RRPF89091779 (6 .P)

行政院國家科學委員會專題研究計畫成果報告

整合式 XML 文件管理系統及其在 Web 出版之應用(I)

Integrated XML Document Management System and its Application to the Web Publishing (I)

計畫編號：NSC89-2413-H-032-017
執行期限：89 年 8 月 1 日至 90 年 7 月 31 日

主持人：林信成　　執行機構：淡江大學資圖系

壹、中文摘要

HTML 是一種簡化版的標注語言，也是建立超文件的標準規範[1]。由於具備規格開放、易學易用、語法簡潔明瞭等特色，加上獨特的超連結，HTML 的確為 Internet 上龐大的數位資料與電子文件，提供了一條有效的整合之道。但是隨著資訊科技的發展，Web 應用愈來愈廣泛，HTML 的弱點也愈來愈明顯。其中最嚴重的，便是 HTML 只能稱得上是一種資料的「展示」語言，擅長版面編排而欠缺內容語意，所以雖然適合人類閱覽但卻不利於電腦理解；其次，HTML 的標籤集是固定的、不可擴展的，無法應付多樣化的應用。這些缺點在電子出版、電子商務、遠距教學、電子圖書館等全新領域急速發展，並期望 Web 朝向自動化、智慧化目標邁進的同時，遂成了 Web 發展的一大隱憂。

有鑑於此，人們開始著手研究改進 HTML 的方法，XML（eXtensible Markup Language，可擴展標注語言）便是在這樣的背景下產生的。XML 具有可擴展性、高度結構化和良好的資料組織能力，能夠有效的表達網路上各種知識，為資料的交換和處理提供新的機制，一般預料，XML 將成為下一代 Web 的整合技術。若將 HTML 比擬為網路的第一波革命，則 XML 極可能繼 HTML 之後為網路帶來第二波革命性的改變，促使網路從資訊處理階段跨越到知識管理階段，並將在電子出版、電子商務、電子圖書館、電子資料交換、遠距教學等領域展現其強大的應用潛能。眾多的特點使得 XML 成為一個強勢語言，並迅速獲得各界的支持及響應。

關鍵詞：XML、HTML、電子出版、電子文件

貳、緣由與目的

1996 年 7 月「XML 工作小組」（XML Working Group）在 W3C（World Wide Web Consortium，全球資訊網協會）的贊助下成立[2]，當年 11 月提交 XML 初稿，並於 1998 年 1 月 10 日正式通過 XML 1.0 規範，成為 W3C 的一個建議標準（Recommendation）。由於 XML 具有可擴展性、結構性、自我描述性，並採用資料和樣式分離原則，使其在資料的管理、交換上擁有極為卓越之性能。XML 和 HTML 一樣都是從 SGML 演變而來的，只不過 HTML 是 SGML 的一個應用語言（Application），而 XML 卻是 SGML 的一個精簡子集（Subset）。XML 將 SGML 去蕪存菁，捨棄約百分之二十複雜罕用的部份，承襲了其他百分之八十的特點，是以具備了 SGML 所沒有的簡易性與靈活性，又有著 HTML 所欠缺的擴展性與結構性。因此，稱 XML 為主導「第二代 Web」（Second-Generation Web）的重要技術實不為過[3]。

不過，XML 並不是被發展出來取代 HTML 的，而是用以彌補其不足之處。XML 相較於 HTML 至少有以下幾個重要的差異：

(1) XML 文件的作者可以自訂標籤（Tags）和屬性（Attribute），HTML 則否。

(2) XML 是屬於一般用途（General Purpose）的標注語言，而 HTML 則是一種特殊用途（Special Purpose）的標注語言。換言之，XML 是一種元語言（Meta-Language），可以用以生成其他語言，HTML 則否。

