圖書與資訊學刊
37(May '01)58-78
ISSN 1023-2125

XML 與電子文件展示技術之探討

Study on the XML and Presentation Technologies for Electronic Document

林 信 成
Sinn-cheng Lin
助理教授 Assistant Professor
E-mail：candme@ms13.hinet.net

龔 裕 民
Yu-min Gong
研究生 Graduate Student
E-mail：st02@ms17.hinet.net

淡江大學資訊與圖書館學系
Department of Educational Media and Library Sciences, Tamkang University

【摘要 Abstract】

XML 由於具備結構化、擴充性、跨平台、內容外觀分離等特性，很快地成為一種公認的文件標準，由 XML 所延伸的眾多應用，如 MathML、SMIL、SVG、XHTML 等，也逐漸普及於多種專業用途上。未來 XML 更可能結合攜帶式行動設備存取網路資訊。本文就目前最具代表性的幾種電子文件技術，如 RTF、Postscript、PDF、SGML、HTML 等，加以分析、探討其優缺點及特性，並與 XML 的文件展示技術比較，藉此闡述 XML 的特質及應用潛力，進一步探究未來電子文件展示技術的可能發展趨勢。

The main features of XML include (1) rigorous structure and extensibility (2) separated processing of content, structure schema, and display (3) platform-independency, high portability (4) easy applications in e-commerce (5) a Meta-language for other applications. Nowadays, a lot of applications based on XML were developed in some professional fields, e.g., MathML for scientific publication, SMIL for multimedia authoring, SVG for vector diagram drawing, and XHTML for Web page publication. In the future, XML incorporated with mobile devices such as PDA's, palm-top computers, in-car computers, cellular phones, and pagers will construct a mobile information service environment which can provide a quick access to the Internet anywhere and anytime. In this study, some representative technologies of electronic documents, such as RTF, postscript, PDF, SGML and HTML, were discussed to reveal their merits and shortcomings. In comparing XML with the previous technologies, this study also explored its characteristics and potentials, with an attempt to trace out a possible trend in presentation technologies of electronic publications.

關鍵詞 Keyword

可擴展標注語言 可擴展超文件標注語言 同步多媒體整合語言 數學標注語言 可調整向量圖
Extensible Markup Language；Extensible Hypertext Markup Language；Synchronized Multimedia Integration Language；Mathematical Markup Language；Scalable Vector Graphics

壹、緒論

一、研究動機

自 1970 年代初期美國的「古騰堡計畫」開始，人們便嘗試著透過網路科技來傳遞文件，「電子文獻」(Electronic text 或簡稱 Etext)成爲一種新興的文件型式，殆至 Internet 及 WWW 盛行之後，電子文獻無論在數量、型式上均產生極大的質變，使用者大量在文獻取得的深度（精確率）及廣度（回現率）上著力，以求取網路資源的最有效運用，而傳統電子出版的概念也正式被引用在 Web 環境上，網路出版的方式，成爲萬眾注目的焦點。

電子出版文件不同於一般文件之處，在於其除了必須便於讀者閱讀之外，尚須具備適合機器閱讀、利於查詢檢索、文件內容可重新排版與再利用等等的特質，尤其在網路的作業環境中，電子文件必須克服各種不同作業平台的瀏覽方式、顯示及排版軟體的差異，必須適時提供一些網路環境獨有的功能（如：超連結、互動程式操作、後端資料庫存取...等），再再都考驗著文件技術的豐富性及穩定性。在過去的數十年中，追求一種「標準文件格式技術」，一直是人們所引頸企盼而戮力以赴的，無論是 RTF、Postscript、PDF、SGML、HTML 乃至 XML 標準，都在某種程度上引起人們的高度注意，其中尤以 XML（eXtensible Markup Language，可擴展標注語言）在近二年中的表現最爲亮麗。

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(註 1)，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，成爲 W3C 的一個建議標準（Recommendation）。(註 2)由於其具備：(1) 結構嚴謹、具擴充性；(2) 採取資料內容、結構綱要、展示呈現分離處理的方式；(3) 跨平台、高可攜性；(4) 適合發展電子商務；(5) 可作爲發展其它應用的元語言（Meta-Language），因此很快地被接受成爲一種公認的網路文件技術。

此外，由 XML 所延伸而出的眾多應用，也逐漸使用於多種專業用途之上，如數理領域上的 MathML、多媒體應用上的 SMIL、製作向量圖的 SVG、以及與 HTML 結合的 XHTML，都是以 XML 爲核心所發展出的特殊用途語言。又基於未來性的考量，XML 更可結合攜帶型行動設備，如個人數位助理器、掌上型電腦、車用電腦、行動電話、呼叫器... 等，來存取網路資訊、瀏覽網頁內容。因此，稱 XML 爲主導「第二代 Web」（Second-Generation Web）的重要技術實不爲過。(註 3)

二、研究方法

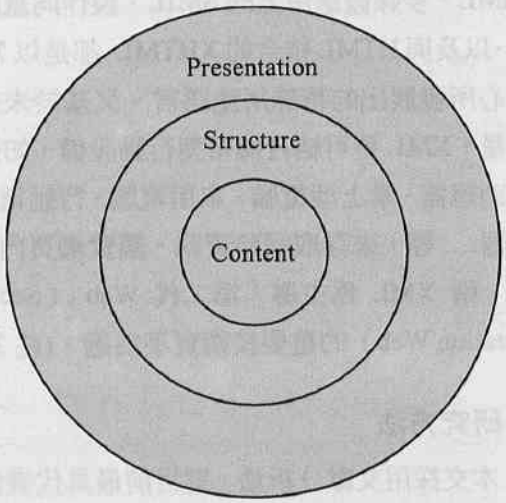
本文採用文獻分析法，就目前最具代表性的幾種電子文件技術（RTF、Postscript、PDF、SGML、HTML）加以分析，探討各別的優缺點及特性，並與 XML 的文件技術進行比較，闡述 XML 的文件特質及相關技術的發展，以進一步探究未來電子出版在文件展示技術上的可能發展。

三、研究範圍

電子文件是電子出版的核心，電子文件所必備的要素眾多，其中最重要的當屬「內容」（Content）、「結構」（Structure）和「外觀」（Presentation）三者：

1. 內容：指文件的資料（Data）本身，也就是文字、圖表等內涵部分。
2. 結構：指文件中有關資料的描述部份，如題名、作者、章節、段落等。
3. 外觀：指文件所呈現出來的外表、樣式、版面編排等。

若將電子文件與人體作一類比，則結構就如同人的骨架，內容則相當於肉體，而外觀就形同外貌，三者相輔相成，形成一個完整的資訊描述體系，如圖一所示，因此本文特將此三者合稱爲「電子文件三要素」。



圖一：電子文件三要素

一般而言，如果電子出版的目的僅在於如何將電子文件順利呈現在讀者面前供其閱覽，則只要將文件的「資料」以適當的「外觀」來展現即可，不太需要考慮到文件的「結構」問題，就此一層次而言，HTML 已經相當足夠。但是，如果要更有效的管理、檢索、交換 Web 上呈指數成長的電子文件，則非得加強其結構性不可，XML 的發展正好爲此提供一個可行之道。XML 的資訊組織能力，的確爲文件的結構化帶來一線生機，這已在筆者稍早的文章中探討過(註 4)，因此不在本文研究範圍之內。本文從另一觀點出發，深入探討 XML 在文件外觀上的展示能力，並與現有的文件展示技術進行比較，以探討 XML 的發展潛力。

貳、網路電子文件的展示

一、常見的電子文件格式與展示技術

在傳統的電子出版系統中，設計者經常要將處理過的文件依照其不同的特性而儲存成不同的檔案格式，而由於電子文件中經常會出現表格、圖像、多媒體元件、特殊效果，因此便有各種使用方便、功能完善的電子文件格式。例如：文字導向型文件常見的 TXT、DOC（Microsoft Word 的格式）；影像導向型文件中的 BMP、GIF、TIFF、JPEG、PNG；多媒體影音導向型文件的 WAV、MIDI、MP3、AVI、MPEG 等等，可謂琳瑯滿目，種類眾多。

在 Internet 掘起之後，新型態的出版方式產生，檔案的存在價值，除了必須考慮到資料的儲存與印刷的能力外，同時還必需要兼顧到四點因素：(1) 方便於在網路上進行瀏覽；(2) 能達到與網路使用者互動的效果；(3) 資料傳輸的效率性；(4) 網路文件的管理。因此，在文件的內容、結構、及外觀各方面，都產生了革命性的改變，早已不是傳統文件所能比擬，諸如此類的特殊文件型式，我們姑且稱呼其爲「網路化文件」或「Web 文件」(Webdoc)。

「Web 文件」因爲具有其特殊的性質，因此在檔案的處理及呈現方式上，也與一般檔案有所不同，以下便以目前常見的「Web 文件」展示技術 RTF、Postscript、PDF、SGML、HTML 等，就其性質加以探討，以便與基於 XML 的展示技術有所比較。

1. RTF

RTF（Rich Text Format）格式，就全名的意思而言，是一種「豐富的文件格式」，最早被用在 Windows 系列作業平台上，適用於 WordPad 或 Microsoft Word 文書處理軟體上的一種特定用途

