大學圖書館
3卷1期(民國88年1月) 頁89-103

SGML、HTML與XML之比較
Comparison of SGML, HTML and XML

陳嵩榮 Sung-Jung Chen
輔仁大學圖書資訊學系碩士班學生
Graduate student, Dept. of Library & Information Science,
Fu-Jen Catholic University
E-mail: ms486131@green.lins.fju.edu.tw

【摘 要】

本文將依發展的時間順序介紹三種值得電子圖書館領域注意的電子文件格式：標準通用標示語言（Standard Generalized Markup Language；簡稱SGML）、超文件標示語言（Hypertext Markup Language；簡稱HTML）與最新崛起的可擴展標示語言（Extensible Markup Language；簡稱XML），並作簡單的比較。

【Abstract】

This article is to introduce three formats of electronic document in order of development which deserve close attention by digital library community. They are SGML（Standard Generalized Markup Language），HTML（Hypertext Markup Language）and the new XML（Extensible Markup Language）. A Simple comparison among the three is made to show their differences.

關鍵詞：標準通用標示語言；超文件標示語言；可擴展標示語言；標示
Keywords：SGML (Standard Generalized Markup Language); HTML(Hypertext Markup Language); XML (Extensible Markup Language); Markup

壹、前言

電腦是處理資訊的工具，硬體是處理資訊的實體資源，軟體則是爲了處理特定資訊設計出來的流程。電腦之所以能對我們的生活帶來很大的效益，是因爲電腦具有處理資訊的強大能力，硬體與軟體的不斷進步代表著電腦處理資訊的能力愈來愈強，但電腦要能夠發揮效益，除了要有硬體和軟體之外，還要輸入資料，資料經過電腦的處理，輸出有價值的資訊，這才是我們使用電腦的主要目的。平常電腦硬體與軟體的進步總會引起較大的注意，例如：Intel發表新一代的微處理器、Microsoft發表新的作業系統或應用軟體總是資訊界的盛事，但一般人較少注意到資料格式（data format）的發展事實上對資訊的處理、管理與利用所帶來的影響並不遜於軟硬體的革新，尤其對電子圖書館等保存大量公共資訊的組織而言，所採用的資料格式必須確保所有的文件資訊能妥善地長期保存，並能以最有效率的方式被存取與傳佈，因爲這些資訊是社會重要的資產，而且通常是對社會較有價值的資訊。（註1）

本文將依發展的時間順序介紹三種值得電子圖書館領域注意的電子文件格式：標準通用標示語言（Standard Generalized Markup Language；簡稱SGML）、超文件標示語言（Hypertext Markup Language；簡稱HTML）與最新崛起的可擴展標示語言（Extensible Markup Language；簡稱XML），並作簡單的比較。

SGML是ISO在1986年所頒佈的國際標準（ISO 8879），在電子圖書館相關計劃中，目前有TEI（Text Encoding Initiative）、EAD（Encoded Archival Description）、CIMI（Consortium for the Interchange of Museum Information）、DIAP（Digital Image Access Project）與美國國會圖書館的 American Memory Project 等採用SGML作爲文件格式（註2, 3, 4）；HTML是SGML的一個應用，是一種用以創造超文件（hypertext）的簡易資料格式，目前在全球資訊網（World Wide Web；簡稱WWW或Web）獲得普遍的採用，是寫作網頁（webpages）的標準語言； XML是全球資訊網聯盟（World Wide Web Consortium；簡稱W3C）在1996年底所提出的標準，1998年2月公佈XML 1.0 Recommendation，相關標準目前仍在發展之中。這個新一代的標示語言被期許能具有SGML的彈性，但又不像SGML般複雜，並能如HTML般能在Web上傳送。

貳、何謂標示（markup）？

SGML、HTML、XML都是標示語言（Markup Language）。最早，標示是文件在排版時，用來指示文字如何編排的指令，包括控制字體的大小、字型的選擇（如楷體、細明體等）、字形的處理（如粗體、斜體、加底線等）、頁面的大小（如A4、B5等）、

