瓶頸。未來將有更多的後設資料規格存入系統之中，可以用來測試 XML 資料庫系統的能力是否足以應付多使用者操作、大量資料儲存與運算之需求。

本研究成果可以讓數位典藏領域的不同計畫與機構，透過網路分享各自的後設資料規格，讓後續欲建立數位典藏系統的典藏單位，在分析與建立後設資料規格時，有一個集中的知識入口，更容易的查詢到相關的資料，使得後設資料的分析更加的周詳及迅速，進而讓整個典藏系統的建置更完整。

誌謝

本研究計畫部份經費由行政院國家科學委員會補助，計畫編號：NSC 93-2422-H-011-001、NSC 94-2422-H-011-001。

參考文獻

[1] 數位典藏國家型科技計畫後設資料工作組，中央研究院歷史語言研究所拓片與古文書數位典藏計畫漢代簡牘後設資料需求規格書 version 1.0，http://www.sinica.edu.tw/~metadata/project/filebox/stone-HangJan/stone_HangJan_spec_v1-0.pdf.

[2] Bourret, XML and Databases, 2004, http://www.rpbourret.com/xml/XMLAndDatabases.htm.

[3] Categories for the Description of Works of Art (CDWA), http://www.getty.edu/research/conducting_research/standards/cdwa/.

[4] A. B. Chaudhri, A. Rashid, and R. Zicari, XML Data Management: Native XML and XML-Enabled Database Systems, Addison-Wesley, Boston, MA, 2003.

[5] Dublin Core Metadata Initiative, http://dublincore.org/.

[6] IEEE Standard for Learning Object Metadata, IEEE Std 1484.12.1, 2002.

[7] Java API for XML Processing (JAXP), http://java.sun.com/xml/jaxp/index.jsp.

[8] H. Katz, et al., XQuery from the Experts: A Guide to the W3C XML Query Language, Addison-Wesley, Boston, MA, 2003.

[9] D. Marco, Building and Managing the Meta Data Repository: A Full Lifecycle Guide, Wiley, New York, NY, 2000.

[10] S. Natu and J. Mendonca, Digital asset management using a native XML database implementation, Proc. 4th Conf. on Information Technology Curriculum, Indiana, USA, pp. 237-241, Oct. 2003.

[11] T. V. Raman, XForms: XML Powered Web Forms, Addison-Wesley, Boston, MA, 2003.

[12] W3C, XQuery 1.0: An XML Query Language, http://www.w3.org/TR/xquery/.