

建置數位內容資訊共享之工作平台

A Content Development Workspace for Information Sharing

杜勇進 陳志達 李嘉銘 蔡尚榮 魏明章

國立成功大學電機研究所

E-mail : {tu, andypony, chip, srtsai, wei}@turtle.ee.ncku.edu.tw

摘要

XML 的出現為網際網路帶來了新的應用領域，XML 的可攜性及可擴展性使得網路上許多應用紛紛採用 XML 作為標準。對於網路上日益增加的資料，資訊的取得和分享更顯重要，採用 XML 作為資料儲存的格式，可有效改善目前網路上資料雜亂不易分類的缺點，讓使用者能夠更精確地取得所需的資訊。為了能夠方便獲得想要的資訊並分享交流自己的資訊，並滿足將來資訊能在個人電腦或各種行動裝置上瀏覽，本文提出一套內容開發的工作平台 (Content Development Workspace 簡稱 CDW)。藉由內容開發工作平台，能夠提供 XML 資料建立、儲存、整理及搜尋功能，以建立資訊共享為目標的環境。

關鍵詞：可延伸式標記語言、可延伸式樣式語言、資訊共享

Abstract

XML is an emerging standard for data description on Internet. Using XML technology in formatting and storing data or documents has many potential advantage over the HTML-based approach. We propose to use XML as the core technology to build a storage server for information search and sharing on Internet.

To facilitate the data access, information presentation and data importing to this XML-based storage server, we develop a Content Development Workspace. The workspace

targets on easy content creation, presentation of information and content retrieval from the storage server. In this paper, we will describe the challenges and the solutions in the design of the content development system for information sharing.

Keyword : XML , XSL , Information Sharing

一、前言

根據使用網路系統的經驗，我們充分了解 WWW 在 Internet 的世界中扮演著資訊共享平台的重要角色。目前在 Web 上所發布的資料格式仍是以 HTML 為主，這樣的資料格式，只具備如何展現資料的描述資料，而且結構鬆散，無法進一步被程式來自動分析與處理；同時，雖然目前有許多網站提供免費線上建構與管理個人資訊網頁的功能(如：PC home Online 個人網站)，但使用者還是得肩負起網頁外觀配置與網頁連結管理，這類需具備相當 WWW 知識的工作；而在 WWW 上瀏覽資訊的人，也沒有管道可以隨意針對某一資訊來進一步提供額外的相關資訊，藉以互相交流分享。

因為 XML 文件所擁有的自我描述能力與其結構化的資料格式，極利於程式來自動分析與處理，XML 標準的種種好處意謂著 XML 將成為新一代 Web 系統的核心資料格式。新一代的 Web 系統將有許多 XML-based information sources，而 XML 文件的來源將會是現今各式各樣的電子文件、database 資料以及提供給資訊提供者使用的 authoring tools，也將有許多 agent 程式負責資料的餵送、轉換與合作交換 XML 文件。許多針對 XML 優點所設計出來的 search services 將更能精準地搜尋被需求的資料，並可以針對不同的裝置做對應的

資料展現。這樣的 Web 環境必能將資訊共享與使用者互動推向一個更有系統與彈性的境界。基於上述的動機，本論文將提出一套內容開發的工作平台(Content Development Workspace 簡稱 CDW)，利用 XML Document Storage 當後端儲存系統，提出 Data Capture Template 與 presentation template 觀念，設計 XML Data Importing Tool，並以 Web 介面提供給使用者一個方便操作之資訊共享工作平台。

二、數位內容資料建立與儲存

資訊共享環境，資料的建立是非常重要的。為了讓使用者能在不熟悉 XML 技術的情況下，很方便透過填寫網頁表格來輕鬆建置 XML 文件，勢必要事先建立網頁的輸入欄位與 XML 文件中元素的對應關係，Data Capture Template 就是要負責記錄著這樣的對應關係。由於 XML 文件中不包含展現資料的描述，因此當要顯示一份 XML 文件時，就必需從另外一份文件得到這份 XML 文件中標籤的展現方式，這樣的文件便稱為 stylesheet。Data capture template 與 presentation template 便是我們用來解決上述問題的方案。當建立了大量的 XML 文件資料，後端需有一個穩固且提供容易查詢搜尋的儲存系統，本工作平台引用 XML Document Storage 當資料儲存體，以下介紹各元件功能。

