

應用 XML Schema 架構之 Metadata 管理系統

余顯強

摘要

本研究使用 XML Schema 來建構 XML 框架 (framework), 透過內部元素的完整階層樹架構, 解決異質 Metadata 之間儲存與管理的相關問題, 分析建置能夠處理多種 Metadata 資料, 並能統籌檢索與交互權威控制的系統架構。

一、前言

在資訊科技快速變遷的網際網路時代, 組織內部舊有的資訊系統面臨著需要重新設計及整合的命運, 以應付外在各式各樣的變動及挑戰。在這樣的情況下, 組織需要一種新的資訊運作模式, 以有效降低系統發展的時間及成本; 同時, 藉由系統本身的自動化處理流程, 以儘量減少人力不必要的介入。但是使用者有不同層級之資訊需求, 如: 檢索需求、管理需求、保存需求等。不同的資訊亦有不同的文獻特質, 如: 地理資訊、博物館資訊、科技文獻、文學創作、音樂作品等, 均各有其不同之文獻結構、媒體特性、使用習性。資訊組織的對象、範圍過去多是以印刷或視聽媒體為主, 然而, 隨著電子出版品、使用者介面與資訊媒體

關鍵詞 (Keywords): 文件型別定義; 詮釋資料; 統一化模型語言; 系統設計

Metadata; XML Schema; DTD; Unified Modeling Language (UML);
System Design

余顯強: 國立交通大學資訊管理研究所博士班、輔仁大學圖書資訊學系兼任講師; E-mail:
u8834804@cc.nctu.edu.tw

等應用與演變，資訊組織的範圍更加擴大。

目前除了圖書館界之外，其他科學領域也積極投入資訊化組織的發展與研究。因此，metadata 成為全球資訊基礎建設中，十分重要的一環。而如何規劃處理與儲存管理這些 metadata 所編製的資料記錄，透過整合性的管理與檢索，亦是一項重要之工作。但是 metadata 的格式相當多，例如：EAD、GILS、FGDC、MARC、CIMI、TEI、DC 等等，甚至也有許多格式是依據這些格式延伸設計的。此外，一個單位還有可能會有多種不同的資料型態和目錄型式，也因此可能會採用多種 metadata 格式。因此，設計 metadata 系統時，決不能只針對一種格式，必須能夠兼容並蓄各種異質 metadata 格式，除了特定的轉換格式之外，在資料匯入匯出方面，亦能遵循 XML 格式交換，才能達成電子資料交換的實際應用需求。

本研究主要是使用 XML Schema 來建構 XML 框架 (framework)，透過內部元素的完整階層樹架構，解決傳統物件導向語言在資訊分享上的缺失，以及元件技術無法表達類別及實體關係的困難點，並而解決異質 metadata 之間儲存與管理的相關問題，分析建置能夠處理多種 metadata 資料，並能統籌檢索與交互權威控制的系統架構。

二、相關文獻

(一) metadata

「meta」一字源自於希臘文，表示更高等或更基礎的本質。^[1]「metadata」在資訊組織界最普遍的解釋是「data about data」^[2]，也就是資料描述資料的意思，用來描述如：指示儲存位置、資源尋找、文件紀錄、評價、過濾等功能。例如圖書館自動化系統所使用的機讀格式 (Machine readable cataloging，簡稱 MARC)，即為一種 metadata。國內對於「metadata」現有的翻譯名詞有元資料^[3]、超資料^[4]、詮釋資料^[5]等。依據不同社群 (community) 多樣性的需求，

^[1] A User Guide for Simple Dublin core, Draft version 5.1 Retrieved July 31, 1998, from http://purl.org/DC/documents/working_drafts/wd-guide-current.htm

^[2] Stuart Weibel, Jean Godby, and Eric Miller, "OCLC/NCSA Metadata Workshop Report," from http://www.oclc.org/oclc/research/conferences/Metadata/dublin_core_report.html

^[3] 吳政勳，從電子檔案和元資料看未來資料著錄的發展趨勢，中國圖書館學會編，《海峽兩岸圖書館事業研討會論文集》(臺北市：編者，1997)，頁163-174。

metadata 也有多種不同的資料格式，目前至少已有 20 種以上屬國際標準或漸形成標準的 metadata 格式存在於各學科領域。依據 metadata 格式的特性及結構，Lorcan Dempsey 和 Rachel Heery 將之分為三種類型：^[6]

1. 單一格式 (simple formats)

此類型的特性多是由某公司 (例如 :Lycos、Altavista、Yahoo 等) 專屬性的資料結構，強調全文式索引功能，通常由機器自動擷取資源後直接產生索引以提供檢索。

2. 結構化格式 (structured formats)

此類型大多是屬於標準化中的格式，特性是以欄位作為結構主體。例如：Dublin Core、IAFA templates、RFC1807、SOIF、LDIF 等。它們所含括的描述資料基本上足以讓使用者找出有興趣之相關資源。這類型格式記錄以簡單的設計為原則，未受專門訓練的資料提供者，也能容易上手建立 metadata 記錄。但是相對地，格式的設計上較缺乏物件或資源間不同層次的關聯性功能。

3. 豐富格式 (rich formats)

此類型為已符合國際標準的格式，其特性是屬於詳盡且複雜的標誌來呈現資源的內涵，例如：ICPSR、CIMI、EAD、TEI、MARC 等，這類型的格式通常需由研究學者或具專門知識者來描述資源的內涵。

資訊檢索可即性 (accessibility) 的提昇，以及如何能夠跨資料庫、跨系統、跨網站之間作資訊透通與整合性的搜尋是重要的目標。當然透過前後一致豐富格式的 metadata，能夠有效提昇搜尋的效率，不過前提是描述性的 metadata 必須相同，或者是在不同 metadata 之間要能夠互相映對。總之，一定程度的相容性是必要的，而達到不同 metadata 之間的相互操作性則是眾所期待的事。^[7]

目前，已有一些單位針對 metadata 之間的相互操作性提出討論，例如美國

[4] 陳昭珍，電子圖書館資訊組織問題之探討，載於：中國圖書館學會編，《海峽兩岸圖書館事業研討會論文集》(臺北市：編者，1997)，頁175-196。

[5] 陳雪華，《圖書館與網路資源》(臺北市：文華圖書館管理，1996年)，頁206。

[6] Lorcan Dempsey and Rachel Heery, "A review of metadata: a survey of current resource description formats." Version 1.0. 19 March, 1997, from http://www.ukoln.ac.uk/Metadata/desire/overview/rev_01.htm

[7] Anne J. Gilliland-Swetland, "Defining Metadata," in *Introduction to Metadata: Pathways to Digital Information*, ed. Murtha Bace (Los Angeles: Getty Research Institute, 1998), pp.1-8.