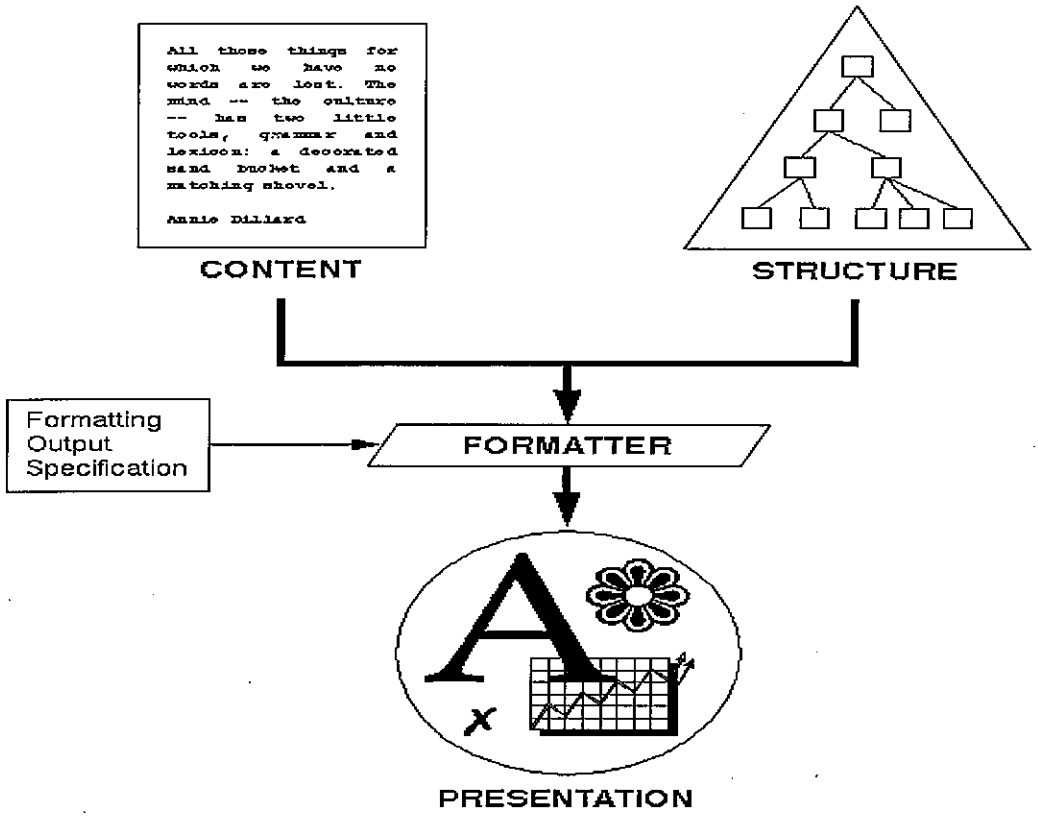
天地左右的留白寬度、標題、段落、註腳、表格……等，這些標示並不處理文件的內容，主要用來處理文件實際的呈現外觀，這類的標示稱爲程序性標示（Procedural Markup）。（註5）大部分的電子出版或文書處理軟體都使用專屬性的程序性標示，也就是說這些系統都使用專屬的控制碼來執行文件的處理，如字體的加粗、放大…等，這些專屬的控制碼大都只能在特定平台的特定的系統或相關軟體中執行，如果所使用的硬體或系統軟體換了，這些標示過的文件往往必須進行重新標示的工作，這種資料轉換所花費的代價通常相當的大；同樣的，使用程序性標示的文件在交換時，文件交換的雙方通常必須使用相同的系統。一般而言，程序性標示的作用都只針對單一文件，例如控制某一份文件以特定的格式輸出，如果同一份文件內容希望以不同的呈現外觀再利用，必須移除先前的標示，加入符合新的呈現外觀的標示；或者將同樣的文件內容複製一份，爲新的呈現外觀進行標示。