標注語言(註 5)，後來被廣泛使用在網路文件的儲存上。由於 RFT 採用了結構與呈現分離的方式，非常適用於使用在不同的網路平台或瀏覽器，而且只要使用一般的文書處理工具，即可輕易地製作 RTF 檔案，因此很快便成爲一種受歡迎的網路文件格式。

由於在各種不同的作業平台間，存在著各式各樣不同的電子文件處理軟體，這些軟體功能各異，因而文件格式也經常隨之不同，這對於不同工作平台上的使用者而言，無疑是一大困擾。RTF 提供了一個文字及圖片的編碼轉換格式，使得文字及圖片可以在各種不同的應用軟體之間進行轉換，使得存放在 MS-DOS、MS Windows、OS/2 及 Apple Macintosh 環境之下各種不同應用軟體的電子文件，皆可以方便地共享。(註 6)

RTF 可以包含許多文字、控制字（Control word）、控制符號（Control symbols）、圖片、群組（Groups）等等，爲了轉換的便利性，RTF 也允許使用 7 位元的 ASCII 碼（絕大多數在 PC 或 Mac 機器上的電子文件處理軟體都支援 8 位元的 ASCII 碼），使文件的相容性更高。而 RTF 文件的閱讀器（RTF Reader）在解讀 RTF 文件時，只要依循固定的程序，便可以對來自不同應用軟體的 RTF 文件進行解讀，這些程序爲：(1) 先從平舖直敘的文句中分辨出控制字的位置；(2) 以控制資訊加以取代；(3) 區分文字、段落、表格、圖片；(4) 訂正並適當地插入文字到文件之中。

RTF 的原始碼並不易閱讀及理解，例如，以下一句簡單的文句：

電 子 出 版 的 未 來

如果我們檢視此一 RTF 文件的原始碼，會得到：

```
{\rtf1\ansi\ansicpg950\deff0\deflang1033\deflangfe1028{\fonttbl{\f0\fmodern\fprq6\fcharset136
\'b7\'73\'b2\'d3\'a9\'fa\'c5\'e9;}}
\viewkind4\uc1\pard\lang1028\f0\fs20\'b9\'71\'a4\'6c\'a5\'58\'aa\'a9\'aa\'ba\'a5\'bc\'a8\'d3\par}
```

上述的 RTF 文件，是由許多控制字及控制符號組合而成，如果對照本文的意思「電子出版的未來」，似乎是完全不相干，很難想像這事實上是由同一份文件而來。

2. Postscript

Postscript 是 1985 年由 Adobe 公司推出，是一種頁面描述語言，主要是用來描述文字及圖形，Postscript 經常被使用於排版及印刷工作，將電子文件儲存成 Postscript 最大的好處是具有跨平台的特質，讀者只要有 Postscript 的閱讀器，不論該文件是在何種作業平台下產生，都可以正常閱讀。Postscript 在 UNIX/Linux、MAC OS、Windows 的環境均可支援，因此它有具備成爲網路上標準文件的條件。

EPS（Encapsulated Postscript Format）是目前常使用的一種 Postscript 格式，使用文字指令來描述圖像，可以將文件區分爲以圖像爲基礎（Pixed-based）以及以文字敘述爲基礎（Text-based）兩種方式，市面上知名的影像處理軟體如 Photoshop，其所支援的格式中便有 Pixed-based EPS；另一些著名的繪圖、排版軟體如 Illustrator 及 Freehand 等，所支援的格式則屬於 Text-based EPS。所謂 Text-based EPS，意指在檔案之中加入些許文字描述語言，以加強文件的內部控制，如：「文件預覽」、「檔案資料」等。另有許多排版軟體，如：PageMaker、QuarkXPress 等，均已支援 EPS 格式。

EPS 最大的好處是可以對文件內容任意的設

定、縮放、旋轉等，可以獲取良好的列印效果；而缺點則在於 EPS 檔比較佔空間。在網頁的瀏覽方面，在閱讀 EPS 文件時，也區分爲文字模式及圖形模式，文字的部份可以與 ASCII 碼相互轉換，而 HTML 的文件格式，也可以轉換爲 Postscript。或者可以在 EPS 中，加入轉換敘述，即可將 HTML 轉換爲 Postscript，如下：

```
require HTML::FormatPS;
   $html = parse_htmlfile("test.html");
   $formatter = new HTML::FormatPS
      FontFamily => 'Helvetica',
      PaperSize    => 'Letter';
   print $formatter->format($html);
```

Postscript 至今已發展至 Level II 階段，EPS 格式也可以採用 ASCII 碼或二元碼（Binary Code）方式儲存，ASCII 爲最原始的文字交換碼，發展最爲成熟，因此不論是 Postscript Level I 或 Postscript Level II 均可支援 ASCII 儲存，但二元碼卻具有儲存空間小，輸出速度快的優點。由於 Postscript 具備良好的文字及圖形顯示功能，除了被用來儲存文件之外，也經常被利用來作爲印表機的控制語言，一台可以支援 Postscript 的印表機，常意味著可以列印出高品質的文件，在近來的噴墨及雷射印表機中都已經被普遍的支援。(註 7)

雖然 Postscript 具有跨平台、可同時顯示文字圖形、可與 HTML 互相轉換等優點，但在使用上仍有不便之處，例如有些瀏覽器仍無法直接瀏覽或儲存 Postscript 檔案，必須使用一些轉換程式，如在 UNIX/Linux 中的 Mozilla Print Gidget(註 8) 或在 Windows 下的 GhostView...等。

3. PDF

PDF 全名是 Portable Document Format，是由 Adobe 公司所製訂，目的在於方便不同平台上的文件交換處理，由於 PDF 在電腦上或網路上對於資料的呈現，可以達到與紙本資料幾乎完全相同的效果，因此又被稱爲是一種「文同文件」。PDF 是電子出版中經常被使用到的一種電子文件儲存與展示技術，也由於它可以在文件傳遞的過程中始終保持原貌，因此在 Internet 上大受歡迎。

就一般文件而言，如果在某編排軟體中使用了某一種字型，那麼在開啓它的軟體中也必須具備這個相同的字型，才可以達可完全的展示效果，如果閱讀端沒有相對的軟體或字型，經常會使顯示效果大爲失真；但如果以 PDF 方式處理，無論文件在何種軟體下建立，也無論作者使用了何種字型甚至特殊效果，在閱讀端方面，只要具有 PDF 的閱讀器（例如 Adobe Acrobat Reader）就可以完全的一覽無遺，而不必擔心是否具有相對應的程式或字型，這對於網路上的讀者而言是十分重要的，因爲 Internet 本身即具備跨平台的特質，能夠解決文件閱讀的問題，無疑是網路讀者的一大福音。

PDF 文件在製作的概念上與 Postscript 文件十分地接近，具有跨平台、可同時顯示文字、圖形等的優點，當文件以 PDF 方式儲存時，已經將其中的各項元件予以格式化，文件在傳遞的過程中以壓縮的方式進行，有助於保持文件的完整性，同時它還具備下列優點：

(1) 高壓縮性

PDF 通常可以將文件壓縮至原來的數十到數百分之一，尤其是希望在網路上發行電子文件時，基於傳輸速度的考量，文件的容量大小會是一個重要的考量因素。

(2)可靠性

在 Postscript 中可能還是會含有一些特殊字體、特殊輸出設備的程式碼，而 PDF 則不含任何的可程式結構，因而更加地穩定，也更適合於電子資料型式的儲存或傳輸。

(3)分頁獨立

在 Postscript 中，文件的各頁間是相互關聯的，因此如果我們想要直接瀏覽某一頁必須先處理完前面所有的跳頁程序，而 PDF 則沒有這個限制，在瀏覽某頁時完全不需考慮到其它的頁數。

(4)即時列印

只要透過 PDF 的閱讀器，如 Adobe Acrobat Reader 或「文電通」，就可以在任意的電腦、任意的印表機上瀏覽或列印出文件。

(5)特殊註記

PDF 文件可以加上特殊的註記，如書籤、縮影、超鏈結（Hyperlink）、備註、媒體盒...等，這些註記可以有效地協助檔案管理工作，如：製作、發送、簽註、修正、完稿以及出版發行。

(6) 檔案保護

PDF 還可以設定密碼及保護，以防止網路的非法攔截及使用。

PDF 似乎爲 Web 出版帶來了一片榮景，不過 PDF 也有一些問題存在，例如 PDF 在製作過程中，將文件的內容加以壓縮及格式化，以達到降低容量及方便閱覽的目的，但這樣以輸出爲導向的處理方式，大大地破壞了文件的原始結構，導致 PDF 的功能只能以閱讀爲主，而無法加以編修或再利用，無論是 Adobe Acrobat Reader 或「文電通」都只是一種閱讀器（Reader）而不是編輯器(Editor)，這對於資源的使用上，可謂是一種浪費。

再則，雖然 PDF 具備極高的使用便利性，但畢竟不是一種國際性的通用標準（PDF 是 Adobe 公司的專利），固然在各種作業平台之上皆可以使用，但因爲不是標準文件格式，大多數的瀏覽器並未將之列爲預設的文件格式；如果一份網路文件在瀏覽器上無法瀏覽，而又沒有 PDF 的閱讀程式，將無法正常地閱讀文件，因此很難成爲一種完全受網路使用者接受的文件格式。