(3) XML 著重於文件的結構，而 HTML 則擅長於文件的表現。

(4) XML 文件的作者可以選擇性的利用 DTD 或 XML 綱要（XML Schema）來確認文

件的有效性，HTML 則否。

依據 XML 的特性，可歸納出以 XML 為核心技術的新一代 Web 出版將具備如下之特色：

- 電子文件具備自我描述性
- 電子文件更能有效整合
- 電子文件更具結構性
- 電子文件具備內容和外觀分離原則
- 標注語言具備多樣性及可擴展性

茲分述如下。

（一）電子文件具備自我描述性

XML 的標籤可根據不同的用途來定義，因此在語意層次上具備一定程度的自我描述（Self-Description）特性，這對於提昇處理程式解讀文件內容的能力與進行自動處理的效率有著莫大的幫助。

（二）電子文件更能有效整合

透過不同的協定轉換，各種不同格式的資料可以轉成 XML 格式，使得 XML 在文件整合（Document Integration）方面，扮演了一個通用集成器（Universal Hub）的角色[4]，而 XML 的名稱領域、XLink 等正是文件整合不可或缺的重要技術。

（三）電子文件更具結構性

XML 具有嚴格的規範以適應廣泛的應用，因而造就了 XML 文件強烈的結構性，在資料處理和機器理解方面具備了先天的優勢，這也是促使 XML 迅速成為重要機讀格式的主因之一。

（四）電子文件的內容和外觀分離原則

XML 強調的是如何以適當的結構來組織資料，對於外在的表現則必須透過其他顯示機制才能達成，這就是 XML 文件的資料、樣式（即內在、外貌）分離原則。這使得文件作者只要專注於內容的撰寫，而將顯示資訊的任務交由版面設計者或使用者，依據不同的需求來展現。如此一來，同一份文件或資料，將可在不同的場合呈現出不同的風貌。

一般而言，展示 XML 文件最簡便的方式是透過樣式表（Stylesheet），一份文件可以使用不同的樣式表而呈現出不同的外觀。CSS（Cascading Style Sheets 層級樣式表）和 XSL（eXtensible Stylesheet Language 延伸樣式語言）即是兩種常用的樣式表語言；此外，XML 文件出版者還可以透過分解 XML 結構樹的方式，將 XML 文件呈現在讀者眼前。

（五）標注語言具備多樣性及可擴展性

XML 既可視為是一種在 Web 上建立結構化文件和資料的通用格式（Universal Format），也可視為發展其他應用語言的低階語法（Low-Level Syntax）[5]，這就是 XML 被稱為 Meta-Language 的原因，也是 XML 最引以為傲的可擴展性（Extensibility）。目前已有許多經由 XML 所定義並使用於不同領域的應用語言，例如應用於網頁出版的 XHTML（eXtensible HyperText Markup Language）[6]、應用於數學方面的 MathML（Mathematical Markup Language）[7]、應用於向量圖的 SVG（Scalable Vector Graphics）[8]、應用於多媒體領域的 SMIL（Synchronized Multimedia Integration Language）[9]、應用於描述網路資源的 RDF（Resource Description Framework）[10]、應用於網路推播頻道的 CDF（Channel Definition Format）[11]... 等，展現了 XML 無限擴展的能力。

綜上所述，XML 不但能有效解決目前網路上電子文件的亂象，更有助於開創電子文件自動交換與傳遞的新契機。因此，有必要建立一個適合一般使用者使用的「整合性 XML 文件管理系統」（Integrated XML Document Management System）。藉由系統分析過程，將此一整合系統劃分為三個子系統。分別是：(1) 編輯子系統；(2) 核心子系統；(3) 出版子系統。其主要任務是基於 XML 規範，對電子文件的結構、內容、表現三要素，進行有效率的管理以利於展示、查詢、編排、維護等加值處理，藉由各個子系統中的編輯模組、剖析及轉換模組、檢視模組、排版模組、發行模組 ... 等，可以建立一個適用於網路環境的電子文件整合出版系統。