(一) Data Capture Template

Data Capture Template 的目的就是要負責記錄網頁的輸入欄位與 XML 文件中元素的對應關係著。雖然 XML Schema 可以完整地描述 XML 文件的格式，但是 Data Capture Template 除了描述最後產生的 XML 文件格式之外，還必需描述輸入提示的文字、HTML 輸入欄位的型態(“textarea”，“select”)等資訊，這是 XML Schema 所無法確實描述的，因此本系統另外設計 Data Capture Template 來描述。CDW 所設計的 Data Capture Template，採用了與 XML Schema 相近的描述方式來描述最後產生的 XML 資料內容，其與 XML Schema 的性質相似，但更為簡單且符合需求。下頁所示是 Data Capture Template 的格式範例(表 1)與其所描述的 XML 文件結構(圖 1)。Data Capture Template 的內容只可能出現兩種 element 的名稱：“element”與“elementType”，分別代表者 XML 文件中的 simple type element 與 complex type element。每一個 simple type element 會對應到一個網頁的輸入欄位，其可能的屬性如表 2 所示。

在範例中(表 1)，我們以 mask =“(99)Z9999999”限制電話號碼的格式(9 代表任一數字，Z 代表非必須的任一數字)；並以 filter=“[0-9]”限制郵遞區號中不能夾雜任何 0~9 以外的字元。雖然 filter 與 mask 的功能不似 regular expression 般強大，但在許多限制條件並不複雜的情況下，其簡單的語法在使用上反而顯得更為方便。

表 1 Data Capture Template 的格式範例

```
<template name="hotel_information">
  <element name="name" description="The hotel's name" type="text" occurrence="Once" />
  <element name="phone" description="Phone number" type="text"
    occurrence="Once or more" mask="(99)Z9999999" />
  <element name="postal" description="Postal delivery zone code" type="number"
    occurrence="Once or more" filter="[0-9]" />
  <element name="address" description="The hotel's address" type="addressType"
    occurrence="Zero or more" />
  <element name="introduction" description="Introduction" type="textarea"
    occurrence="Zero or once" />
  <elementType name="addressType">
    <element name="city" description="The city" type="text"
      occurrence="Zero or once" />
  </elementType>
</template>
```

```

<element name="street" description="The street" type="text"
          occurrence="Zero or once" />
</elementType>
</template>