國會圖書館主持的一項計劃就是建立起USMARC、Dublin Core 以及 GILS 三種格式之間的溝通結構與運作機制。^[8] 相互操作性是建立一致性資源儲存系統的重要條件，並且依賴於各互通 metadata 格式之間差異門檻的最小化。進一步更理想的作法是藉由建立一個更上位層次的 super-metadata 來達到相互可操作性的目標，如此則可使現有及未來的 metadata 格式能夠同時顧及本身的特殊性又能與其它 metadata 之間有良好的整合。^[9]

(二) XML

由於標準通用標示語言 (Standard Generalized Markup Language, SGML) 的複雜，及軟體工具成本過高與使用不易，因此限制了其應用的範圍。而 HTML 則受限於其在 Web 上使用的專屬性，因而全球資訊網聯盟 (World Wide Web Consortium, W3C) 公布一套依據 SGML 標準製訂，且能適用於 Web 作業環境的 XML，作為新一代的資料標示語言。XML 支援語言中立 (language neutral) 的定義和平台中立 (platform neutral)，並且能提供定義在 Web 環境上結構化文件交換的資料格式。XML 是由 SGML 的專家們和 W3C 合作制定簡化 SGML 的子集合規範，並於 1998 年 2 月正式公布 1.0 版本。不像 HTML 由固定的標籤集合所構成，XML 允許使用者自行定義所需的標示語言。因此，可以將資料內容以清楚的標籤表現其意義，並可廣泛地應用在各種領域，例如：化學、電子、商業等等。應用相關的標籤界定個別的資料項目或資料群組，使應用程式能夠很容易地將 XML 文件中的資料分離而加以利用。^[10]

基於 XML 支援語言中立的定義和平台中立的特性，因此自動化系統都能夠輕易地修改符合以 XML 交換資訊的功能。而在實際應用的因素上，XML 已有許多應用在電子商務成功的例子，不僅可整合不同類型的文件，並且可提供機器之間方便的資料交換格式。

(三) XML Schema

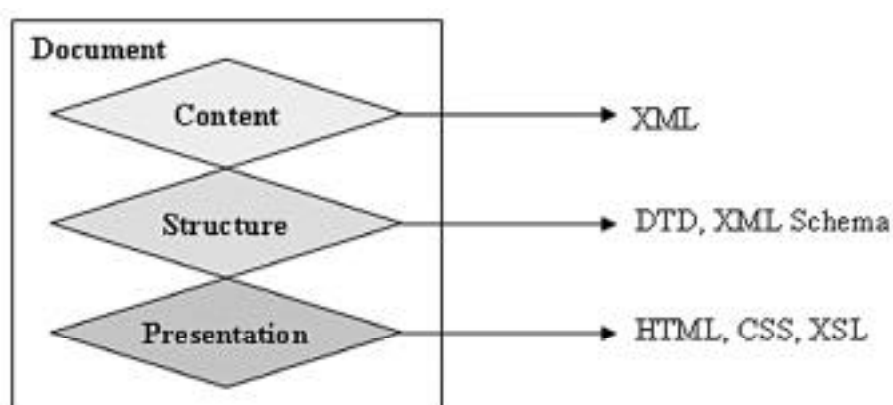
[8] Network Development and MARC Standards Office, Library of Congress, "Dublin Core/MARC/GILS Crosswalk," (2001.-03-12), from <http://www.loc.gov/marc/dccross.html>

[9] A. Chilvers and J. Feather, "The Management of Digital Data: a Metadata Approach," *The Electronic Library: the International Journal for Minicomputer, and Software Application in Libraries* 16:6 (Dec. 1998), pp. 365-372.

[10] W3C Recommendation, World Wide Web Consortium, "Extensible Markup Language (XML) 1.0," (1998), from <http://www.w3.org/TR/1998/REC-xml-19980210>

XML 文件有兩種形式，一種稱為 Well-formed XML，另一種則稱為 Valid XML，兩者最大的差別則是在於是否有 DTD。DTD 主要是定義和規範 XML 文件架構的一組規則，以便能提供應用系統或剖析器 (parser) 驗證 XML 文件結構的有效性，或是提供編輯工具 (authoring tool) 能夠遵循定義的架構來產生 XML 文件。不過由於 XML 只是 SGML 的子集，而 DTD 主要是移植於 SGML 的型別定義，因此，DTD 的語法宣告不僅難以學習，且使用上能夠提供的定義並不能完全滿足現實 metadata 資料定義的需求。例如，DTD 僅能處理文字型態的資料，並不具備其他種類的資料型態宣告；且僅提供屬性 (attribute) 欄位的預設值宣告，而無元素 (element) 欄位的預設值定義；另外，也無法將 XML 文件視為一個物件導向的型別。^[11]

基於上述的問題，W3C 的 Schema 工作小組參酌業界 Schema 相關的提案，如：「文件資訊內容描述」(Document Content Description, DCD)、「資源描述架構」(Resource Description Framework, RDF)、「物件導向 XML Schema」(Schema for Object-oriented XML, SOX) 以及 XML Data，整合這些業界所提出規格的優點及特性，以訂定出 XML Schema 的標準供實際使用上的需要，並於 2001 年 5 月 2 日正式成為 W3C 的一個建議 (Recommendation) 標準 (圖一)^[12]



圖一：文件在內容、結構及展示方面所使用的不同標準

[11] Kurt Cagle, *XML Developer's Handbook* (San Francisco: SYBEX Inc, 2000), p.272.