系統方塊圖如圖 1 所示，其中的 Web 子系統是供讀者使用的。

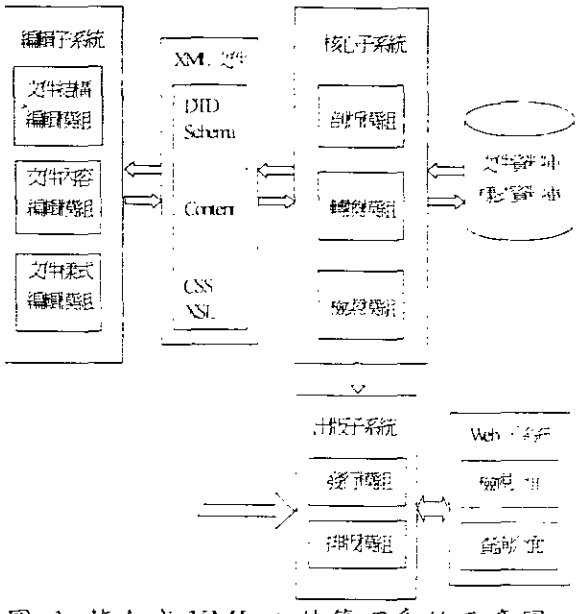

圖 1 整合式 XML 文件管理系統示意圖

參、結果與討論

在經過一年的時間研究之後，現階段已完成：

（1）收集相關文獻，探討 Web 出版之現況及缺失；

（2）針對 HTML 及 XML 之特色深入研究，瞭解兩者的異同處；

（3）基於 XML 之件管理之需求，進行系統分析，規畫各模組之功能；

（4）收集技術文件，研討如何以「文件物件模型」（DOM）實現本系統。

（5）進行系統設計，完成剖析模組之開發；

（6）進行系統測試。

要完全發揮 XML 之所長仍需要電腦程式的配合。XML 文件是一種基於文字模式的開放規格，雖然具備嚴謹的樹狀資料結構，但若要以電腦程式對其文件內容進行剖析、處理和操縱（Manipulation），仍需要藉助特定的演算法才行。幸好，樹（Tree）是一種極為普遍的資料結構，已經有許多現成的演算法被發展出來。[12]因此，任何應用程式都可以很容易的利用樹狀結構的演算法，對 XML 文件內容進行諸如讀取、修改、刪除、新增、搬移或走訪等動作。然而，若是能有一個介面標準直接支援 XML 樹狀結構的存取，將可簡化並縮短程式開發之流程。由 W3C 所推動的「文件物件模型」（Document Object Model，簡稱 DOM）便是一個這樣的介面標準。[13]不過，DOM 並非是針對 XML 量身定做的，它適用於 HTML 及 XML 文件。由於 DOM 是一個語言中立（Language-Independent）的應用程式介面（Application Programming Interface，簡稱 API），目的在於建立一個跨平台的文件操作環境，因此，DOM 可以在不同的作業系統中，使用任何程式語言加以實現。在 DOM 的規範中，定義了文件的邏輯結構以及存取、處理、操縱文件的方法。藉由 DOM，應用程式可以輕易的建立文件，可以在文件結構中來回穿梭，可以新增、修改或刪除文件中的元素或內容。

DOM 規範依其複雜層度分為三個層級：Level 1、Level 2 和 Level 3。

（一）DOM Level 1

DOM 層級一（DOM Level 1）規格於 1998 年 10 月正式成為 W3C 的一個建議標準（Recommendation），[14]其焦點集中在解決文件剖析的核心問題上，定義了 HTML 和 XML 文件的結構模型，提供文件巡航（Navigation）和操縱（Manipulation）等功能，使得應用程式得以非常容易的存取文件內容（新增、刪除、編修內容、編修屬性及文件類型 ...），於是程式設計師便可藉以發展出適用於各種瀏覽器、伺服器及工作平台的應用程式；程式設計師或許會使用不同的程式語言（如 C/C++、Java、JavaScript、VB、VBScript ... 等），但不需要去改變程式模型。[15]