另一類的標示稱爲描述性標示（Descriptive Markup），也稱爲通用標示（Generic Markup或Generalized Markup），所描述的是文件的內容（content）或結構（structure），而不是文件的呈現外觀。描述文件結構的方式是以標示指定結構中的元素（elements），以文章爲例，其結構可能有標題、章、節、參考書目……等；以書信爲例，結構可能有寄件人、收件人、書信內容……等。通用標示的基本概念是文件的內容必須和文件的呈現資訊（即所謂的樣式）分開（註6），由於內容、結構與呈現資訊分開，所以同一份文件內容可以有多種呈現方式，例如同一個檔案可能以紙本、線上、CD-ROM與Web版本發行。

以圖一（註7）爲例，Content 是文件的本文資訊，Structure 是標示的結構，這兩部份是通用標示所要描述的，Presentation是文件的呈現外觀，在Content、Structure與Presentation之間的Formatter主要是用來設定文件的輸出格式，樣式表（Style Sheet）的角色就是屬於這個部分。由於內容、結構與呈現資訊分開，如果同一份文件內容希望以不同的呈現外觀再利用，只須修改樣式表或產生一份新的樣式表即可，文件內容與結構的標示完全不須更動，因此通用標示在文件的再利用方面要比程序性標示有彈性的多。

XML、SGML都是典型的通用標示語言，HTML就比較特殊了，HTML的標籤集（tag set）中大部分是用來設定文件在Web上的呈現外觀，少部分描述文件的結構（如：<head>, <title>, <body>）。由於HTML是SGML的一種應用，所以HTML並非程序性標示語言，而是以描述性標示的方式來控制文件的呈現外觀；HTML的樣式是內建的，並不像XML、SGML有分離的樣式，所以HTML標示的作用與程序性標示類似，是針對單一文件，並不像XML、SGML有較高的再利用彈性。

圖一



參、SGML

SGML是ISO在1986年所訂定的標準，編號是ISO 8879。SGML 是一種元語言（meta-language），元語言是一套可以用來定義其他更專門性的標示語言的通用規則，HTML就是由SGML所定義出來，專門使用在WWW上的標示語言。（註8）SGML主要應用在文字資料的交換，但也適用於其他的資料型態，SGML是很好的資料儲存格式，適用於任何複雜的文件結構，但缺點是在網路傳輸（network delivery）方面較爲不便。

由於介紹SGML的中文文獻已有一些（書目請參考註9, 10, 11, 12, 13），本文不再對SGML作全面性的介紹，只介紹SGML的優點與限制。

一、SGML的優點：（註14）

1.有彈性（flexibility）：SGML 能描述任何的資訊結構與任何複雜的文件，其應用可以簡單如 HTML，也可複雜如 TEI、EAD、CIMI；SGML是完全可擴展的（extensible），可以針對各種類型的文件結構訂定出合適的標籤集（tagset）；SGML是理想的資料儲存格式，提供了相當多的選項功能，可以適用於最複雜的資訊處理。

2.非專屬性（non-proprietary）、平台獨立（platform-independent）與系統獨立（system-independent）：SGML並不專屬於特定的平台與特定的應用系統，因此SGML文件可以在彼此不相容的系統間交換，不會造成資訊遺失（information loss），這個特性使得SGML文件可以長久保存。

3.資訊再利用性（re-usability）：透過 SGML 文件內容模組（content module）的再利用，使得文件的產生更有效率、更經濟，SGML文件的內容可以重複利用，或者被其他的SGML文件使用，不須重新產生內容。同一份文件內容也可以透過樣式表（style sheet）以多種呈現方式出版。

二、SGML的限制：（註15, 16, 17）

1.SGML應用程式不易開發：SGML龐大且複雜的選項功能雖然使得 SGML 具有較高的彈性，但也增加了應用程式開發上的難度，即使 SGML 工具的主要供應廠商 ArborText 所發表的產品，也沒有百分之百支援 SGML 標準。事實上，SGML 有許多選項很少被應用，如果把這些不常用的選項去掉，將使得應用程式的開發變得更容易。