[12] W3C Recommendation, World Wide Web Consortium, "XML Schema," (2001), from <http://www.w3.org/XML/Schema>

XML Schema 本身也是一種標準的 XML 文件，其功能與 DTD 一樣，都是定義 XML 文件結構的一種標準。和 DTD 比較，XML Schema 能夠提供下列的優點：^[13]

1. 使用標準的 XML 來定義整個 XML Schema，不僅編輯容易，也能夠透過標準的剖析器來解析。
2. 能夠明確地設定結構中的元素數量，亦能宣告節點的內容為封閉式（僅能包含特定宣告的子元素），或是開放式（能夠包含任何子元素）的結構。
3. 透過名稱空間（namespace）的使用，使能夠在相同的文件中載入多個 XML Schemas。
4. 能夠在 XML Schema 內定義原型（archetype），提供基本的繼承（inheritance）封裝（encapsulation）等物件導向功能。

(四) 統一化模型語言 (Unified Modeling Language, 簡稱 UML)

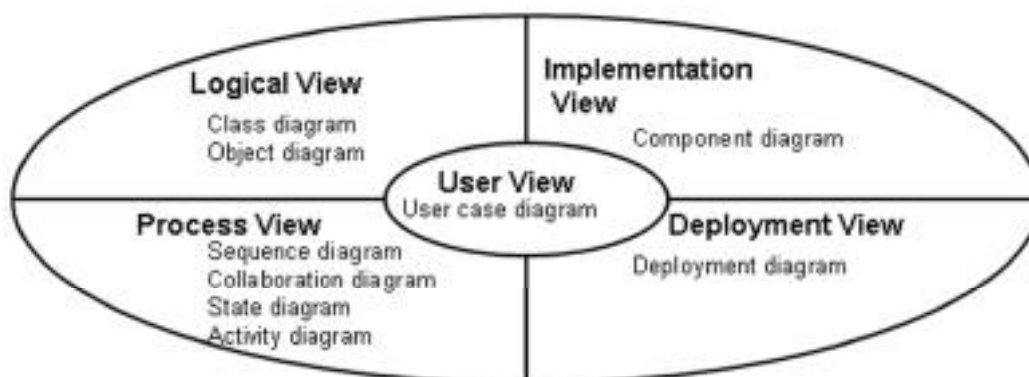
UML 是由三位世界級物件技術大師 James Rumbaugh、Grady Booch 和提出 OOSE (Object-Oriented Software Engineering) 方法的 Ivar Jacobson 所共同研究開發出來的物件導向分析與設計的標準語言。

過去由於軟體業界沒有一個共通的物件導向分析與設計的標準，因此不同的開發者及使用者間，要進行溝通是一件很困難的事情。UML，就是為了要把軟體開發初期所進行的物件導向分析與設計，用一套大家皆遵循的標準化語言來開立規格，並且利用圖示法 (Notation) 表達來建立及保存一個完善的文件紀錄，運用以元件為基礎的物件導向技術來開發軟體。

如圖二所示，UML 總共從五種觀點定義了九種不同的圖形，分別是使用者觀點的使用者案例圖 (Use Case Diagram)、結構觀點的類別圖 (Class Diagram)、物件圖 (Object Diagram)、行為觀點的循序圖 (Sequence Diagram)、合作圖 (Collaboration Diagram)、狀態圖 (State Diagram)、活動圖 (Activity Diagram)；實作觀點的元件圖 (Component Diagram) 以及環境觀點的部署圖 (Deployment Diagram)。^[14] 以便從各種不同的角度將概念透過符號表示，並將概念間的相互關係藉由符號的路徑來描繪出整個系統。

^[13] Ioannides, Demetrios, "XML schema languages: beyond DTD," *Library Hi Tech* 18:1(2000), pp. 9-14.

^[14] Sinan Si Albir, "UML in a Nutshell" (Sebastopol, Calif.: O'Reilly, 1999)



圖二：UML 模型的五種觀點及九種圖形

三、設計理念與系統特性

metadata 存在著內在與外在的形式，外在即是資料的呈現方式，包括語法與結構；而內在則是資料的語意 (semantic)。因此，我們可以歸納出兩種結論：

1. metadata 就是資料實體 (instance)，藉由語法與結構描述出的資料物件。而使用 XML Schema 來定義 metadata，可以達成資料間繼承、封裝、合併與衍生的特性，對於資料的再製、應用與處理可以達到更複雜的應用模式。
2. metadata 與資料間存在互動的關係，因為資料實體是藉由 metadata 描述而成，其作用有如資料庫系統之資料字典 (data dictionary) 般的詮釋意義。

此外，由於 metadata 主要目的在於描述資源的屬性、特徵。就本系統架構而言，是依據 XML Schema 來描述系統之綱要 (schema)，再藉由系統綱要控制與管理 metadata。所以，可將本系統界定為透過 metadata 來建構處理 metadata 現實需求的 super-metadata 系統。

本系統考慮分析的層面包括下列三點：

1. 結構

metadata 主要是為了描述資源而產生的，參考現有各式 metadata 的定義，均具備一定程度相似的結構化組織，彼此之間只是詳簡程度與設計觀點有所不同而已。因此，主要結構的執行能力包含結合處理 XML 的剖析工具 (parser) 與編輯工具 (authoring tool)，以及判斷是否符合 XML Schema 結構的有效性檢查。

2. 深度

由於 metadata 主要是描述資料的屬性與特徵，因此和圖書館目錄的目標並無太大的差異。欄位基本是根據階層式結構往下細分，建構各元素與子元素 (sub-element) 之從屬關係，並倚靠屬性值作為各元素所屬的特性。欄位的組成模式基本包括定長欄與變長欄兩種，而定長欄又有一般性定長資料型態 (例如身分證字號、ISBN、ISSN 等固定長度資料) 與特性指示型定長欄位 (例如 CMARC 的欄位 100 「一般性資料」、105 「資料代碼欄」，其每一個組成字元均有其特定的意義與範圍)。^[15] 無論是定長欄或變長欄，其子欄位或屬性均有交互出現定長欄或變長欄之可能性，因此系統在欄位結構的處理能力，必須能夠涵蓋上述的情況。

3. 廣度

metadata 範圍不僅是描述資訊資源之外，更重要的是如何標記不同物件之間的相互關係。也就是說，metadata 除了著重在資訊本身實體的描述之外，還需能詮釋不同 metadata 之間的關係，而且此種串聯關係，必須能由單純的物件範圍，擴充至時間、空間、人、事等層面。因此，在系統的廣度必須橫向考慮到下列不同的應用範圍：

(1) 多種不同的 metadata 格式與型態

為能符合各種學科領域的需求，絕非一種 metadata 能夠涵蓋。即使是圖書館目錄，也絕非一種資料的記述方式能夠涵蓋，所以 super-metadata 系統必須能處理多種的 metadata 格式。本系統「多種 metadata 並存」之系統分析，主要即是基於此項需求。

(2) 多種不同的使用者類型

例如一般的使用者所希望的只是能夠知道有哪些物件 (資源) 可以取得，而研究人員則可能希望能從既有的資源取得延伸的文件。為了符合各種不同層面的使用者應用，本系統「使用者介面」之系統分析，主要即是基於此項需求。