```

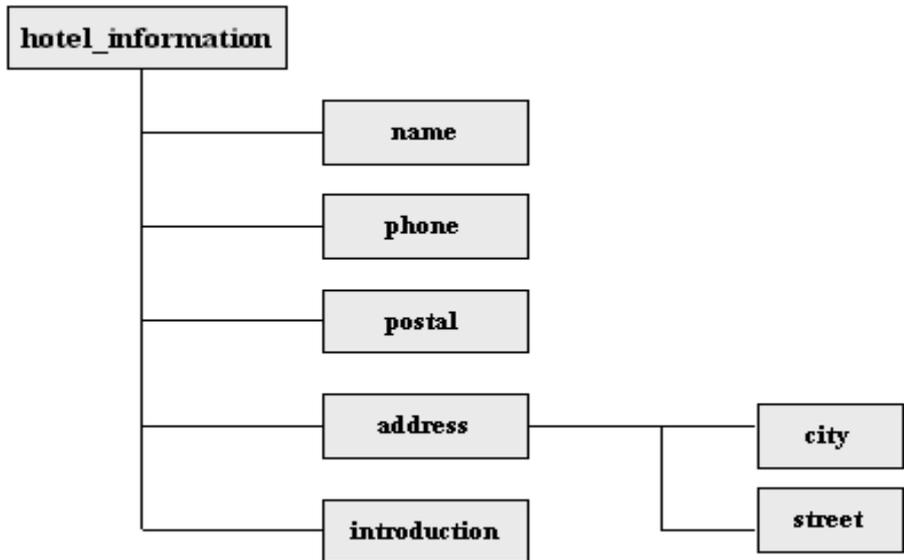


圖 1 XML 文件結構

表 2 Data Capture Template 中 “element” 元素之各個屬性的用途

Attribute Name	Attribute Value
name	該 element 在 XML 文件中的名稱。
description	該 element 對應之 HTML 網頁輸入欄位的輸入提示。
type	該 element 對應之 HTML 網頁輸入欄位的輸入型態。 共有十種：text, textarea, date, checkbox, number, radio, select, image, binary, url
occurrence	該 element 在 XML 文件中與在網頁中的出現次數限制。 共有四種：Zero or once, Once, Zero or more, Once or more
filter regexp mask	限制各個 element 中資料的內容。 filter：輸入的資料不能內含 filter 以外的字元。 regexp：以 regular expression 限制資料內容的格式。 mask：與 regexp 用途相同，但以較簡單的方式表示。

(二) Presentation Template

由於 XML 文件中不包含展現資料的描述，因此當要顯示一份 XML 文件時，就必需

從另外一份文件得到這份 XML 文件中標籤的展現方式，這樣的文件便稱為 stylesheet。Stylesheet 中通常包含了許多規則，描述了欲展示的文件中各個元件的展現方式；而這些

stylesheet 中描述規則的語法便稱為 stylesheet language。以 XML 來說，它的 stylesheet language 為 XSL，在 CDW 系統中，處理展現部分的 presentation template 也就是使用了這項技術來展現 XML 文件。Database designer 在建立 data capture template 之後，可以得知 XML 文件的格式；接下來便可以設計 XSL stylesheet，將 XML 文件轉換成 HTML 的格式，以便在瀏覽器上展現。透過使用 presentation template，information provider 只需在輸入資料後選擇適合的 XSL stylesheet 即可，接下來系統便會自動地將 XML 資料轉換成 HTML 網頁作為展現的用途。如此可以讓 information provider 專心於資訊的提供上，不必學習 HTML 與 XSL 的相關技術，也不必煩惱資料展現的細節。

```

<Picture category="風景類" area="
台灣" AuthorName="Chipmunk">
    <title>夜晚的高雄</title>
    <image> ..... </image>
    <playData day="15"
month="4月" year="2000" />
    <ARecord rate="6">
    <comment>高雄港很美麗
</comment>
</ARecord >
</Picture>

```

(三)XML Document Storage

XML Document Storage[2](簡稱 XDS)是一套 XML 檔案儲存系統，歸類上是屬於 Native XML database。XDS 與一般的檔案系統一樣保障被儲存資料的恆久性(persistence)。其發展的目的是為了提供一般 XML 資訊共享應用系統一個儲存、搜尋與管理 XML 資料的管道。XDS 提供了底下以 XML 特性為主的存取功能：

- 提供一份 XML 文件為單位的輸入/輸出介面

XDS 所提供的輸入/輸出介面不同於 SQL 語法而相當的單純，輸入介面可以直接置入一份 XML 文件的內文，然後取得該文件的 ID；而輸出的介面則是指定目標 XML 文件的 ID，然後可以直接將一份 XML 文件的內文取出。

- 以 XML Schema 標準來當成區分資料的依據

XDS 是以 XML Schema 標準來當成區分資料的依據，在 XDS 中必須以一個 XML Schema 來建置一個 collection，而每一份符合特定 collection 中之 XML Schema 規範的 XML 文件會被 XDS 解讀成是一個 XML object，並會被分配到對應的 collection 裡頭。

- 提供 XPath-based 的 query 介面

XDS 所提供之 XPath[10]-based 的 query 方式邏輯簡單且 query 的內容非常容易來解讀，例如現在有一份描述照片的 XML 文件如下：

現在假設該份文件的 XPath 值：“/Picture/@name”與 “/Picture/title”已被 XDS 做成索引。使用者一開始並不知道這份 XML 文件的存在。但他想找地點是台灣，且標題含有高雄的 XML 文件。所以他要下的 query 內容就是：

```
/Picture/@area=台灣 & /Picture/title~高雄
```