2.SGML文件不易在Web上傳佈：要能夠瀏覽SGML文件，必須要有文件型別定義（Document Type Definition；簡稱DTD）及樣式表（Style Sheet）。DTD定義了文件結構間的關係，樣式表定義了這些結構的呈現格式，如果少了DTD與樣式表就只能看 SGML 文件的原始碼了。由於目前Web上的主流瀏覽器只支援HTML，HTML文件並不需要DTD與分離的樣式表，因此SGML文件在Web上只能透過特定的瀏覽器(如Panorama)才能閱讀，不過這類的瀏覽器並不普及。如果希望SGML資訊能在Web上被大多數人瀏覽，只好透過轉換程式將SGML轉成HTML，但這樣的轉換往往會造成資訊遺失（Information Loss），原本SGML文件中所標示的結構在轉換成 HTML文件後並無法繼續存在。

3.缺乏廠商的支援：Web上的主流瀏覽器廠商 Microsoft 與 Netscape 支持HTML的發展，但並沒有支援SGML的意願；由於SGML過於複

雜，也只有少數廠商願意投資開發 SGML 的相關應用程式，這使得 SGML 在普及上造成很大的障礙。

肆、HTML

HTML是SGML最著名的應用，是一種專爲WWW網頁顯示及瀏覽而設計的簡易標示語言，目前是WWW上製作網頁的標準語言格式。WWW的開山祖師 Tim Berners-Lee 對 HTML 所下的定義是：「HTML是一種用以創造超文件的簡易資料格式，其所創造出來的文件可在不同的作業平台間移動。」由此可知，可攜性(portable)與簡易性（simple）是HTML的兩大特徵。（註18）HTML文件除了包含文字資訊外，尙可包括聲音、影像等多媒體資訊，而HTML的超鏈結除了網頁內的鏈結，也包括網頁之間的鏈結。以下介紹HTML的特色與限制。

一、HTML的特色：（註19）

1.HTML DTD的設計主要是滿足線上顯示的需求。許多標籤純粹是用來指定網頁的呈現細節，例如：
- br 是換行(line break)。
- hr 是畫一平行線(horizontal rule)
- b 表示粗體字(bold)。
- i 表示斜體字(italic)。

2.HTML有內建的樣式（style)。HTML希望直接使用SGML標示來控制網頁呈現的樣式，亦即HTML不需要有分離的樣式表，這使得HTML更爲簡單易用，但缺點是較沒有彈性，同樣的文件內容只能設定一種呈現樣式。

3.HTML引用SGML的標示最簡化特徵(markup minimization feature)。這是爲了盡量減少HTML文件標示的數量，使HTML的標示更簡單，最明顯的例子是結束標籤(end-tag)的省略。例如在 HTML 2.0 DTD 中，p（paragraph）的結束標籤是可以省略的，其元素型別（element type）的宣告如下：

<!ELEMENT p -O (%text)*>

其中 "-" 表示起始標籤（start-tag）是必備的，大寫字母 "O" 表示結束標籤是可省略的（ "O" 是 "omissible" 的意思）；（註20）倘若結束標籤是必備的，則元素型別宣告應該如下：

<!ELEMENT p -- (%text)*>

結束標籤的省略雖然使得HTML的標示更爲簡單，但也成爲描述文件結構時的限制，HTML文件中所有的段落標示<p>在地位上都是平行的，省略了結束標籤就無法描述「段落中又有段落」的結構。

4.HTML 沒有採用 SGML 的超鏈結（hyperlinking）機制。HTML的超鏈結是利用標籤 <a>（a是anchor的意思）的 "href" 屬性來指向Web上的任意文件，例如：