（二）DOM Level 2

DOM 層級二（DOM Level 2）規格則於 2000 年 11 月 13 日成為建議標準，其規格比 Level 1 複雜許多，除了加強定義如何操縱及處理文件結構及內容的核心（Core）介面外[16]，尚包含了文件的外觀（Views）[17]、樣式表物件模型（Style sheet object model）[18]、事件模型（Event model）[19]、走訪範圍（Traversal Range）[20]等規範，以便應用程式可以很容易的處理附加的樣式表、各種不同的外觀、事件以及允許在文件中四處走訪。

（三）DOM Level 3

DOM 層級三（DOM Level 3）規格其內容主要包含了下列各項目：[21]

- 擴展 DOM 層級二之物件模型（Object Model）：允許使用者存取鍵盤事件；增加定義群組事件的能力。
- 新增內容模型（Content Model）和有

效性（Validation）：一個可以直接存取和修改文件內容之物件模型。

● 提供載入（Load）和儲存（Save）介面：提供直接載入 XML 原始文件至 DOM 結構以及直接將 DOM 結構儲存為 XML 文件的程式介面。

● 內嵌式（Embedded）文件物件模型：目前 Web 的發展趨勢正朝向以混合式標注語言方式寫作文件，例如將 SVG 片段內嵌在 XHTML 文件中，這使得 DOM 必須面對新的挑戰，因為 DOM API 需要具備從混合的詞彙集中剖析文件結構與內容的能力。

● 適應 XML 核心功能改變的能力：DOM 既然是 XML 文件的 API，一旦 XML 發展出附加功能（如 namespaces、XML Base 等），DOM API 也應能夠隨之調整其對應模型。

● XPath DOM：使用 XPath 來查詢 DOM 結構樹的解決方案。

不論 Level 1、Level 2 或 Level 3，實際上其根基為 DOM 核心規格（DOM Core Specification），其中定義了一個階層式物件模型。在此一物件模型中，不論是文件、元素、屬性、實體、文字或其他在 XML 文件中出現的單元，都被視為一個一個的節點（Node），而所有節點則被組織成一個樹狀結構（Tree Structure）。[22]

XML 文件、剖析器、DOM 物件樹和應用程式之間的關係如圖 2 所示，從圖中我們可以清楚的看到：XML 文件經由剖析器（Parser）剖析之後，形成一個 DOM 樹狀結構，此一樹狀結構將文件中的元素、屬性、資料 ... 等以階層式節點方式加以儲存，應用程式則藉由存取樹中的節點而讀出資料或寫入資料。

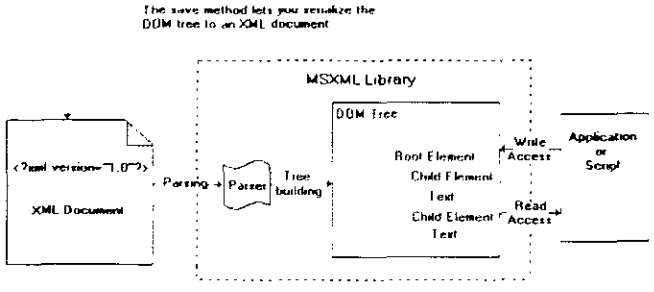

圖 2 XML 文件之剖析與存取

資料來源：http://msdn.microsoft.com/

建立好的 XML 文件，必須先檢測其語意或結構上有無錯誤才能進行爾後的處理。因此，剖析模組的任務便是檢測文件的良好性及有效性，並將發現的錯誤告知使用者，以便於除錯及修改。在設計系統的過程當中，本文以 DOM 為基礎，在視窗環境下開發一個具備圖形化使用者介面（Graphic User Interface，GUI）的剖析模組。