4. SGML

SGML 是「標準通用標注語言」(Standard Generalized Markup Language)的簡稱，於 1986 年制定，爲 ISO-8879 國際標準，具有跨平台、可攜性、結構化、移植性、重覆使用等特性；SGML 提供一種文件結構的規範，允許將文件的內容分割儲存在不同的地方，也就是將一份單純的文件予以「結構化」。

SGML 製訂了一種「文件型別定義」(Document Type Defination，簡稱 DTD)，主要是對文件進行結構的描述，只要是符合 DTD 的文件，都可以在不同的文件處理系統之間自由地進行交換。一份標準的 SGML 文件基本上由三個主要的部份所組成：

(1)結構（Structure）

SGML 文件以 DTD 爲核心，DTD 扮演著類似資料庫綱要(Database Schema)的角色，負責定義文件的結構。

(2)內容（Content）

內容是指資訊或文件的主體，文件中的主要資料部份放置於此。

(3)樣式（Style）

SGML 強調結構與內容之間的關係，但並不指定文件應該以什麼方式呈現，呈現的部份主要

是依據 ISO DSSSL(Document Style Semantics and Specification Language)的標準規範。

SGML 的主要功能爲：

(1)作爲不同文件處理系統之間，資料交換的標準。
(2)標示文件的邏輯結構，並指明文件中每一個元素的角色。
(3)用以檢測文件的完整性。
(4)可以作爲開發全文資料庫的標準。

SGML 是一種一般用途的標注語言(General Markup Language)，以標籤(Tags)來定義文件的展示方式，因此被稱爲「標注語言」;SGML 也是一種「元語言」(Meta Language)，經由 SGML 可以描述或產生另一種語言，它不是專爲某種特定用途而設計，因此也稱爲「上層語言」(註 9)，例如，主導網路文件的 HTML 便是經由 SGML 所定義出來的。

SGML 強烈地要求文件的結構性，DTD 明確地規範文件的結構，而應用程式也經由 DTD 來讀取文件內容；另外 SGML 還具備了許多的優點：

(1)可以增加作者的生產力，適合應用於網路出版。
(2)資訊可以再利用，延長資訊的生命週期。
(3)對資料的結構、內容、呈現皆有良好的控制方法。
(4)資料具有可移植性、可攜性、跨平台的特質。
(5)穩定性高，制定完成後便很少被修改。
(6)具擴充性，可以對各種不同領域定義新的語言。
(7)是經過國際認証的統一標準，放諸四海而皆準。

這些優點，使得 SGML 很快地便成爲網路上的文件標準，它設計週詳、功能龐大，對於網路文件大部份的使用都詳作規劃，但在網際網路的發展史上，SGML 卻不是促成 Internet 成功的主要推手，反倒是由 SGML 延伸出來的 HTML 成爲 Internet 普及化的最大功臣，這主要歸因於 SGML 過於複雜，學習不易，要了解它的元件（Elements）、屬性（Attributes）、實體（Entries）並不是一件容易的事，因此僅管 SGML 如此優秀，它的應用也始終環繞於政府機關、學術單位、企業行號等大型的機構，也唯有花費龐大的人力、精力來建置與維護，才能發揮 SGML 的完美效益。

基於上述因素，就網路出版的角度而言，SGML 並非理想的方式，畢竟網路出版的目標是要將文件普及化，透過網路的傳播，使文件的流通、重製、再利用都能達到便利的目的，這一點上 SGML 顯然不足。不過，由 SGML 延伸的 HTML 及 XML 卻扮演了決定性的重要角色。

5. HTML

HTML 是 "HyperText Markup Language" 的縮寫，起源於 1989 年的歐洲量子物理實驗室（CERN），主要是架構在 SGML 上的特定用途標注語言，乃是利用 SGML 定義出一套簡單易用的標籤集，使其達到容易編寫、快速傳輸的目的，我們從 Hypertext（超文件）這個字，很容易會連想到 Hyperlink（超連結），而 HTML 確實也提供了十分良好的超連結功能。HTML 自 1990 年起開始大量在 WWW（World Wide Web）上使用，由於它可以在文件中呼叫程式（如 Java Script、VB Script、CGI、ASP 等等）、可以隨著視窗大小改變文件樣式、具有良好的多媒體展示效果、便利的超連結功能，再加上它易學易懂，很快地便成爲 Internet 上的文件主流。(註 10)

HTML 也是經過 ISO 認証的國際標準，但它卻採取了比較開放的架構，任何對於 HTML 有興趣的人都可以加入這一個開放性的環境，隨著許多大型軟體公司（如 Microsoft、Netscape...）及個人的加入，HTML 自 3.0 以後的版本，遂漸出現了一些非協定內的語法，雖然都號稱與 HTML 3.0 或 4.0 相容，但卻是各有各的語法，而且與瀏覽器產生了高度的相依性，如此一來，HTML 的功能和效果增加了，但卻也使它變得愈來愈不標

準化。未來的網路環境強調跨平台、無障礙，一個愈來愈不標準的國際標準 HTML 確實另人憂心忡忡，Web 應用愈來愈廣泛，而 HTML 的瓶頸也愈來愈明顯：(註 11)

(1)HTML 著重在資料的呈現，於版面格式、體裁編排方面性能卓越，但在以結構化方式有效組織資料的能力方面，卻表現平平，使得 HTML 文件在語意（Semantic）上缺乏自我描述性。因此，HTML 文件較適合人類閱讀卻不利於機器理解。

(2)HTML 是一個固定用途的標注語言，無法根據不同領域的應用需求，由文件作者自訂標籤，此種無法擴展的特性也就侷限了它的發展。

(3)HTML 將電子文件的資料、結構和外觀三要素夾雜在一起，使其內容雜質過多，以致於欲從此種「不純」的文件中擷取有意義的資訊時困難度增加。雖然採用層級樣式表（Cascading Style Sheets）將版面編排的部份獨立於結構化資料之外，可以有效的改善此一問題，但至目前爲止似乎成果有限。

(4)內容提供者、版面編排者、資料著錄者對 HTML 文件所作的任何處理，都必須進行原始碼內容之更動，不符合資料處理原則。

在一片要求改革的呼聲中，XML 被提了出來，而且也於 1998 年正式成爲 W3C 的推薦標準，當大眾都將期望寄托於 XML 之時，我們不禁要問：XML 真的可以肩挑重擔，成爲下一代的網路文件標準嗎？

二、XML 的文件特質

XML 意爲「可擴充性標注語言」（Extensible Markup Language），是由 SGML 所精簡而來的一種通用標注語言，主要是要簡化 SGML 煩雜的結構，強化 HTML 過於簡單而不夠嚴謹的語法，希望能夠建立一個可以爲 WWW 廣泛使用、具有彈性而不失嚴謹、可以增添功能但卻不失統一標準的標注語言；綜而言之，XML 是爲了能夠同時改善 SGML 與 HTML 的缺點並擷取兩者的優點而發展，可說是 Web 文件的明日之星。HTML 是一種固定的顯示格式，內容標籤一經定義便無法隨意更改，經常會使文件製作人受限於標籤的範圍，無法進行靈活的擴充；而 XML 則允許對結構及展示部份進行雙向擴充，作者可以依個別需求而自訂標籤，而且它也解決了 HTML 過度強調文件外觀呈現而忽略文件語意結構的缺點，繼承了 SGML 文件的結構化特色，使文件內容再被處理和搜尋時更有效率，甚至可以將文件內容以資料庫的方式儲存及查詢，對於資訊檢索技術是一大突破。

XML 的發展並不是用以取代 HTML，兩者的目的不同、功能不同，可以相輔相成卻無所謂取代的問題。就目的而言，HTML 主要應用於網頁文件的設計及呈現，所有的排版效果已經預先定義，或可配合排版樣式表（如 CSS 樣式表）將資料內容依樣式表定義來呈現；XML 則可以利用自訂標籤及 DTD，將文件先予以物件化及格式化，轉換爲元件及屬性的格式，再配合排版樣本（如 CSS、XSL 等）將文件呈現出來。XML 文件經過轉換之後，也可以利用 HTML 的方式呈現出來，這個過程可以在使用 XSL（eXtensible Stylesheet Language）樣式表時由 XSLT (XSL Transformation) (註 12) 來完成。簡單地說，HTML 是 SGML 的一個應用；而 XML 卻是 SGML 的一個子集（Subset）。XML 與 SGML 同樣具備了一般性標注語言的特質，是一種可以定義其它語言的元語言（Meta Language）；而 HTML 則是一種已經被定義完成的標注語言，不具擴充與延展性，只能不斷地透過公開修訂的方式，增加一些新的功能。以 SGML 而言，自訂定完成之後便少有更動，但 HTML 卻已經修訂到第四版；XML 與 SGML 一般，具有穩定及可擴充的能力，雖然

XML 目前才剛完成 1.0 的推薦規格，但可以預見未來的應用層面將遠大於 HTML。

此外，XML 也開啓電子商務的新跑道，因爲 XML 可以自訂標籤的特性，使企業之間的來往文件都可以各依需求而定，在傳統的商業行爲中，最重要的就是訂單，而不同訂單有各種不同的面貌，而 XML 恰可以用來描述這些訂單的內容。配合使用者的需求來改變 XML 的樣貌，可以節省開發處理程式的時間及成本，因此世界知名的軟體公司，如 Microsoft、Sun、Oracle、IBM 紛紛投入 XML 相關軟體的開發，並制定許多 XML 所需的使用規範。(註 13)