<a href="http://www.w3c.org/"> 是一個外部鏈結（external link）

< a href = " # top " > 是內部鏈結（

（internal link）

而內部鏈結的目標點是利用標籤 <a> 的 "name" 屬性來指定，例如：

<a name="top">

SGML則是利用ENTITY或ENTITIES屬性型態及 ID－IDREF機制來達到超鏈結的效果。（註21）

二、HTML的限制：（註22）

1.結構上的限制： HTML 最大的限制就是它的標籤集是固定的，而這些標籤大都屬於呈現導向（presentation-oriented）的標籤，主要用來指定網頁的顯示格式，這個特性使得HTML只能支援固定且簡單的文件結構，而且在資訊再利用、資料交換與自動文件處理方面都造成很大的限制。

2.資訊再利用的限制：許多企業組織都有需要將相同的資訊以不同的形式來出版，例如印刷版本、CD-ROM版本、Web版本等，尤其隨著電子出版時代來臨，數位化資料不管在複製、編輯、傳佈上皆較傳統出版來得便利，將同樣的資訊以各種不同的形式出版也變得更可行。如果以HTML作爲電子出版的資料格式，設定不同的呈現格式，如標題字體的大小、條列（lists）與表格的使用等，就能產生不同的Web版本，如果列印出來就是相對應的印刷版本，但由於HTML文件的資料內容與呈現外觀是結合在一起，如果原始文件的內容有所改變的話，所有不同形式的版本全部都要跟著轉換，這道轉換的程序必須耗費不少的人力與時間。如果採用SGML作爲電子出版的資料格式，由於資料內容與呈現外觀是分開處理，因此可以避免掉因原始文件內容改變而造成所有的版本都必須轉換的問題。

3.資料交換的限制：由於Web的普及，上網人口不斷增加，使得Web成爲許多企業組織交換資料最理想的場所，但由於 HTML 的標籤集是固定的，且這些標籤大都屬於呈現導向的標籤，利用 HTML 作爲資料交換的格式，很難對每一項所要交換的資料作清楚的描述。例如：有一家網路書店想要透過Web從出版商那裡取得一些新出版書籍的書目資料，並希望把這些資料自動轉入自己的資料庫中，再動態地把新書資訊呈現在網站上，書目資料包括了作者、書名、出版社、ISBN…等欄位，以 HTML 標籤來標示這些書目資料，並沒有辦法逐一標示每個欄位，通常是把它包裝成表格的形式，以利於瀏覽，但如此一來卻沒有辦法利用程式將 HTML 中的書目資料轉入資料庫中，因爲程式沒法分辨HTML檔中哪一段資訊是作者、哪一段資訊是書名…，就算出版商以SGML來儲存書目資料，清楚地描述每一個書目資料

的欄位，但一旦要透過Web傳送，將SGML轉成HTML後，這些書目資料的欄位結構就無法存在了。

4.自動文件處理的限制：自動文件處理可節省人力操作的成本，降低人工輸入的錯誤，改善整體作業流程的品質，並提高文件傳遞的速度。透過Web有許多的文件處理流程事實上可以被自動化，尤其在Intranet或Extranet若能將資料庫中的資料轉出後作自動處理，或文件經過自動處理後直接轉入資料庫，將能大幅提高效率。目前在Web上一些表單應用程式就是自動文件處理簡單應用，如有些線上問卷系統或線上投票系統，使用者將填完的問卷資料直接傳入伺服端的資料庫後，可以直接即時讀取資料庫的統計結果。由於HTML的標籤集是固定的，而這些標籤大都屬於呈現導向的標籤，因此HTML文件所能做的自動化處理事實上有很大的限制。所有文件處理高度自動化的流程，都必須透過統一的資料格式，而且這個資料格式必須能攜帶豐富的內容語義，從這個角度來說，HTML並不是一種適合作自動文件處理的資料格式。