(3) 多種不同的資源類型

某些資源的重要性可能有時間區間的限制，或是只在很短暫的時間範圍內有效 (例如氣象資訊的 metadata)；某些則是依據不同的描述內容而

^[15] 余顯強，《多種機讀格式並存之編目與檢索架構研究》(臺北市：漢美圖書有限公司，1997)，頁 54-60。

分別適用於學術單位或企業商務。某些資源的組成非常簡單；而有些資源又極度的複雜，這些不同資源類型都有可能會依據實際的需要，必須被著錄於同一系統內。因此在系統分析上，必須考慮「權威控制」之架構，以便能達成多種不同資料類型間透通性的檢索需求。

(4) 多種不同的資訊提供者

企業往來的商務資料、研究機關與學校的研究文獻等，不同組織依據不同目的而產生的資源，均有數位化的需求。另外，各相關組織之間所建立的 metadata 亦有彼此間資源交換的需求。因此，系統必須能保有 metadata 本質的結構，提供透通的轉載 (Import /Export) 或是連結外界的資源。本系統「資料匯入 / 匯出」之系統分析，主要即是基於彼此間資源交換的需求。

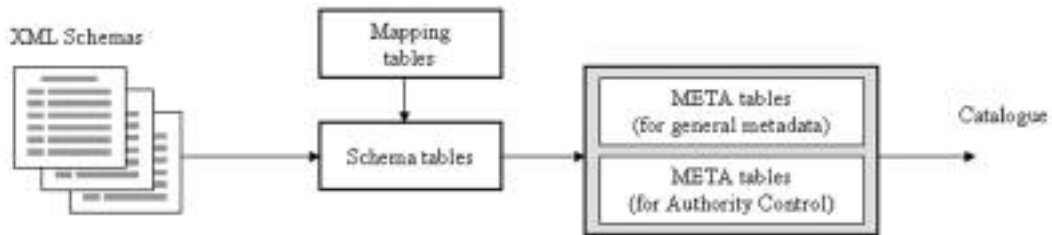
如何將 metadata 資料藉由抽象化語言描述，令其不僅可與電腦溝通，並可作為異質系統間的整合溝通介面，則首推 XML。XML 具有 SGML 的特性，又不似 SGML 那麼複雜；XML 便於在網路上傳輸，又提供 HTML 所沒有的彈性及精確性，所以已成為廣泛使用的標識語言。而 XML Schema 提供的基礎資訊，讓 XML 處理器能夠剖析文件內容和分析出應用程式所需全部的資訊，或讓應用程式可以存取的格式，也就是說，XML Schema 提供了資料檔案之間連結的關鍵。

為了符合前述系統分析之需求，必須採用 XML 框架來設計一個能夠同時兼容多種 metadata 綱要的新系統架構模型。藉由多種 XML Schema 來定義出系統綱要的結構，提供系統能夠處理各種 XML，使之能夠達成多種異質 metadata 並存處理的需求。而透過檢索點的串聯定義，使各種異質 metadata 能夠在一致性的查詢功能之下被檢索出來。為使多種 XML Schema 能夠同時存在於系統之內，因此系統的主要設計方法便是仿效 XML 的階層式架構。而在資料庫設計方面，依據上述需求的分析結果，如圖三所示，必須包含四組主要的表格：

1. XML Schema 表格：將匯入之 XML Schemas 轉換成關聯式表格，決定系統所能呈現的 metadata 架構。
2. 映對表格 (mapping table)：用來結合 XML Schema 與系統索引表格之間的關聯，定義每一個元素的檢索點 (access point)。
3. META 表格 (參數檔)：融合映對表格與 XML Schema 表格，產生系統實