參、XML 的文件展示技術

一、XML 之樣式表 Stylesheet

XML 對於文件的處理方式，也是採取內容與呈現分離的方式，一份 XML 文件可以呈現出各種不同的風貌，不同的文件內容，也可以呈現出相同的風貌，這主要是透過一些文件展示技術而來。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來呈現出不同的風貌。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。

對於網路上的標注語言（Markup Language）而言，樣式表的功能，是將文件的指定效果由瀏覽器中顯示出來，儘管作業環境不同，但由於樣式表的作用，使得文件的效果不打折扣；使用樣式表最大的好處，是可以自由地設定文件的字型、顏色、表格、段落...等，在早期的 SGML 及 HTML 中，都已使用樣式表作爲文件排版工具，XML 在發展初期，同樣認爲樣式表有其存在的必要，所以不但保留了與傳統 CSS 樣式表的相容性，更發展出 XML 專用的 XSL 樣式表，更突顯出 XML 文件的特色：

(1) XML 文件與樣式表獨立分開，因此兩者均具有獨立運作的能力，相同的 XML 文件可以使用不同的樣式表呈現，相同的樣式表，也可以顯示不同來源的 XML 文件，如此一來，文件本身更具有機動彈性，甚至可以適用到一些特殊的媒體，如 PDA（Personal Digital Assistant）等。

(2) XML 文件與樣式表獨立分開，所以 XML 文件在建立之時可以先不必考慮到以後所會呈現出的型式，因此文件的製作者可以全心投入，將注意力放在文件的結構及內容上。

(3) XML 文件與樣式表獨立分開，使電腦作業上可以直接針對結構及內容進行動作，有助於機械式的資料擷取作業及以對於資料的有效管理，網路文件管理作業，將可有效地結合資料庫，提供日後的搜尋及取得，對於網路資料的再使用，提供了龐大的效益。

前曾述及文件的三要素爲資料、結構和外觀，XML 強調的是如何以適當的結構來組織資料，對於資料外在形式的表現則必須透過其他顯示機制才能達成，這就是 XML 文件的內在、外貌分離原則。電子出版最終的讀者是「人」，因此除了注重文件的結構之外，當然不能忽略文件的外在形式。

二、層級樣式表 CSS

層級樣式表 (Cascading Style Sheets，CSS)規格分爲 Level 1 （簡稱 CSS1）及 Level 2（簡稱 CSS2）兩種，目前都已正式成爲 W3C 的建議標準。CSS1 是在 1996 年 12 月完成，並在 1999 年 1 月增修(註 14)；而 CSS2 則是在 1998 年 5 月完成的。(註 15)目前的瀏覽器都已經支援 CSS1，只是程度不同而已；至於 CSS2 則只支援其中少

許功能。

CSS 適用於 HTML 和 XML 文件，提供作者針對文件內容設定顯示時的版面格式及樣式，不但便於內容的顯示，而且可以對版面的邊界、邊框、背景及字體種類、大小、形態、字距、行距、對齊方式等進行精細的控制。CSS 分爲外部樣式，內部樣式及行內樣式，採用外部樣式更可以使 HTML 文件達到內容與樣式分離的目的。CSS2 建構在 CSS1 的基礎上，除了 CSS1 的功能外還提供了與輸出媒體相關的樣式表，不但可針對不同類型的瀏覽器、語音輸出設備、列表機、盲胞用點字設備或掌上型設備等進行特殊的設定，還提供了更精準的內容定位、字型下載、表格格式、國際化及一些與使用者介面有關的設定。

CSS 的發展在 XML 之前（CSS1 在 1996 年底便已完成，但 XML 1.0 在 1998 年 2 月份才確立），雖然在 1998 年修訂了 CSS2，但在網路標注語言的發展史上是屬於比較早期的樣式表，大致而言，XML 只是沿用 CSS 的功能而已；儘管最近有專爲 XML 所設計的 XSL 誕生(2000 年 3 月份 W3C 通過 XSL 1.0 推薦書)，但至今卻仍無法完全取代 CSS 成爲 XML 的唯一顯示標準，爲什麼一個號稱足以影響日後網路生態的 XML 語言，會選用一個早期的樣式表呢？當然 CSS 本身的優點眾多是它受到青睞的主因：(註 16)

(1) 雖然簡單卻能夠滿足顯示 XML 的基本需求。
(2) CSS 已進入到 CSS2 推薦標準，具備國際標準化的條件。
(3) 軟體的支援度高，大多數的瀏覽器都可以支援 CSS。

CSS 目前仍廣泛地使用於 HTML 及 XML 上，可以說是使用得最普遍的樣式表，HTML 呼叫 CSS 的方式是必須在<HEAD></HEAD>標籤之內宣告，而 XML 則必須在文件檔頭（Prolog）內進行宣告，若 CSS 的名稱爲 mystyle.css，則使用 HTML 及 XML 對 CSS 的呼叫方式分別如下：

在 HTML 文件中引用外部 CSS 的語法為：

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

在 XML 文件中引用外部 CSS 的語法為：

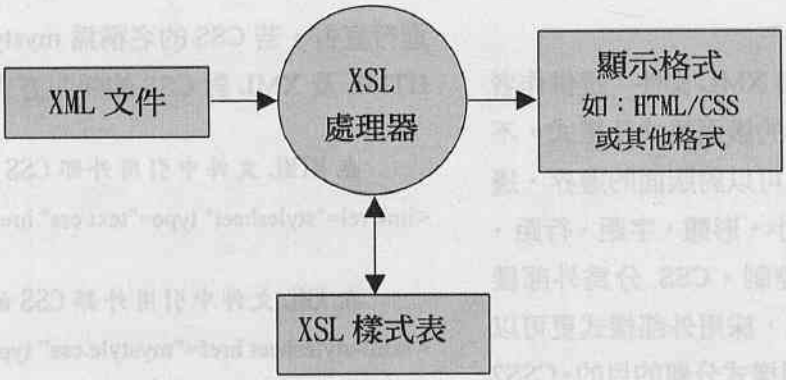
```
<?xml-stylesheet href="mystyle.css" type ="text/css"?>
```

CSS 的文件中，充滿了標籤、屬性、屬性值，在 HTML 中，屬性名稱或屬性值，在英文部份大、小寫被視爲相同，執行的結果完全一樣，但在 XML 中，英文字母的大、小寫所代表的是不同的屬性或屬性值。雖然 CSS 在網頁文件的顯示上仍居主導地位，但由於專爲 XML 設計的 XSL 也已經於 2000 年 3 月份完成了 1.0 版的推薦規格，在未來對於 XML 的需求更爲殷切，功能及效能的要求日益提高的情況下，相信 XSL 也將逐漸展露頭角，扮演舉足輕重的地位。

三、可擴展樣式語言 XSL

XSL（eXtensible Stylesheet Language，可擴展樣式語言）是一個最新研議的排版樣式，W3C 於 1998 年 12 月提出了 1.0 版的工作草案，2000 年 1 月公告修訂版，該工作草案的最終版本（Last Call）則已於 2000 年 3 月 27 日正式公佈(註 17)，相信不久後正式的建議標準（Recommendation）便會出爐，爲網路文件展示方式，掀起另一陣旋風。

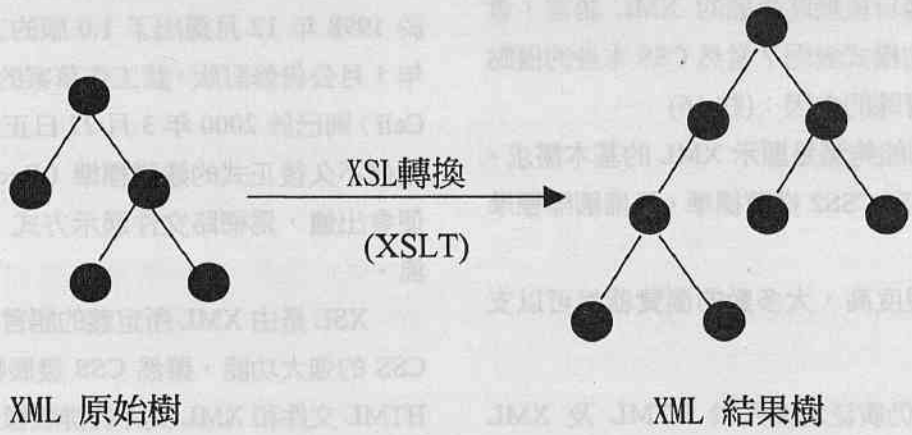
XSL 是由 XML 所定義的語言，它提供遠超過 CSS 的強大功能，雖然 CSS 發展較早，並可作爲 HTML 文件和 XML 文件的排版樣式，但卻無法對原始文件進行內容過濾或結構重組，因此，在處理高度結構化的 XML 文件以及複雜多樣的 Web 應用時，必須仰賴 XSL 所提供的功能才能達成。



圖二：以 XSL 顯示 XML 文件(註 18)