5.無法支援較精確的查詢：目前在Web上使用者可以透過搜尋引擎（search engine）所提供的關鍵詞查詢（key word search）來尋找相關的資訊，但由於目前Web上的資訊不斷增加，使得搜尋引擎的查詢結果往往會找到太多的資訊，而這些資訊又不一定能符合自己的資訊需求，往往使用者花在過濾出所需資訊的時間，會超過真正去看這些資訊的時間。搜尋引擎的準確率（precision rate）不高是因爲所用的查詢模式是對網頁進行全文檢索，雖然也可以將搜尋的目標限制在HTML文件的Title部分來提高準確率，但這樣又會降低查詢的回收率（recall rate）。一個較好的辦法是提供可以指定內容（content-specific）的標籤，例如：<author> 莊子 </author> 與 <title> 莊子 </title> 便有所區分，如此一來查詢時便可做較精確的限定，其效果有如欄位化查詢一般，不過 HTML 並無法讓網頁製作者自行定義可以指定內容語義的標籤，XML將能解決這個問題。

6.HTML的不斷修訂造成了許多網站維護的額外工作：由於HTML是一個演進中的標準，每當HTML的標籤集不能滿足需求時，W3C就會爲HTML加入新的標籤，推出新的HTML版本。從HTML 2.0到HTML 32，再到HTML 4.0，每當新的HTML版本推出，一些必須維護大量HTML文件的單位就得重新回頭檢視這些舊版的HTML文件，看看有沒有需要重新標示文件。除了W3C會以官方立場身分修訂HTML外，瀏覽器大廠 Microsoft 以及 Netscape

也會伴隨著新版的瀏覽器推出自己的HTML延伸標準，而兩家廠商推出的延伸標準又不完全相容，對於許多網站維護人員來說，每當有新版的瀏覽器問世，就代表著可能又要對部分的網頁重新標示。有些組織爲了徹底避免重新標示文件的困擾，乾脆決定採用SGML來標示文件，再把SGML轉換成HTML，因爲將 SGML 轉成 HTML 只要透過轉換程式批次進行並不需花費太多資源，但若要重新標示成千上萬的HTML網頁就工程浩大了。

伍、XML

XML 是 W3C 在 1996 年底提出的標準，它是從 SGML 衍生出來的簡化格式，也是一種元語言（meta-language），可以用來定義任何一種新的標示語言。XML的制定是爲了補足 HTML 的不完美，使得在 Web 上能夠傳輸、處理各類複雜的文件，它去除了 SGML 複雜不常用及不利於在Web傳送的選項功能，讓使用者可以很容易地定義屬於自己的文件型態，程式設計師也能在更短的時間開發 XML 相關應用程式。（註23）XML 1.0 Recommendation 已於 1998 年 2 月公佈，相關標準目前仍在發展之中，XML的發展獲得了各界的支持，其中包括了Sun Microsystems, Microsoft, Netscape, Adobe, ArborText……等軟體大廠的背書。（註24）

一、XML 的發展背景：

XML的發展背景主要是因爲HTML的諸多限制已經影響了WWW的發展，HTML的限制在前面已經介紹過了，XML的發展成員大都對SGML及結構化的資訊（structured information）有相當豐富的應用經驗，他們相信引進SGML技術，能夠彌補HTML的不足，對WWW的發展能有以下幾個方面的貢獻（註25）：

1. 電子資料交換（EDI）：結構化資訊的一個主要應用是資料交換，不同的領域可以針對領域的特性制定共同的資訊內容模型（content model），並以這個共同的內容模型來標示資訊，如此可以促使同領域的資訊可以更容易且更有效率地交換，這個共同的內容模型，我們稱之爲DTD。無疑地，Web是理想的電子資料交換的媒介，但HTML並非理想的資料交換格式，也難以充分地表現各種資訊內容模型與語意結構，而XML所要提供的正是一套可以在Web上承載各種結構化資訊的框架。
2. 與Java技術更緊密結合：Java技術的出現使得瀏覽器能成爲通用的應用系統平台，但HTML固定的標籤集及不擅長描述語義的特性，使得Java程式沒有太大的發揮空間，而XML正好可以給予Java程式大顯身手的環境，以XML作爲各種結構化資訊的標準格式，搭配上Java