際運作之系統編目參數檔案。

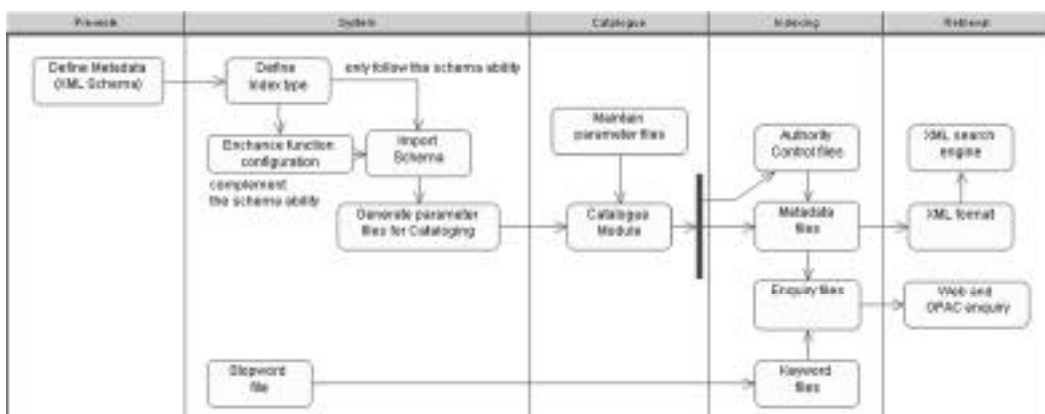
4. 權威控制表格：提供資料與資料之間相互參照的規範依據來源。



圖三：將 XML Schema 匯入系統產生編目參數之流程

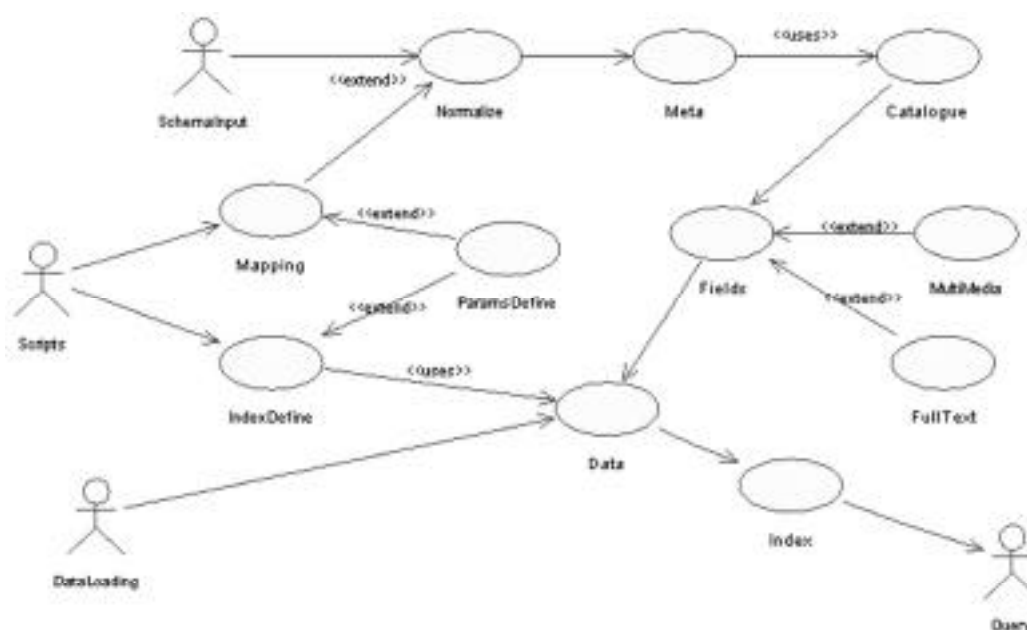
四、系統架構分析

設計資訊系統時，首先需決定的是其資訊該如何組織，資訊組織方式決定了系統之資料庫架構。因此本系統之架構圖如圖四所示，主要即著重於資訊之組織模式。基於此組織模式的目標，依序切割成為前置作業、系統作業、編目作業、索引處理、檢索作業等五個部分，而此五個行動之分隔集合的圖像化結構，以 UML 描繪的區塊稱之為 swimlane (Grady Booch, James Rumbaugh, and Ivar Jacobson (1999) *The Unified Modeling Language User Guide*, Reading: Addison-Wesley Inc.)，主要目的是將系統之活動狀態分割為群組，方便後續分析作業能夠將這些群組分配給必須執行這些活動的物件。



圖四：系統整體活動圖 (Activity diagram)

整體系統行為如圖五所示，依據 UML 之 User Case，透過外部觀點，而非內部結構來說明本系統的外部需求與系統所提供之主要功能。



圖五：系統 User Case 模型圖

(一) 模組架構

一般資訊系統均是將 XML 作為文件處理結構，但本系統設計之核心，不僅是將處理的 metadata 以 XML 作為其結構依據，且以 XML 之 XML Schema 作為整體系統運作的架構使之成為系統的主體框架。依循前述五個階段之系統組織模式，劃分成下列四個模組：

1. 系統綱要模組 (Schema construct module) —提供 XML Schema 輸入並轉化成系統綱要結構的功能。
2. 編目模組 (Catalogue module) —包括權威紀錄與一般 metadata 資料之編輯維護功能。
3. metadata 資料匯入 / 匯出模組 (Import /Export module) —主要是以 XML 格式作為匯入 / 匯出之基本格式，並可依據需要選擇為 Well-formed XML，或是 Valid XML (包含原生 XML Schema 或轉換之 DTD 宣告)
4. 查詢模組 (Enquiry module) —包含 Web 和 OPAC 檢索介面。

(二) 資料庫架構

由於 XML 和所屬的 XML Schema 均為階層式的結構，但是關聯式資料庫系統主要是架構在表格與表格之間的關聯性。因此在本系統的資料庫結構上，必須將階層式的資料來源轉換成關聯式資料庫表格。依據資料檢索和資料間的參照關係，需要建構下列三組表格：

1. 能夠儲存階層式的 XML 和 XML Schema 的關聯式資料庫表格。
2. 儲存與管理 metadata 資料檢索時，不同檢索點之間的索引表格。
3. 各個不同檢索點之間彼此的權威控制所需的表格。

(三) 檢索功能

分散式資訊的整合檢索雖然不是 metadata 的直接內涵，但在網路式環境下卻是一個相當重要的課題。因此在設計檢索功能時必須能夠完整、快速和精確地搜尋 metadata 文件，且提供更深入及更多面向的資訊，以便能夠完整呈現 metadata 彼此之間的價值。由於資料存取方式 (access method) 及檢索方式 (query method) 的層次不同，資料檔案可以有各種不同的結構，但檢索的基本原則卻大致相同，尤其是透過結構化查詢語言 (Structured Query Language, SQL) 的標準介面執行資料庫的查詢。而系統在處理 metadata 時，主要的功能必須是能考慮到中文與英文在檢索上的差異：

1. 除了 Unicode 以外的字集，英文字元 (Character) 以單一位元處理，中文字元則是以雙位元處理。
2. 英文單字字數不定長，中文單字則固定。
3. 英文單字以空白為區隔，中文則無區隔符號。
4. 英文字元依據字母順序排列，中文普遍是以筆畫、筆順、部首依序排列。

根據上述文字處理的差異，metadata 文件的內容在建立索引時必須將中英文分開抽離建立索引檔。因此，在功能上必須有所分別：

1. 英文以單字查詢為主，中文則以語詞查詢為主。
2. 英文單字查詢需有容錯功能，例如拼音近似查詢，中文單字則無。
3. 英文有單複數、第三人稱、時態變化等區別，中文則無。
4. 英文單字有切截查詢需求，中文則有語詞切截查詢需求。
5. 中文語詞有空格詞句、折行詞句查詢需求，英文則無。
6. 中文有斷詞需求，英文則無。

(四) 系統架構

基於本系統各模組應用上的需求，現已完成之主要功能如下列所示：

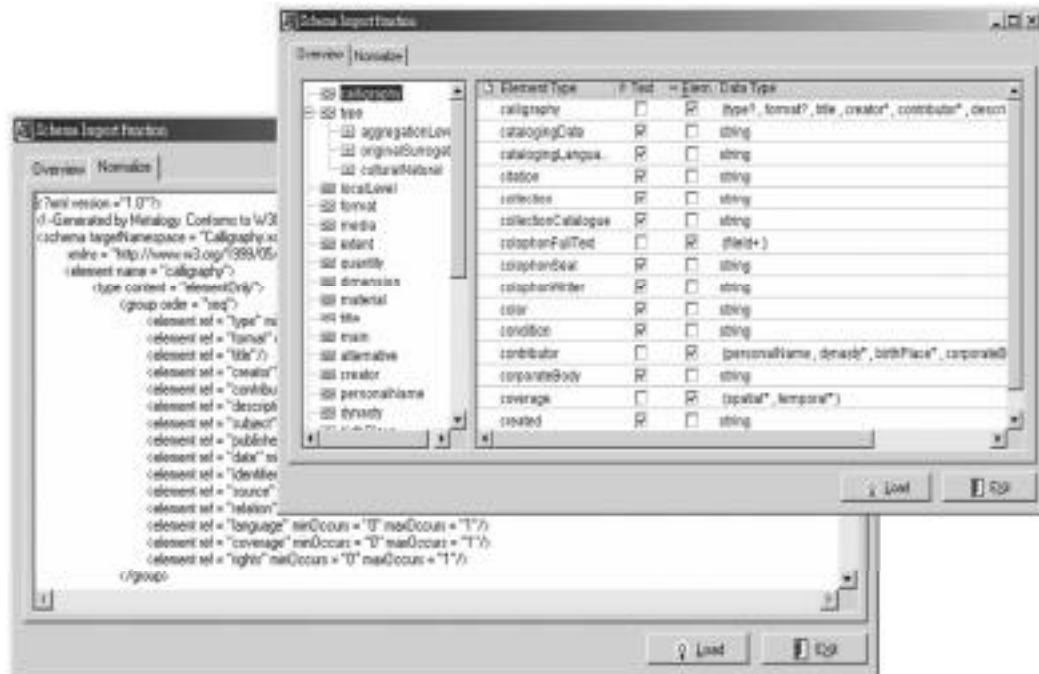
1. 載入 XML Schema 建立資料庫所需的 meta 綱要。
2. metadata 編目功能。
3. 索引典建立及權威控制功能。
4. 數位化資料的管理與描述。
5. 檢索功能。
6. 權威款目參照查詢功能。
7. XML 格式資料匯入匯出。
8. 使用者權限管理與控制。
9. 多語文操作介面 (Multilingual)。