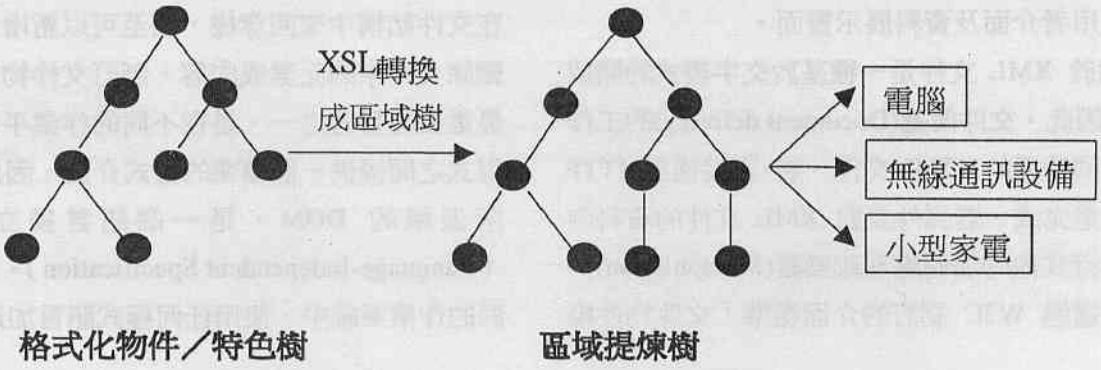
XSL 處理 XML 文件的過程如圖二所示。其中，XSL 樣式表以「樣版」(Template) 和「樣式」(Pattern)來描述 XML 文件的轉換規則；而 XSL 處理器則負責先將 XML 文件以樹狀結構建立「原始樹」(Source Tree)，接著將樣式與原始樹中的元素相匹配，再根據樣版建立「結果樹」(Result Tree)；結果樹便構成了顯示的文件結構。由於結果樹和原始樹是分離的，原始樹可以被過濾、重組或增刪而產生結果樹，因此，目的文件可以擁有和原始文件完全不同的結構。是以使用 XSL 不僅可將 XML 文件轉換爲 HTML 文件或其他排版格式，如 PDF、RTF、TeX ... 等，也可以將文件結構轉換至另一個結構。

XSL 是一種樣式表示語言，任意給予一個具有結構的文件或檔案，設計者便可以使用 XSL 呈現出文件的原始內容、外貌，並且可以將結果輸出到任意的媒體之中，因此文件的「轉換」是一個重要的關鍵，至於如何轉換則依賴另一個相關的標準，稱爲 XSLT (XSL Transformation)，藉由 XSLT，一份原始的 XML 文件可以被轉換成任意的輸出模式，整個轉換的過程由 XSLT 執行，如圖三所示。(註 19)



圖三：XML 文件的轉換（一）

XML 文件經過剖析之後會先被「物件化」，產生類似於物件節點的結構，再經過 XSLT 的轉換過程之後，產生一個以元件（Element）及屬性（Attribute）爲主的結果樹，而這個結果樹又稱爲「元件／屬性樹」（Element and Attribute Tree）；但轉換的工作並未結束，XSLT 會再針對「元件／屬性樹」再進行一次「格式化」的工作，將臨時性質的結果樹再轉換爲「區域樹」（Area Tree），如圖四所示。



圖四：XML 文件的轉換（二）

在 XML 文件中引用 XSL 的語法為：

<?xml-stylesheet type="text/xsl" href="mystyle.xsl"?>

若將 CSS 與 XSL 兩相比較，則 CSS 採用簡單的比對方式，直接將作者所設定的樣式套用在文件上，完全沒有改變原始文件的結構及內容順序；而 XSL 則提供了較複雜的機制，可以對原始文件進行內容過濾和結構重組，產生全然不同的文件。下表基於幾項要點，列出兩者的差異：(註 20)

表一：CSS 與 XSL 比較表

	CSS	XSL
是否適用於 HTML？	是	否
是否適用於 XML？	是	是
是否是一種轉換語言？	否	是
使用的語法	CSS	XML

四、文件物件模型 DOM

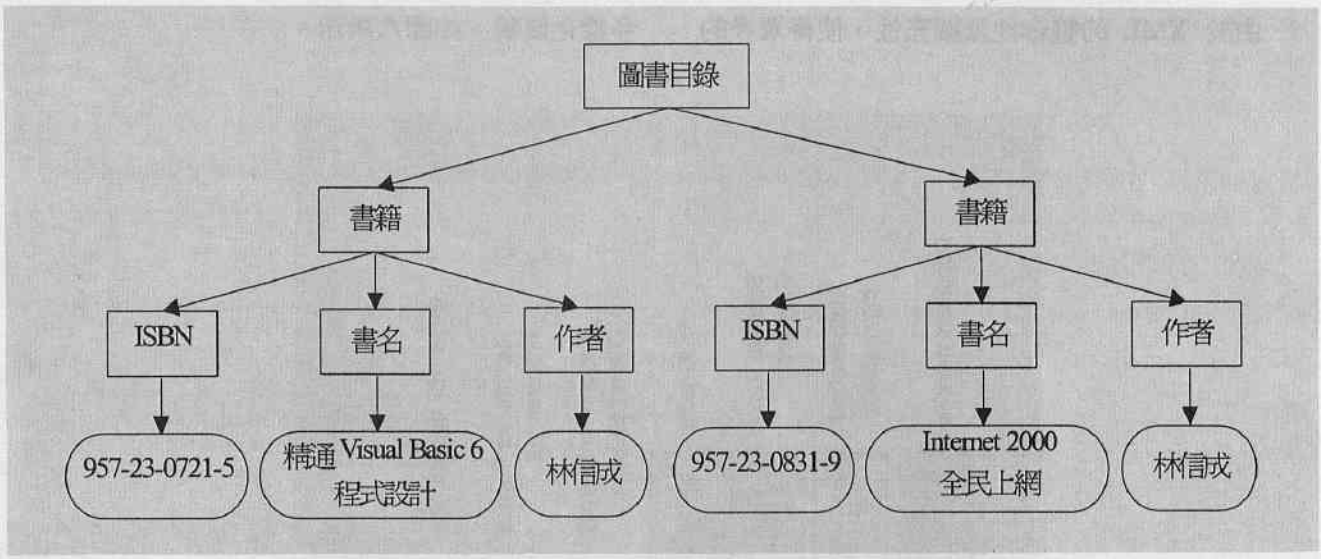
對於無法使用 CSS 或 XSL 完成的工作，文件出版者可以透過分解 XML 結構樹的方式，將 XML 文件呈現在讀者眼前。此種方式是直接以程式剖析 XML 文件：首先藉由文件物件模型 DOM，建立資料分析、擷取的管道，再利用程式語言的處理能力對資料進行各種加值處理，最後建構使用者介面及資料展示畫面。

由於 XML 文件是一種基於文字模式的開放規格，因此，文件傳遞(Document delivery)的工作可以如同普通的HTML文件一般，直接透過HTTP通訊協定完成。若另外要對 XML 文件的資料內容，進行其他的加值處理或操縱(Manipulation)，則可以透過 W3C 制訂的介面標準「文件物件模型」(Document Object Model，簡稱 DOM)爲之。(註 21)其實 DOM 並非是針對 XML 量身定做的，而是一套普遍適用於類似 HTML、XML 等文件的應用程式介面（Application Programming Interface，API)。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，電腦程式可以輕易的建立文件，可以在文件結構中來回穿梭，甚至可以新增、修改或刪除文件中的元素或內容。制訂文件物件模型的最重要的目的之一，是在不同的作業平台和應用程式之間提供一個標準的程式介面。因此，W3C 所倡議的 DOM，是一個語言獨立的規格（Language-Independent Specification)，可以在不同的作業系統中，使用任何程式語言加以實現。

```
<?xml version="1.0" encoding="BIG5" standalone="yes"?>
<圖書目錄>
    <書籍>
        <ISBN>957-23-0721-5</ISBN>
        <書名>精通 Visual Basic 6 程式設計</書名>
        <作者>林信成</作者>
    </書籍>
    <書籍>
        <ISBN>957-23-0831-9</ISBN>
        <書名>Internet2000 全民上網</書名>
        <作者>林信成</作者>
    </書籍>
</圖書目錄>
```