程式，可以使得應用程式大部分的運算得以在客戶端執行，這和目前大部分的Web-based應用程式主要透過伺服端的CGI scripts來完成大部分的運算是相反的模式。藉著XML與Java技術的結合，將應用程式的運算從伺服端移到客戶端來，有助於降低網路的流量與增加網路的速度。

3.攜帶平台獨立（platform-independent）資訊：HTML與XML的始祖SGML提供了一套能夠指定資訊的結構與語義的語法規則，而且具備了平台獨立性。不像 Microsoft 的RTF、Adobe的PostScript以及其他專屬性的文件格式，SGML所提供的是一套具備平台獨立性與系統獨立性的語法規則。

二、XML的設計目標

根據 XML 1.0 Recommendation（註26），XML 的設計目標如下：

1.XML將能直接在Internet上使用。
2.XML將支援各種不同的應用。
3.XML將與SGML相容。
4.處理XML文件的程式能很容易被開發。
5.XML的選項功能將保持最少，最好是零。
6.XML文件應該是易讀且清晰的。
7.XML的設計應該很快就緒。
8.XML的設計將是正式且簡潔的。
9.XML文件將很容易被產生。
10.精簡對於XML標示來說是最不重要的。(HTML的標示便盡求精簡)

三、XML與SGML主要的不同(註27)：

1.DTD不是必備的：要處理SGML文件必須要有DTD，而對於XML文件而言，DTD不是必要的。爲了要使XML文件在處理上不需透過DTD，XML文件必須遵守更嚴謹的語法規則。對於XML應用程式而言，沒有DTD的好處之一是可以節省下載DTD所用掉的頻寬，以及應用程式在開發時可以不必考慮解譯DTD的模組。

2.必須符合特定的語法規則（Well-formedness）：雖然XML文件可以不需要有DTD，但每一個XML文件都必須是Well-formed的形式，所謂Well-formed的意思是必須遵從XML所定義的幾條語法規則。例如：一份XML文件至少要有一對標籤；所有的元素（elements）必須是巢狀的(nested)結構，而且標籤必須是成對的，也就是每個元素都要有起始標籤（start tag）與結束標籤（end tag）；任何被引用的實體（entities）一定都要先宣告。這些強制性的語法規則使得開發XML的相關應用程式能夠更簡單，不須像SGML的應用程式一般，必須參照DTD進行文件結構的確認（validation）。

3.不支援例外處理（Exceptions）：SGML的使用者可以使用包含 (In-

clusion)與除外（Exclusion）這兩種語法規則來指定內容模式（content model）的例外處理，例如：可以利用除外（Exclusion）規則來控制附錄的文字中不能有附錄參照。例外處理的功能對於一些無法處理不可預期結構的應用程式很重要，由於XML並不支援例外處理，這使得現存許多包含例外處理的 SGML DTD一時還不能以XML來取代。

4.不支援AND內容模式 (content model)：XML並不支援AND(&)內容模式。AND內容模式主要用來控制一群指定的元素必須同時出現，而出現的順序是任意的。例如：（A&B&C）的意思是A, B, C必須同時出現，但可以是任意的順序，這種內容模式主要是要求一群元素的完整性。XML不支援AND內容模式，對於一些必須使用AND內容模式來控制特定元素群的完整性的SGML DTD，也無法很快被XML取代。XML可提供兩種是較接近（A&B&C）的簡單內容模式：一種是結構較鬆散的 ((A|B|C)+)，另一種是嚴格限制出現順序的（A, B, C）。其實AND內容模式可以用其他的內容模式組合成相同意義的結構，只是表示法可能會相當複雜。

5.不支援 SDATA 內部實體（internal entities）：SGML允許使用者利用SDATA內部實體來定義特定系統（system-specific）的符號，如一些數學符號。XML並不支援這種機制。