這個 query 內容是以兩個 query rules 以 “&” 作連接，所以 XDS 會先找符合 “/Picture/@area 內容完全等於台灣” 的資料，再找符合 “/Picture/title 內容部分等於高雄” 的資料，兩邊符合的 ID 作 AND 的邏輯運算。這時後該份 XML 文件就會是符合條件的文件，XDS 會回應該份 XML 文件的 ID 給使用者。

可以為 XDS 所接受的 query pattern 是：

```

《xpath value》 [= ~ < > ] 《condition》
{ [ & ] 《xpath value》 [= ~ < > ]
《condition》 }*

```

如上所示，XDS 支援了日期與數值大小的比對以及多個 query rules 的 AND/OR 邏輯運算，其搜尋功能已能充分支援一般應用的需求。

- 支援動態索引

Collection 的擁有者可以依需求動態增加/減少該 collection 資料要索引的欄位，這樣一來未來擴充容易，二來也可避免全文檢索會索引到使用者不需要知道的資訊。

在與 XDS 的溝通方面，XDS 提供了一套

名為 XDS Client API 的 Java 程式介面，我們可以藉由這個介面完成對所有 XDS 資源的存

取。表 3 為 XDS Client API 所提供的主要程式介面：

表 3 XDS Client API 所提供的主要程式介面

Interface	Implementation
createCO	Creating a collection by the XML Schema which had been uploaded.
defineSchema	Define the index information of the specified collection.
getCOInfo/ updateCOInfo	Get / Update the metadata of the specified collection.
addXO / getXO / delXO	Add / Get / Delete the specified XML object.
updateXO	Update the specified XML object.
getXOMeta / updateXOMeta	Get / Update the metadata of the specified XML object.
queryXO	Query in one specified collection by the XPath query rules.
queryAllMeta	Query all XML objects' metadata in all collections.
addGroup / delGroup	Add / Delete a group.
addUser / delUser	Add / Delete a user.

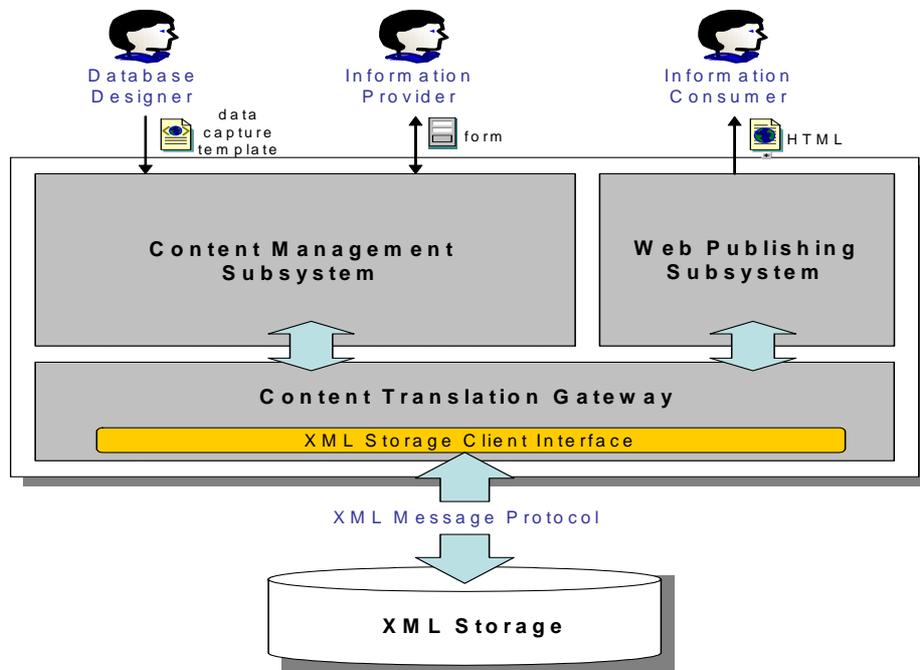


圖 2 系統整體架構圖

三、系統設計

(一)系統架構

整個 XML Content Development Workspace 主要分為 XML Client Interface、Content Management Subsystem、Web Publishing Subsystem 和 Content Translation Gateway 四大部分，系統的整體架構圖如圖 2 所示，以下就各元件的功能分別說明。