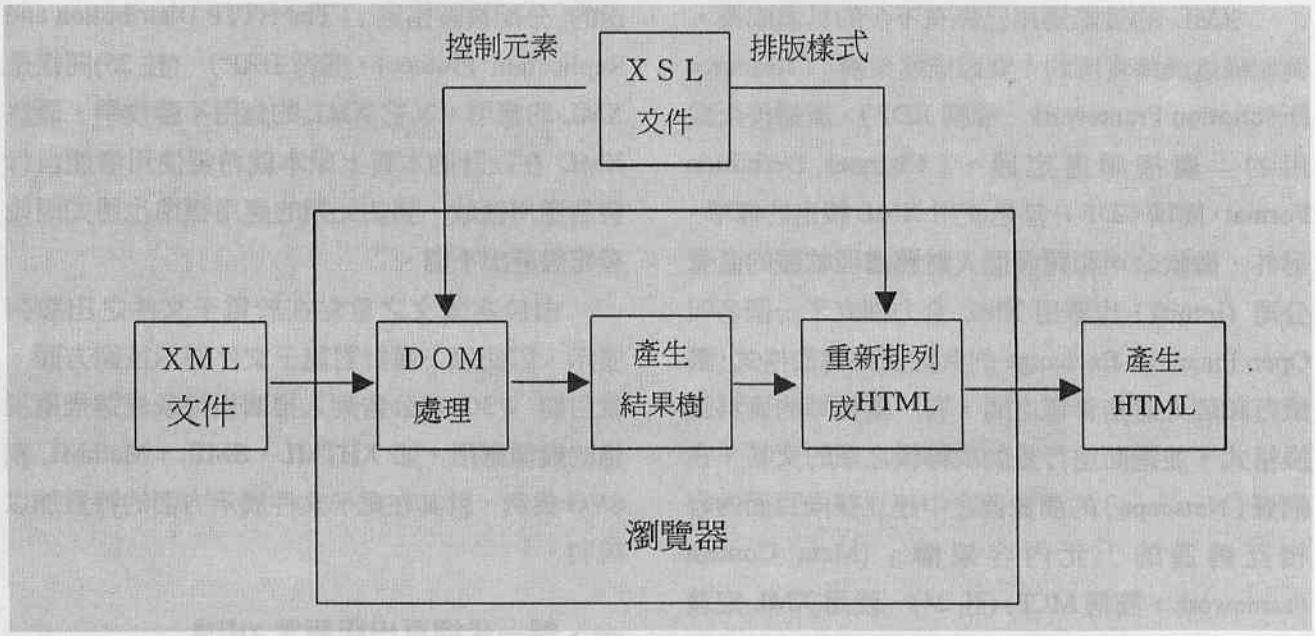
圖五：以 XML 撰寫之圖書目錄

由於網路上的資料分別存放在各種異質系統中，而 XML 以文件方式來描述各種資料的作法，使得 DOM 成了管理資料的有效途徑。在 DOM 模型中，文件中的各個元素（Element）都被視爲一個一個的節點（Node)，而整份文件則被描述成樹狀結構（Tree Structure)。例如，圖五是一個以 XML 撰寫之簡易圖書目錄，若將此 XML 文件以 DOM 展開，則形成如圖六所示的樹狀結構。圖中的矩形方塊表示文件中的節點，而橢圓形則爲該節點的內容。這樣的資料結構可以很容易的以程式加以處理，如新增、修改、刪除節點等，而這些動作正好就是文件處理所需的各項功能。



圖六：以 DOM 展開之樹狀結構

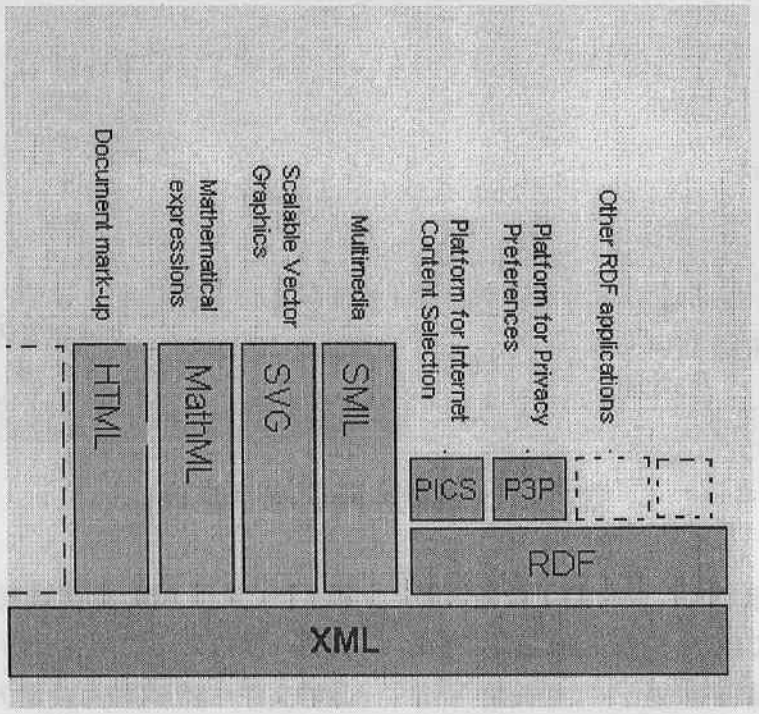
一般瀏覽器在展示 XML 文件時，會經過兩個主要的流程：(1) 以 DOM 分析 XML 文件的結構，以便將文件中的資料擷取出來，並使用 XSL 中的控制元素加以重新排列，產生一份暫存資料檔；(2) 使用 XSL 的樣式表，將暫存資料檔的資料再一次排列，並產生一份可以在瀏覽器中被顯示的文件，這份文件可能是 HTML 或其它型式(註 22)，如圖七。



圖七：XML 文件的處理流程

肆、XML 文件展示專用語言

由於 XML 的整合性及擴充性，使得眾多的 XML 應用語言可以互相結合，展現出電子文件的多樣化風貌，如圖八所示。



圖八：XML 文件展示專用語言之整合(註 23)

XML 的實際應用已經有不少的具體成果，例如描述網路資源的「資源描述架構」(Resource Description Framework，簡稱 RDF)、廣播技術採用的「廣播頻道定義」(Channel Definition Format，簡稱 CDF)，都是使用 XML 製定的標準。另外，微軟公司和開發個人財務處理軟體的直覺公司 (Intuit)，也應用 XML 合力創立了一個名叫 Open Financial Exchange 的共同資料傳輸格式，讓銀行和個人財務管理之間，有一個標準的資料互換格式，並進而進行查詢或轉帳之類的交易。在網景(Netscape)的瀏覽器之中建立雙向頁面內容相互溝通的「元內容架構」(Meta Content Framework，簡稱 MCF) (註 24)，使用 XML 定義並已在 W3C 研議之中，將來或許有機會成爲共通標準。另一種可以減少相同檔案在網路上重覆傳送的「分配複製協定」(The HTTP Distribution and Replication Protocol，簡稱 DRP) (註 25)同樣是 XML 的應用。其它 XML 的應用不勝枚舉，既然 XML 在設計的本質上原本就希望使用者能自行發展應用領域，諸如此類的應用標準也將如雨後春筍般層出不窮。

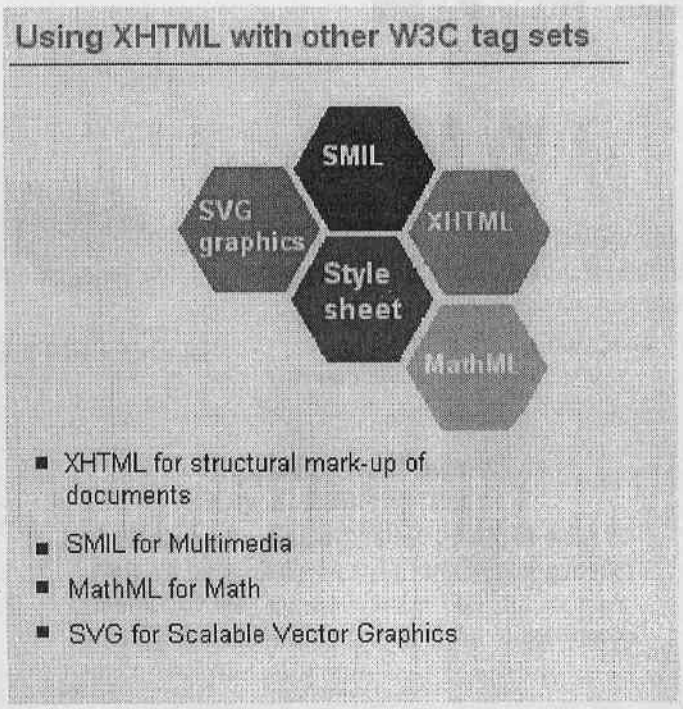
由於本論文之重點在於電子文件之出版與展示，因此以下僅針對電子文件展示技術方面，就目前 W3C 已公告列入推薦規格或候選推薦規格的幾種應用，如 XHTML、SMIL、MathML 和 SVG 爲例，就其在電子文件展示方面的特質加以說明。

一、新一代網頁出版語言 XHTML

無論如何，HTML 歷經多次的演進，已經擁

有支援廣泛 Web 應用的能力，因此目前仍是網路上最通用的 Web 出版語言。HTML 是 SGML 的一個應用，其最新規範是 W3C 於 1999 年 12 月所發佈的 HTML 4.01 建議標準 (註 26)，該標準修正了 1998 年 4 月發佈的 HTML 4.0 規範中的某些缺陷（Bug）。(註 27)

在 XML 規範正式成爲 W3C 建議標準之後，HTML 工作小組便擬定將 HTML 從原有的 SGML 框架中移植至 XML 框架上的計畫，開始著手將 HTML 4 以 XML 語法重新定義，並於 2000 年 1 月頒佈了基於 XML 之新一代網頁出版語言 XHTML 1.0（Extensible HyperText Markup Language，可擴展的超文件標注語言）。(註 28)XHTML 不但具備了 HTML 強大的超媒體網頁表現能力，最重要的是它擁有 XML 的整合性及擴充性。XHTML 支援 XML 的 Namespace，使得 XHTML 可結合眾多 XML 應用語言，例如：SMIL、SVG、MathML ... 等，展出電子文件的多樣化風貌，如圖九所示。