本剖析模組是以樹狀結構來顯示 XML 文件，並將各階層以加、減表示，如圖 3 所示。

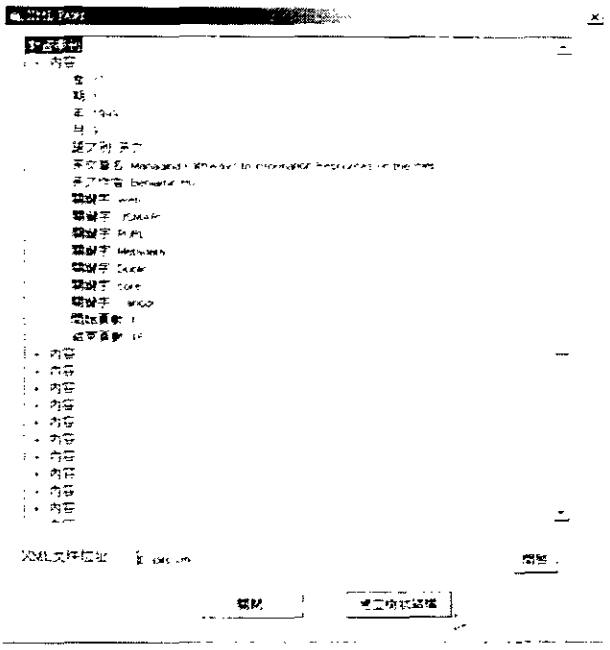

圖 3 文件剖析結果

若是剖析到錯誤的 XML 文件時，就無法建立樹狀結構，並且出現一警告視窗，如圖 4 所示。

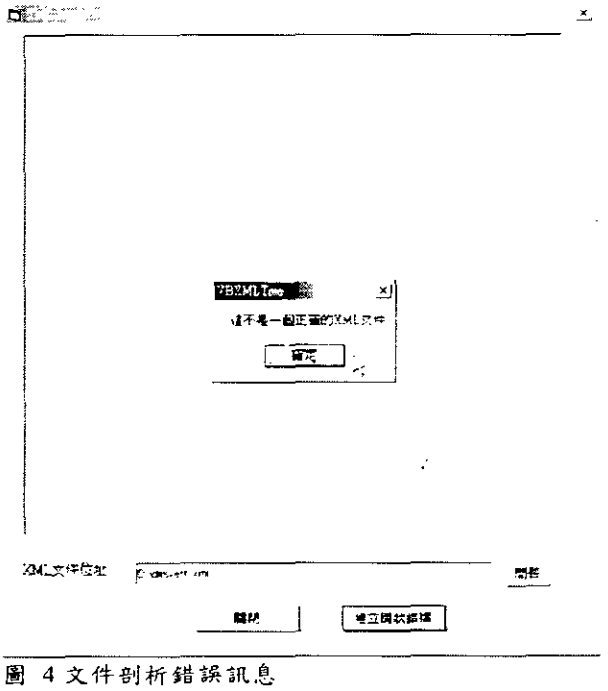

圖 4 文件剖析錯誤訊息

肆、計畫成果自評

在進行系統設計時，就困難的地方莫過於建構一個通用的文件管理系統給予使用者，這樣一來，需考慮的層面就非常的廣泛，如 DTD、XML Schema、XSL 等，要能精確的與後端資料庫相互對應，並與當初使用者所建置的結構相同，是系統設計一大考驗。諸多考量之下，先行建置剖析模組，讓文件的驗證有所標準，方便日後其它模組的處理。

經過一年多的研究努力，作者已經著手將在本計畫所獲得的初步成果整理成期刊論文共七篇[23-29]，並發表在多本學術期刊上。

這一年的研究過程中，發現以一個通用型文件管理系統來做為目標，並不是個最好的呈現方式，所以在第二年的研究計畫中，將以圖書館利用教育做為系統主題，其資料為欲管理之文件，以 ASP 開發一個網路社群，提供圖書館利用教育之課程、討論區等，供使用者使用，並將其文件以 XML 的結構儲存、連送及管理，具體以一個主題來呈現 XML 文件管理系統，完成本研究計畫。

伍、註釋

[1] "HyperText Markup Language". available at <http://www.w3.org/pub/MarkUp/>.