(二) XML Client Interface

XML Storage 是一個能夠輔助 AP 處理 XML 資料的系統，在 CDW 中，XML Storage 當作 XML 資料儲存的地方，在不更動原本 XML 資料下，XML Storage 能夠提供類似資料庫的方式處理 XML 資料，而且其功能與模型都能配合 XML 的特性。XML Message Protocol 為 XML Content Development Workspace 和 XML Storage 溝通的訊息格式。

(三) Content Management Subsystem

Content management subsystem 負責與 database designer、information provider、information consumer 的溝通，主要負責下列的動作

- Data capture template 的建立：database designer 可以透過 Data Capture Template Editor 完成 data capture template 建立的動作，或是自行編輯 data capture template 完成後再上傳至系統中。
- 工作環境的操作管理：系統提供了類似 Microsoft Windows 檔案總管的介面，讓系統使用者方便作內容管理的動作，包括了檔案或資料夾的新增、刪除及分享，查詢介面，文件內容的修改，個人檔案的上傳等等，也可以在工作環境瀏覽其他使用者分享出來的文件內容。
- XML 資料的輸入及修改：Data Capture Template Processor 會依據 data capture template 的內容產生供 information provider 輸入資料的表

格；若是要修改舊有的 XML 資料，也是透過相同的表格式介面進行。在完成資料的輸入或修改後，XML Data Importer 便會將資料轉為 XML 的格式儲存。此外，系統也提供了 lock 的機制，可避免多人編輯修改同一份 XML 文件時產生資料不一致的情形。

(四) Web Publishing Subsystem

Web publishing subsystem 負責系統中 XML 資料的展現。展現的網頁皆是以 XML 資料搭配 presentation template 動態產生的，可以避免傳統網站中靜態 HTML 文件更新不易、重複利用性低、相關文件內容易有不一致情形等問題。

(五) Content Translation Gateway

Content translation gateway 負責與 XML Storage 的溝通，主要負責下列的動作：

- Data capture template 的轉換：在 CDW 中，是以 XML Schema 與 XML Storage 進行後續溝通的動作，Content translation gateway 會將 database designer 制訂好的 data capture template 轉換成 XML Schema，當作 XML Storage 驗證 XML 資料的依據。
- XML 及 XML Schema 資料的上傳：當 information provider 輸入的資料變成 XML 格式之後，系統會將這些 XML 資料轉換成符合 XML Storage 溝通協定的格式，透過 XML Storage Client Interface 上傳到 XML Storage 儲存。系統轉換完成的 XML Schema 也是經由這種方式上傳到 XML Storage，當作 XML 資料格式的定義。

四、相關研究

Apache Cocoon 是 Apache XML 計畫的一部份，預設支援 Apache Xerces 和 Apache Xalan，目前已經發展到了第二代，是一個完全以 Java 為主的 XML 排版系統。Cocoon 的架構是以 Java Servlet 為基礎，可以接納任何遵守標準的 XML parser。

Cocoon 計畫的目標是著重在資料的產生、展現和使用的整個流程，簡單的想法可以想像一個新聞的網站，新聞的內容是由專業的記者報導撰寫，網站的對外展現樣式則交由網站美工設計師來處理，新聞整篇報導的邏輯則是由編輯部總編來負責，透過 Cocoon 把內容、樣式和邏輯三者完美的區隔開來，讓一份文件可以同時由不同的人員手中獨立分工完成，提升系統出版的效率。

Cocoon 內部使用了 Content 和 Style 分開設計的觀念，更進一步地可以根據 client 端瀏覽器的不同，動態地搭配不同的 XSL 輸出各種不同的樣式，這個觀念對資料的展現相當實用。由於 Cocoon PI 對我們的系統設計並不完全適用，因此我們未採用 Cocoon 的作法，而是藉由 XSL Taglib 的技術，實現了資料和版面配置分開設計的觀念，使得同一份資料可以用不同的外貌呈現出來，或是多份相同格式的資料也可以套用同一份 XSL，讓多份資料有一致的展現標準。