五、實 作

(一) 系統綱要模組

在系統所使用的資料結構部分，本研究是採用透過 XML Schema 宣告而定義出來，因此必須提供匯入 XML Schema 與建立系統綱要的功能，此功能主要包括三個執行部分：

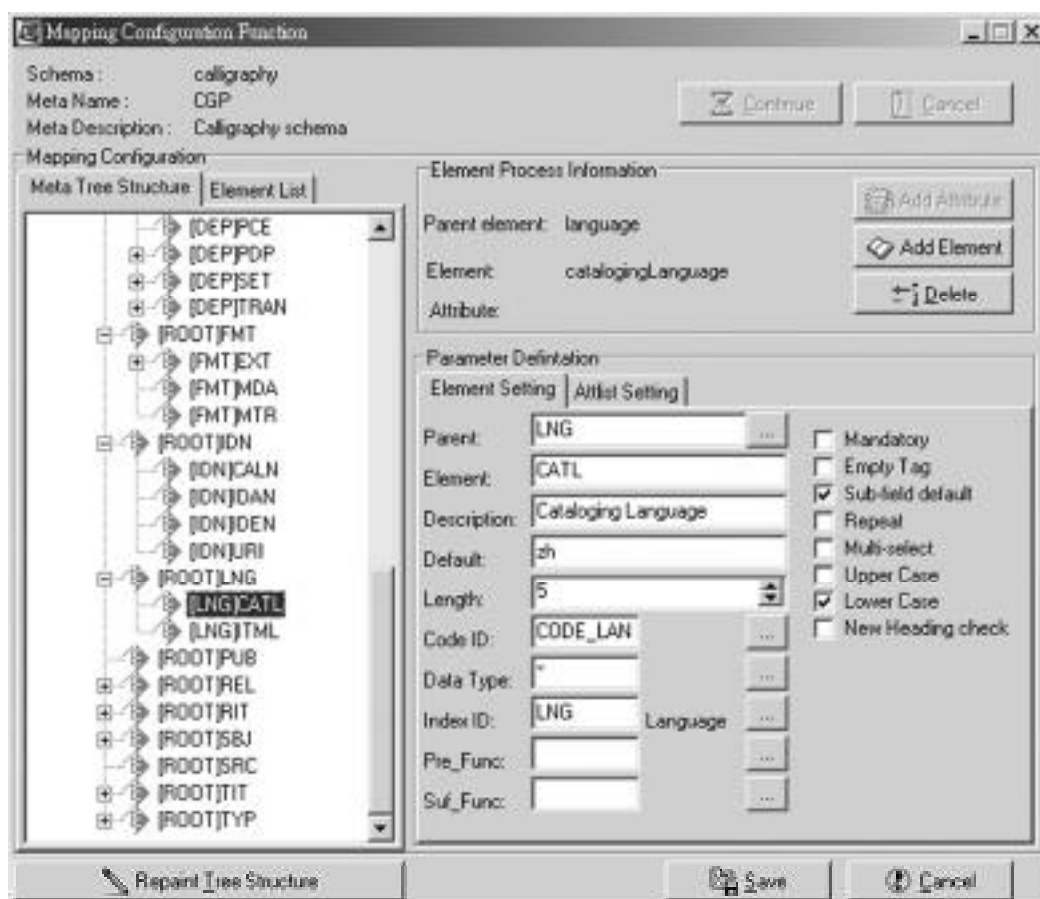
1. 因為本系統主要資料結構的綱要是來自 XML Schema，也就是 metadata 的依據。因此資料庫內部必須能夠儲存匯入之 XML Schema 資訊（參見圖六）。由於 XML Schema 是屬於巢狀式的階層結構，類似於物件導向資料庫表格。而本系統為簡化前端的軟體開發複雜度與後端資料庫透通性，採用關聯式資料表格做為主結構。因此一個 XML Schema 檔案匯入至系統時，首先必須依據 XML Schema 元素的特性將之分解成多個副表格。



圖六：匯入 XML Schema 建構所需的 metadata 綱要

2. 由於 XML Schema 在宣告上並不具備特殊處理功能 (extra function) 所屬權威款目、索引種類等欄位定義。另外，使用者在處理資料時也可能希望將匯入之元素做適度轉換、合併，或只採用部份元素等等。基於上述考量，系統必須另外存在一組映對表格，作為將 XML Schema 檔案定義轉化成系統內部 meta 結構的中間檔，也就是形成本系統綱要的附屬參數依據。
3. 映對表格可以在匯入 XML Schema 時由系統依照該 XML Schema 內容自動產生，但仍需使用者人工檢視各元素的資料格式、特殊處理功能、輸入長度、所屬權威款目、索引種類等欄位定義等設定內容 (參見圖七)。另外也可事前先編輯好，再執行匯入 XML Schema 的作業。