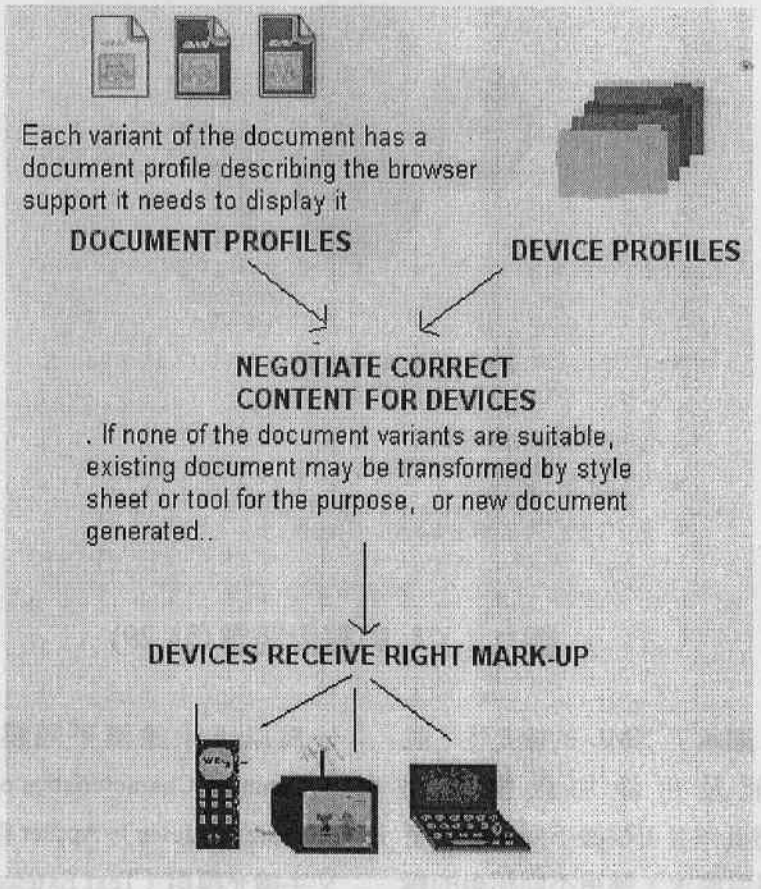


圖九：XML 與應用語言 (註 29)

此外，XHTML 繼承了 XML 的擴充性，並將標籤集依功能及用途加以模組化（Modularization）。網頁作者可根據不同的使用者終端設備或不同的應用需求，選用不同模組的標籤子集；若是現有的標籤集不敷使用，模組化也提供了一個擴充標籤外集的正規途徑。依據 W3C 目前的工作草案，XHTML 標籤集依其用途大致分爲以下十餘個不同模組：共同字元模組（Common Characteristics of Modules）、基本模組（Basic Modules）、Applet 模組（Applet Module）、文字擴充模組（Text Extension Modules）、表單模組（Forms Modules）、表格模組（Table Modules）、影像模組（Image Module）、客戶端影像映射模組（Client-side Image Map Module）、伺服端影像映

射模組（Server-side Image Map Module）、物件模組（Object Module）、框架（Frames Module）、Iframe模組、事件模組（Intrinsic Events）、詮釋資訊模組（Metainformation Module）、描述語言模組（Scripting Module）、樣式表模組（Stylesheet Module）、連結模組（Link Module）、基底模組（Base Module）和 Legacy 模組（Legacy Module）等。(註 30)

基於未來性的考量，人們除了桌上型電腦（Desk-Top Computers）之外，將藉由更易於攜帶的行動設備（Mobile Devices），如個人數位助理器（Personal Digital Assistants，PDA）、掌上型電腦（Palm-Top Computers）、車用電腦（In-Car Computers）、行動電話（Cellular Phones）、呼叫器（Pagers）... 等，來存取網路資訊、瀏覽網頁內容。此一目標必須藉助設備描述檔（Device Profile）和文件描述檔（Document Profile）來達成。設備描述檔紀錄著某一已知設備究竟支援了哪些 XHTML 模組；而文件描述檔則記錄著某一份特定文件要能正常呈現在某個已知設備時，所需要的 XHTML 文件和樣式表爲何，當然，這其中仍須某些轉換程式的輔助，其運作過程如圖十所示。



圖十：文件在不同的設備上呈現出不同的版面(註 31)

二、同步多媒體整合語言 SMIL

雖然 Web 已經成爲多媒體文件的重要發行管道之一，但由於先天的限制，HTML 在同步性、互動性和即時性方面一直存在著若干限制。爲了提升 Web 處理多媒體文件的能力，使獨立的多媒體文件能被整合成爲媲美電視的同步多媒體節目，W3C 遂於 1997 年 3 月成立了同步多媒體工作小組，基於 XML 規範制訂了適合 Web 展示的同步多媒體整合語言 SMIL（Synchronized Multimedia Integration Language），並於 1998 年 6 月正式成爲 W3C 的建議標準。(註 32)

SMIL（唸作 Smile）是一個非常容易學習的語言，它也是 XML 的一個應用，如同 HTML 文件一樣，SMIL 文件同樣可以利用一般簡易文書編輯器來製作。SMIL 文件可內含串流語音（Streaming audio）、串流視訊（Streaming video）影像（Images）、文字（Text）或任何其他的媒體類型。SMIL 最重要的特性在於作者不但可以描述多媒體文件的時序行爲，精確規劃螢幕上的佈局，將多媒體物件與超連結相關連，更能設定各種多媒體互動情境，適合在網路上進行即時互動的出版、廣播、教學、娛樂等。

SMIL 的主要特點爲：

1.同步多媒體播放

完美的多媒體環境中，圖、文、影、音等多媒體元素必須能夠依時序同步播放，例如：「播送視頻文件 A 的同時一起播放旁白文件 B」，或「在旁白文件 B 播放完畢 1 秒後接著播放動畫文件 C」等。Web 雖然被標榜爲一個多媒體環境，但 HTML 卻缺乏時序上的同步控制機制。SMIL 可以表述這類訊息，因而可以在 Web 上建立基於時序的同步多媒體播送系統。

2. 簡化創作工具

目前，在 Web 上發行同步多媒體文件，都需要專用的創作工具（Authoring tools）或者要進行複雜的程式設計。而撰寫 SMIL 文件則和 HTML 文件類似，只需一個簡單的文書編輯器即可，並使用一些簡易的 XML 元素，不用學習複雜的腳本語言。

3. 與 Web 技術緊密結合

SMIL 中包括現階段及未來的 Web 技術，如 CSS 樣式表、HTML 超連結以及基於 XML 的語法。因此，SMIL 可藉由 XML Namespace 整合到其他需要同步多媒體功能的 XML 應用環境中。

4. 提高頻寬利用率

一般而言，視頻訊號是多媒體元素中頻寬需求量最高的。而 SMIL 的特色是在顯示視頻內容時，可以盡量避免將低頻寬需求的文字、圖片轉換成高頻寬需求的視頻訊號，以提高頻寬的利用率。

5. 促進訊息國際化

SMIL 可以滿足多國語言的需求，例如：在同一網頁中包括中、日、英等多國語言的語音文件，然後根據使用者的參數設定自動選擇下載中文、日文或英文版本。

三、數學標注語言 MathML

在科技文獻中，數學公式是非常重要的表現元素，非常可惜的是 HTML 雖然擁有無數好用的標籤群，然而對於數學公式的支援卻明顯不足，以致於科技文獻的作者只好另謀他法來呈現數學公式，例如利用圖形方式來展現。這造成了三個主要問題：其一，圖形資料量龐大，佔用網路頻寬，影響傳輸效率；其二，圖形化的公式無法重複利用，例如無法編輯或修改公式內容；其三，

圖形化公式只適合人類閱讀，機器無法理解公式內涵。這些都使得目前要在 Web 上傳遞公式內涵或顯示公式畫面都極其麻煩，以致於在 Web 上進行科技文獻的線上出版困難度增加。

有鑑於此，W3C 遂於 1998 年 4 月提出專爲出版數學公式的標注語言 MathML（Mathematical Markup Language），並於 1999 年 7 月修訂。(註 33)MathML 的標籤群可概分爲展示標籤（Presentation tags）和內容標籤（Content tags）兩大類：展示標籤主要用途在將數學公式外觀以高解析度的方式呈現出來供人類閱讀；而內容標籤則用來將公式內涵以語意化的方式標注，作爲不同應用程式之間的傳輸介面，進行分散式科學運算及處理。

四、向量式圖形標注語言 SVG

向量化（Vector）圖形和點陣式（Bitmap）圖形是電腦圖形的兩大類型，目前 Web 在處理圖形上所使用的標準格式，如 JPEG 和 GIF，基本上都屬於點陣式圖形。爲了能增強 Web 處理向量圖的能力，W3C 遂組成工作小組進行向量圖形語言之制訂，並於 1999 年 2 月頒佈第一份 SVG（Scalable Vector Graphics）工作草案，而最新的草案則於 2000 年 3 月公布。(註 34)SVG 也是 XML 的一個應用，主要是作爲描述二維圖形的語言，它支援向量圖形（Vector graphic shapes）、影像（Image）和文字（Text）等物件，並允許作者對物件進行群組化、樣式化、轉換和合成。SVG 物件可以直接以腳本程式透過 DOM 來呈現動態及互動式功能，並且由於 SVG 物件支援豐富的事件控制器(Event handler)，如 Onclick、Onmouseover ... 等，因此可與現有的 Web 標準相容。再者，藉由 XML Namespace，SVG 元素與全球其他 XML 元素可同時整合在同一個網頁中。