[2] 「XML 工作小組」最初稱為「SGML 編審委員會」（SGML Editorial Review Board）。

[3] Jon Bosak and Tim Bray. "XML and the Second-Generation Web". Scientific American. May 1999. also available at <http://www.sciam.com/1999/0599issue/0599bosak.html>.

[4] Jon Bosak, "XML, Java, and the future of the Web". available at <http://metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.html>, 1997.3.10.

[5] "Extensible Markup Language (XML) Activity". available at <http://www.w3.org/XML/Activity.html>.

[6] "XHTML 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0". W3C Recommendation 26 January 2000. avalible at <http://www.w3.org/TR/xhtml1/>.

[7] "Mathematical Markup Language (MathML)". available at <http://www.w3.org/Math/>.

[8] "W3C Scalable Vector Graphics (SVG)". available at <http://www.w3.org/Graphics/SVG/>.

[9] "Synchronized Multimedia". available at <http://www.w3.org/AudioVideo/#SMIL>.

[10] "Resource Description Framework (RDF)". available at <http://www.w3.org/RDF/>.

[11] "Channel Definition Format (CDF)". available at <http://www.w3.org/TR/NOTE-CDFsubmit.html>.

[12] Ellis Horowitz. "Fundamentals of data structures in Pascal." FREEMAN: San Francisco. 1989.

[13] W3C Recommendation. "Document Object Model (DOM) Level 1 Specification Version 1.0". 1 October 1998. available at <http://www.w3.org/TR/REC-DOM-Level-1>.

[14] 同註 13。

[15] W3C. "Document Object Model (DOM) Activity Statement." Last modified date: 2000/12/14. available at <http://www.w3.org/DOM/Activity>.

[16] W3C Recommendation. "Document Object Model (DOM) Level 2 Core Specification Version 1.0." 13 November 2000. available at <http://www.w3.org/TR/DOM-Level-2-Core >.

[17] W3C Recommendation. "Document Object Model (DOM) Level 2 Views Specification Version 1.0." 13 November 2000. available at <http://www.w3.org/TR/DOM-Level-2-Views

>.
[18] W3C Recommendation. "Document Object Model (DOM) Level 2 Styles Specification Version 1.0." 13 November 2000. available at <http://www.w3.org/TR/DOM-Level-2-Styles>
.
[19] W3C Recommendation. "Document Object Model (DOM) Level 2 Events Specification Version 1.0." 13 November 2000. available at <http://www.w3.org/TR/DOM-Level-2-Events >.
[20] W3C Recommendation. "Document Object Model (DOM) Level 2 Traversal-Range Specification Version 1.0." 13 November 2000. available at <http://www.w3.org/TR/DOM-Level-2-Traversal-Range >.
[21] 同註 15。
[22] 同註 13。
[23]林信成，"文件物件模型及其在 XML 文件處理之應用"，教育資料與圖書館學，第三十八卷，第四期，頁 407-438，民 90 年 6 月。
[24]林信成、龔裕民，"XML 與電子文件展示技術之探討"，圖書與資訊學刊，第三十七卷，頁 58-78，民 90 年 5 月。
[25]林信成，"軟性計算理論與知識處理技術之研究"，教育資料與圖書館學，第三十八卷，第二期，頁 148-173，民 89 年 12 月。
[26]林信成，"XML 在電子出版之應用 --XHTML、SMIL、MathML 與 SVG 初探"，國家圖書館館刊、第 2 期，頁 157-172，民 89 年 12 月。
[27]林信成，"基於 XML 之新一代 Web 技術及其在電子出版之應用"，佛教圖書館館訊，第 23 期，頁 18-40，民 89 年 9 月。
[28]林信成，"基於 XML 之分散式模糊知識管理系統模式"，即將刊載於教育資料與圖書館學，第三十七卷，第四期，民 89 年。
[29]林信成，"XML 相關技術與 Web 出版趨勢之研究"，教育資料與圖書館學、第三十七卷，第二期，頁 184-210，民 88 年 12 月。