RedDot CMS(Content Management Server) 是一套 content management system，主要目的在於幫助企業內部網站資料的內容管理，隨著電子商務網站的擴張，為了解決網站內容建構與維護的種種問題，網站發展工具也趨於更完善的維護工具，由於 RedDot CMS 能讓使用者達到資料內容的建立、出版和存取的目的，使得內容管理的過程更顯方便。RedDot server 和 browser-based 的使用者相互之間的關係，包含了資料內容的建立、出版和存取。RedDot CMS 提供 Multi-level Users 的機制，可有效區分工作的權限，這點值得我們的系統學習參考，未來可當成系統下一步發展的參考。

五、結論

Internet 的興盛與 Web 的蓬勃發展主要是由於網路上有許多人的參與，每個人都可以提供自己的資訊，也可以收集到其他人貢獻的資訊，XML 是更適合資訊收集以及重用性更高的文件格式，本文針對 XML 資料的需求，設計了一套數位內容開發工作平台，而在這個工作環境中，XML 技術對使用者而言是透明的 (transparent)，藉著本系統所提供的資料輸入工具，讓資訊的提供者可以方便地輸入資料，並

由系統輔助產生 XML 文件，透過 XML Storage 可以有效率地儲存 XML 資料，同時提供了完整的索引及搜尋功能，使用者也因此可以更快且精確地取得所需的資訊。在累積了大量的 XML 文件之後，我們便可以利用既有的 XML 文件提供其他更多整合性的服務給一般使用者來使用。每一使用者擁有個人的 Web 工作目錄，具有建立、刪除、編輯、查詢、上傳等功能，透過瀏覽器，使用者可以輕鬆地維護資料，而且使用者隨時可以瀏覽其他人分享的文件與檔案。系統也提供動態的文件展現工具，除了內定的展現樣式，系統也提供自動產生好的程式碼供使用者下載，使用者只需要照著這個樣版稍微修改，便可以指定自己設計的樣版為展現的依據，滿足使用者對於畫面展現不同的需求。

六、參考文獻

- [1] 江牧寰、蔡尚榮，“An XML-based Information Server – Tools for Data Importing”，*DSL Lab, NCKU*, June 2001.
- [2] 李嘉銘、蔡尚榮，“An XML-based Information Server – The Storage System”，*DSL Lab, NCKU*, June 2001.
- [3] Arnold-Moore, T.; Fuller, M.; Kent, A.; Sacks-Davis, R.; Sharman, N.; “Architecture of a content management server for XML document applications”，*Web Information Systems Engineering, 2000. Proceedings of the First International Conference on*, Volume: 1, 19-21 June 2000, pp. 97 -108
- [4] Jerry G. Wang, “Integrated Web Publishing Framework for the Java Platform “，*JavaOne*, <http://servlet.java.sun.com/javaone/conf/bofs/2054/google-sf2001.jsp>, June 2001.
- [5] Nianjun Zhou, Dikran Meliksetian, Louis Weitzman, Sara Elo Dean, Jeff Milton, Peter Davis, Jessica Wu, “XML Content Management: Challenges and Solutions”，*XML Europe 2001*, May 2001.
- [6] Petr Kroha、Lars Gemeinhardt, “Using XML in a Web-Oriented Information System“，*Proceedings. 12th International Workshop on*, 2001, pp. 217 -221
- [7] Takeshi Sannomiya, Toshiyuki Amagasa, Masatoshi Yoshikawa, Shunsuke Uemura, “A Framework for Sharing Personal Annotations on Web Resources using XML”，*Workshop on Information Technology for Virtual Enterprises (ITVE)*, pp. 40-48, January 2001.
- [8] Tsuda, I.; Uchino, K.; Matsui, I.; “WorkWare: WWW-based chronological document organizer”，*Computer Human*

- Interaction, 1998. *Proceedings. 3rd Asia Pacific*, 15-17 July 1998, pp. 380-385
- [9] W3C, "XML Path Language (XPath) Version 1.0", <http://www.w3.org/TR/xpath>, November 1999.
- [10] Yuriyama, M.; Nakamura, H., "Filtering contents by efficient evaluation of XPath expressions", *Applications and the Internet, 2003. Proceedings. 2003 Symposium on*, 27-31 Jan. 2003, pp. 148 -155