伍、結論

網路化服務的時代已經來臨，現今在網際網路上的文件格式大多是 HTML 文件，儘管有許多程式語言如 CGI、Script、ASP 等已經可以提供相當程度的互動效果，但是當企業希望透過網路來做更多的事，或是希望透過即時的服務來收取費用時，簡單的 HTML 功能就顯得捉襟見肘，例如：利用網路全文檢索引擎提供快速資訊檢索、提供聲音/影像的多樣性功能、管制每個用戶的存取權限及消費記錄、提供即時的付費系統等；這些新增的服務都必須要有相對的系統功能來支援，而且必須將所有的系統整合在一起，當文件的功能越來越複雜，維護的人力也就越來越多，當然，成本也就相對的提高許多。

網路電子文件的發行需要一套完整的出版架構，這個架構必須是可以用來簡化資料處理、進行資料存取控制、提供檢索服務等工作，這些工作在個人電腦環境中，可能只需要一套資料庫管理系統（Database Management System, DBMS）或資訊檢索系統就可以解決了，但是個人電腦的資料數量十分有限，而且具有封閉性，管理上的問題不多，一旦進入到 Internet 上，一則資料的數量無法估計，再則 Web 文件通常缺乏結構，很難加以分門別類，三則必須考量到各種不同作業平台之間的文件轉換，都使得網路文件管理的工作浩大而困難。

在過去，電子文件的出版，大多是以網頁的方式呈現，但是 HTML 的文件缺乏互動性及延展性，文件的內容大多以瀏覽爲導向，即使加上了一些互動性的元件，也大多只限於程式所能夠控制的部份，即使能夠進行資料庫處理，也必須預先建好欄位資料，以供人存取；更進一步，如果想要提供即時電子出版的資訊，大多必須透過一些線上服務公司，如 Amercian Online、CompuServe、

Prodigy 等，來提供資料傳送、計費、付款、客戶服務等業務。

平面紙張、光碟與網路是出版的三種方式，這三種方式是相輔相成的，與電子出版相較之下，網路具有四項優勢： (1) 網路的受眾多；(2) 參與網路的產業廣泛；(3) 網路的互動性強；(4) 網路的資訊更新速度快。電子出版與網路結合已是無法避免的趨勢。(註 35)

XML 是一種良好的文作展示技術，而且愈來愈多的人相信 XML 可以爲目前混亂的網路世界帶來新的希望，但是一份 XML 文件要能夠順利地產生展示結果，實際上需要經過許多程序，包括將 XML 文件依 DTD 確認是否是爲 "Validating" 或 "Well-formed"，依據 DOM 透過 XML 的剖析器分析出文件的結構，完成了文件的轉換工作之後，再使用樣式表完成輸出工作，這一連串的動作看似繁雜，但隨著電腦硬體能力的快速提昇，在處理上不成問題，倒是 HTML 行之有年，要改變使用者的習慣可能需要一段時間。

至於誰該使用 XML 呢？我們已經知道 XML 不是爲了要取代或消滅 HTML 而設計的，我們也知道 XML 文件可以用 HTML 作爲輸出的結果，那麼我們爲什麼要選擇 XML 呢？基本上，如果文件的目的只是要讓讀者閱讀資料，使用 HTML 已經綽綽有餘，但如果 Web 文件需要進行一些內容處理，如資料庫查詢、電子商務、網路出版、無線通訊、小型通訊家電等，XML 所展現出的能力就非 HTML 所能比擬；在一個以「內容提供」（Content provide）及資料管理爲主的應用領域上，XML 是最佳的選擇。

（收稿日期：2000 年 12 月 28 日）

註　釋：

註1：「XML 工作小組」最初稱爲「SGML 編審委員會」(SGML Editorial Review Board)。

註2："Extensible Markup Language (XML) 1.0," W3C Recommendation 10 Feb. 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

"Extensible Markup Language (XML) 1.0 (Second Edition)," W3C Recommendation, 6 Oct. 2000, <http://www.w3.org/TR/2000/REC-xml-20001006>.

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

註4：林信成，「XML 相關技術與下一代 Web 出版趨勢之研究」，教育資料與圖書館學，37:2 (民國 88 年 12 月)，頁 184-210。

註5：標注語言（Markup Language）一般可區分爲兩大類：(1).一般用途標注語言；(2).特殊用途標注語言。

註6：Rich Text Format Specification (RTF) 是 Microsoft 公司於 1994 年所制定的一種文件格式，與 Adobe 公司的 PDF 格式相似，是一種具可攜性的文件格式（Portable Document Formats）。<http://www.a2e.de/oas/dokporten.html> (13 Apr. 2001).

註7：Postscript 可以應用在印表機的列印字型上，<http://bbs.ece.fcu.edu.tw/~hp/linux/9/7/1.html>.

註8：HTML 檔案可以轉換爲一般 ASCII 檔、Postscript 以及其它可以列印的格式，<http://www.tp.edu.tw/wwwfaq/document/cwwwfaq/printing.htm>.

註9：陳錦輝、江鈞，從 HTML 到 XML，(台北市：文魁資訊，民國 89 年)，頁 1-12。

註10：林信成，WWW 超媒體網頁 DIY，(台北市：第三波，民國 89 年)。

註11：同註 4。

註12： XSLT(XSL Transformation)是 XSL 樣式表的轉換元件，XSLT1.0 於 1999 年 11 月 16 日成爲 W3C 的推薦規格，完成的時間稍早於 XSL 1.0，20000 年 3 月 27 日，XSL 1.0 通過完成。

註13： Malcolm Maclachlan, "New Products Help XML Gather Momentum," TechWeb News, Feb 1998, <http://www.techweb.com/wire/story/TWB19980202S0007>.

註14： Håkon W. Lie and Bert Bos, "Cascading Style Sheets (CSS1) Level 1 Specification," W3C Recommendation, 11 Jan. 1999, < http://www.w3.org/TR/REC-CSS1>.

註15： Bert Bos, Håkon Wium Lie, Chris Lilley and Ian Jacobs, "Cascading Style Sheets, level 2 (CSS2) Specification," 12 May 1998, <http://www.w3.org/TR/REC-CSS2>.

註16： 同註 9，頁 7-5。

註17： Sharon Adler, Anders Berglund, Jeff Caruso, Stephen Deach, Alex Milowski, Scott Parnell, Jeremy Richman and Steve Zilles, "Extensible Stylesheet Language (XSL) Version 1.0," 12 Jan. 2000, <http://www.w3.org/TR/xsl>.

註18： Microsoft Web Workshop, "XML Architecture," MSDN Online, <http://msdn.microsoft.com/xml/general/architecture>.

註19： "Extensible Stylesheet Language (XSL)Version 1.0," W3C Working Draft, 27 March 2000

註20： Jakob Nielsen, "Web Style Sheets," <http://www.w3.org/Style/>.

註21： Vidur Apparao, Steve Byrne, Mike Champion, Scott Isaacs, Ian Jacobs, Arnaud Le Hors, Gavin Nicol, Jonathan Robie, Robert Sutor, Chris Wilson and Lauren Wood, "Document Object Model (DOM) Level 1 Specification Version 1.0, " W3C Recommendation, 1 Oct. 1998, <http://www.w3.org/TR/REC-DOM-Level-1>.

註22： 同註 7, 頁 8-4。

註23： Jon Bosak and Tim Bray, "XML and the Second-Generation Web," Scientific American, May 1999, <http://www.sciam.com/1999/0599issue/0599bosak.html>.

註24： "Meta Content Framework Using XML," Submitted to W3C, 6 June 1997, <http://www.w3.org/TR/NOTE-MCF-XML/ >.

註25： "The HTTP Distribution and Replication Protocol," Submitted to W3C, 25 Aug. 1997, <http://www.w3.org/TR/NOTE-drp-19970825.html >.

註26： "HTML 4.01 Specification," W3C Recommendation, 24 Dec. 1999, <http:// www.w3.org/TR/html4/>.

註27： HTML 4.0 建議標準最初於 1997 年 12 月發佈，並於 1998 年 4 月修訂，規範內容請參見 "HTML 4.0 Specification," W3C Recommendation, 24 April 1998, <http://www.w3.org/TR/1998/REC-html40-19980424/>.

註28： "XHTML™ 1.0: The Extensible HyperText Markup Language, A Reformulation of HTML 4 in XML 1.0," W3C Recommendation, 26 Jan. 2000, <http://www.w3.org/TR/xhtml1/>

註29： "HyperText Markup Language -- Activeity Statement," <http://www.w3.org/ Markup/Activity>.

註30："Modularization of XHTML," W3C Working Draft, 5 Jan. 2000, <http://www.w3.org/TR/xhtml-modularization/>.

註31： 同註 23。

註32： "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," W3C Recommendation, 15 June 1998, <http://www.w3.org/TR/REC-smil/>.

註33： "Mathematical Markup Language (MathML™) 1.01 Specification," W3C Recommendation, 7 July 1999, <http://www.w3.org/TR/REC-MathML/>.

註34： "Scalable Vector Graphics (SVG) 1.0 Specification," W3C Working Draft, 3 March 2000, <http://www.w3.org/TR/SVG/>.

註35： 「開啓兩岸電子出版業的合作契機」，兩岸電子出版研討會，<http://web3.ttimes.com.tw/2000/06/28/1/book/200006140396.html>.