透過系統將 XML Schema 原始定義與映對表格比對結果，即產生系統內部的 meta 結構，也就是作為該類型資料編輯的依據。

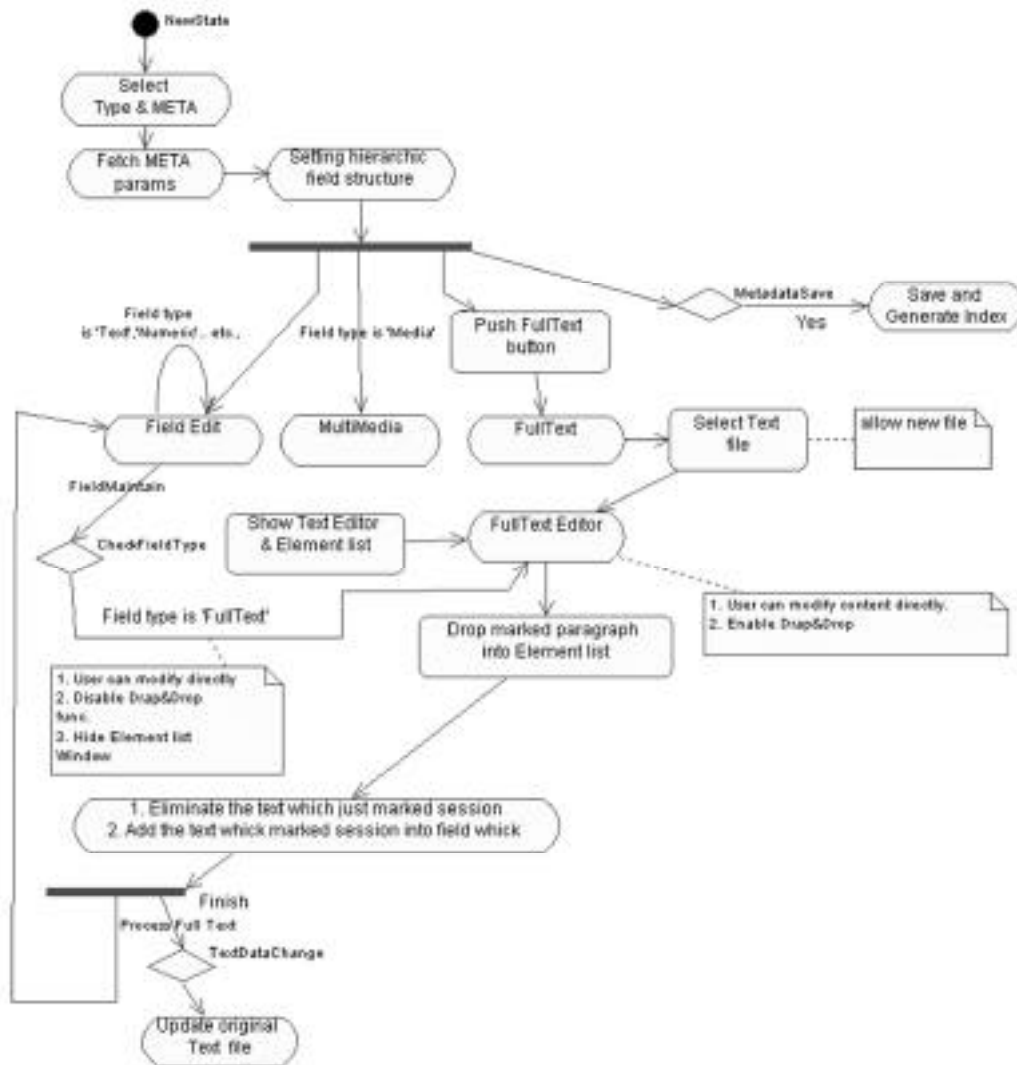


圖七：映對表格參數設定檔設定畫面：定義各元素之索引項目與延伸功能

透過系統將XML Schema原始定義與映對表格比對結果，即產生系統內部的meta結構，也就是作為該類型資料編輯的依據。

(二) 編目模組

編目模組主要提供各種不同 metadata，以及權威資料的編目作業。詳細之編目模組資料活動圖 (Activity diagram) 如圖八所示。透過活動圖來描述整體編目功能模組的活動流程，以方便檢視此一模組的內部處理行為。



圖八：編目模組資料活動圖

1. 資料輸入功能

此功能主要是提供 metadata 的編輯作業，系統允許輸入之 metadata 種類，取決於先前匯入之 meta 種類，也就是匯入之 XML Schema。

當系統完成載入任何一種 XML Schema，即會建立對應之 META 表格（也就是綱要表）。如圖九所示，當進入編目模組時，即可依據欲編輯之 metadata 種類，選擇所屬之 meta 項目進行編目。此時，系統能夠依據所屬之 meta，針對各個欄位提供設定之複製、刪除、插入子欄位、代碼與連結多媒體檔案等功能，並



圖九：編目類型依據匯入之 XML Schema 多寡，編目時可選擇資料的編目格式

可在編目作業中直接啟動查詢作業，進行metadata複製功能。編目人員若認為編目模組之欄位編輯器檢查程序過多，或是需要加快編目程序，省略即時線上標目查核檢查時，可以先離線使用任何文書編輯器編輯某一metadata資料，再透過全文編輯器，載入成為系統的metadata。

全文資料編輯器主要是提供標示全文的獨立編輯工具，使系統能夠將報告、文件、書刊、論文等全文資料標示成metadata。編輯器內分為兩個編輯區，左邊為操作人員所選擇之 schema 的元素表列，右邊則為匯入之全文內容編輯區，操作人員可以直接在編輯區內增修刪改內文。標示的方式為，操作人員先反白一段內文，再將反白之內文拖拉至左方之目的元素上。為避免文字內容含有XML標示所用之保留字元（&、<、>、"、'），因此在處理過程會將相關字元轉換成通用實體（general entity）字元^[16]，例如將「&」字元轉換成「&」，「<」字元轉換成「<」。

多媒體編目可以整批或單筆匯入多媒體檔案資料，並進行簡略描述。若於metadata編目作業前進行，可於編目時進行多媒體資料之鏈結；若於編目之後進行，亦可修改metadata資料，以進行鏈結。若有大批多媒體檔案需同時進行匯入，則可使用批次的方式，以節省作業時間。由於多媒體資料包含影像、聲音、圖片等型態，所佔資料空間可能非常龐大而不適合直接儲存於資料庫中，因此操

^[16] Simon St.Laurent, *XML: The Primer* (Foster City, CA: M&T Books, 1999), pp.103-104.

作人員可將多媒體資料依據內容大小決定儲存的資料庫中或其他儲存體內，再透過資料庫建立虛擬指標，將metadata與多媒體資料鏈結起來。

2. 權威控制

如同前一項編目功能，當系統載入任何一種XML Schema時，只要指定其為權威控制之XML Schema，系統即會建立對應之權威控制META表格。權威控制的目的，在於維護標目的獨立性、一致性與關聯性。Metadata中的標目元素均應有明確的定義，持其獨立性，以避免混淆不清。當進行metadata編目，在參數指定之權威控制欄位編輯時（例如：年代、姓名、地名等欄位），系統會隨時執行標目查核，以便自動進行權威控制之欄位連結，編目人員亦可依據實際需要選擇是否新增標目，以擴增相同款目名稱卻不同標目之權威紀錄，並自動建立相關之權威紀錄。

除此之外，系統尚可依據已匯入資料庫之權威META表格，執行下列功能：

1. 匯入外界已建立的權威紀錄。
2. 權威檔的匯入匯出語法為XML。
3. 可建立權威紀錄之權威款目、見、參見款目。
4. 新增、修改紀錄時，可同時新增、修改相連結的權威紀錄。
5. 可顯示相同或類似之權威紀錄供編目者選取。
6. 當使用者鍵入錯誤的查尋詞時，系統可在權威檔中找到近似的一串語詞，提示使用者。
7. 可以利用權威檔做查尋詞的擴充檢索。

(三) metadata 資料匯入 / 匯出模組

metadata資料的匯入匯出主要是提供系統透過XML和外界做資料交換，並可以多種格式載出Well-Format的XML檔案，供外界其他系統處理。匯出時可以透過查詢來載出所需的特定資料，或指定範圍執行批次之載出作業。而載出的metadata可以一筆或多筆記錄於一個檔案，但不支援一個檔案中包含多種不同的metadata紀錄格式的XML文件。

metadata資料載入時，可以透過重複條件之判斷來決定載入之資料是否已經存在資料庫內。重複條件判斷之參數可以設定一個以上的判斷欄位，設定其"AND"或"OR"的關係。其中"AND"表示所有設定的欄位內容值都必須相同才視為重複資料，而"OR"則是設定的欄位內容值有任何一個相同就視為重複資

料。

整體而言，資料匯入 / 匯出模組主要執行的功能如下：

1. 只接受對應到特定 XML Schema 的 XML 文件實體，包含了 Valid XML 文件實體及對應到已知 XML Schema/DTD 的 Well-formed XML 文件實體。
2. 針對某個 XML Schema，系統能整批匯入或匯出 XML 格式的 metadata 紀錄。
3. 如果欲匯入的 XML 文件實體，其 XML Schema 在系統中尚未存在，可選擇讓系統建立內部 XML Schema / DTD 定義，或選擇放棄。
4. 如果欲匯入的 XML 文件實體，其 XML Schema 在系統中已存在時，系統會針對這些文件實體進行有效性驗證，並可供檢視驗證的紀錄 (validation log)
5. 匯出 XML 文件實體時，可選擇是 Valid 或 Well-formed 的形式。
6. 匯入或匯出時，系統會提示轉換中的筆數訊息。

(四) 查詢模組

可藉由查詢的功能來搜尋已編目完成的資料，並顯示其狀態，以做進一步之維護或載出。查詢模組之顯示格式可經由參數設定而不同，包括條列式顯示、簡略顯示與詳細顯示等三種顯示格式。不論是查詢一般的 metadata 或權威資料，皆可針對全部 meta 或單獨的 meta 所屬之 metadata 進行查詢動作。執行查詢時，為考慮各種 meta 的 metadata 欄位相關的特性，系統能夠將一個以上的索引項目結合成一個檢索點 (既檢索項、檢索欄位名稱)，將相似的欄位屬性結合，以獲得更多的查詢結果，或提供使用者在查詢上的彈性。例如將題名、副題名、其他題名等，各自設定其索引項，但以「題名」作為實際的檢索點 (access point)，同時對此三個索引項進行查詢。

六、結語

資訊管理主要著重於資訊的收集、分類、儲存、檢索和傳輸，尤其是在數位化需求殷切且媒體格式眾多的今日，更有其重要性。採用數位化技術來達成資訊管理的需求，才能創造出知識管理的應用目的，而 metadata 是各種知識、典藏物品儲存管理的描述媒介。因此，藉由兼容多種 metadata 達到整合相關的知識體系，再由資訊系統的統合管理，才能符合實際上的應用趨勢。即於此，本研究採

用 XML 框架來設計一個能夠同時兼容多種 metadata 綱要的資訊系統。

而以 XML Schema 為架構所開發之系統來處理 metadata，除了可以獲得 XML Schema 本身所具備的優點，如：物件繼承、封裝、衍續，與相同於 XML 的語法等等，使得以此為基礎發展出之系統所帶出的效益包括下列五點：

1. 使異質 metadata 之間的整合成為可行
2. 提高系統處理綱要結構的彈性
3. 可以有效處理非結構性的資料
4. 使 metadata 不僅是單純的描述性資料，亦可是結合各種物件型態的集合體。
5. 提供網際網路上開放性檔案傳送的標準。

本研究藉由 XML 的 XML Schema 定義系統綱要的結構，且允許多個 XML Schema 能夠同時存在於系統之內，因此，能夠達成多種異質 metadata 並存處理的需求。透過檢索點的串聯定義，可使各種異質 metadata 亦能夠在一致性的查詢功能之下被檢索出來，促使儲存之 metadata 能夠達到資訊的透通性。

整體而言，架構在此研究模型之下所完成的系統能夠達成下列幾點需求：

1. 系統綱要主要依據輸入之 XML Schema 決定。
2. 系統允許兼容多種不同 XML Schema 同時存在。
3. 多種不同格式之資料能夠同時被檢索。
4. 資料處理之結構允許使用者依據綱要調整內定之欄位格式與使用條件。
5. 資料匯入與匯出能依循所屬之 XML Schema 格式。
6. 系統能判定匯入之資料是否符合指定之 XML Schema 格式，並執行重複查核。
7. 資料內部處理能包含欄位式結構、多媒體結構與全文。
8. 系統包含存取控制 (access control) 異動記錄 (transaction log) 等系統管理功能。

A Study on Digital Library Information System Framework Using Multi-XML Schema

Shien-chiang Yu

Abstract

This article is to discuss the storing and managing metadata on XML framework and construct the framework through XML Schema. Also covered is the system structure of handling many kinds of metadata, centralizing the retrieving and cross authority control.

Keywords (關鍵詞): Metadata ; XML schema ; DTD ; Unified Modeling Language (UML) ;
System design
文件型別定義 ; 詮釋資料 ; 統一化模型語言 ; 系統設計

Shien-chiang Yu : Ph.D. Student, Institute of Information Management, National Chiao-Tung University ; Lecturer, Department of Library and Information Science, Fu-Jen Catholic University ; E-mail:u8834804@cc.nctu.edu.tw