四、XML與HTML主要的不同：

1.資訊提供者能任意定義新的標籤與屬性名稱。

2.文件結構可以是任意階層或複雜的巢狀結構（nested structure）。

3.XML 文件可以包含文法（grammar）的選擇描述，讓必須執行結構確認（structural validation）應用程式使用。（註28）

4.XML不像HTML只有內建的樣式，XML提供了樣式表標準，稱爲可擴展樣式語言（Extensible Style Language；簡稱XSL）。（註29）

5.XML除了支援像HTML的簡單鏈結（simple link），也提供了幾種功能更強大的超鏈結機制。XML的超鏈結機制被制定爲XML鏈結語言（XML Linking Language；簡稱XLink）（註30）與XML指標語言（XML Pointer Language；簡稱XPointer）。（註31）

五、XML的樣式表

XML提供的樣式表標準，稱爲XSL，1998年12月發布1.0版草案，簡介如下：（註32）

1.以DSSSL（Document Style Semantics and Specification Language）爲基礎：SGML 成爲國際標準之後，支援 SGML 的樣式表（Style Sheet）標準就開始被發展，這些

標準的制定主要是爲了促進樣式表的交換與改善處理文件的軟體之間的互通性（interoperability）。這些樣式表標準中最著名的就是DSSSL，後來DSSSL也被建議成爲ISO的標準，不過 DSSSL一直沒有得到商業軟體的支援。

2.與CSS（Cascading Style Sheets）相容：CSS是Microsoft與 Netscape所支持的樣式表標準，作爲HTML預設樣式的替代機制。由於HTML的樣式是內建的，並沒有提供樣式表，透過CSS就能使得HTML在顯示格式上有較大的彈性。XSL將在功能上涵蓋 CSS 的功能，並且使能從 CSS 透過程式自動轉換到XSL，如此一來現有以CSS所設定的格式不至於重新來過。

3.具備重新排序（Reordering）的能力：藉著XSL樣式表，不需透過伺服端程式的重新處理，在客戶端的瀏覽器上就能改變資料呈現的順序，這個特色對於一些藉著將資料以任意的順序呈現來達成互動性的應用程式特別有用，而且由於重新排序的動作全部在客戶端就可實現，不需透過伺服端與客戶端的來回通訊，有助於節省網路頻寬與避免因網路速度太慢影響了應用程式執行的反應速度。

4.對於文件中元素（elements）的前後文關係（context）更加靈敏：CSS支援對每一個元素的父元素（parent）設定樣式；XSL更進一步允許對每一個元素的祖先元素（ancestors）、後代元素（descendants）、兄弟元素（siblings）設定不同的樣式，這對於以文件中元素的位置及前後文關係來設定樣式提供了更大的彈性。

5.同時支援線上顯示與列印的格式：CSS只有支援線上顯示，XSL除了支援線上顯示的格式外，也支援文件列印能有更豐富、更複雜的格式。

六、XML的超鏈結機制：

XML 的超鏈結機制被制定爲XLink 與 Xpointer兩個標準，1998年3月各發佈了 1.0 版草案，簡介如下：（註33）

1.基於HyTime（Hypermedia／Time-based Structuring Language）與TEI（Text Encoding Initiative）的鏈結概念：XLink與XPointer 的鏈結概念引用自 HyTime與TEI，這些標準目前尚未被軟體廠商所普遍支援。XML將提供幾種比目前HTML的超鏈結機制更強大、更有彈性的鏈結機制。

2.與現有的URL 鏈結機制相容：XLink 將完全支援現有的 Web的URL鏈結格式。

3.支援雙向鏈結（Bi-directional Links）：雙向鏈結允許使用者能在鏈結的兩端自由來回跳躍。目前HTML的單向鏈結在使用時，有時

經過幾次的超鏈結跳躍後會有「迷路」或「找不到回家的路」的情況發生，透過雙向鏈結的機制，這些情況可以改善一些。

4.支援定址（Addressing）：XPointer允許鏈結到目的文件的階層結構的某個精確位置，也就是可以利用文件的結構來定址，這是比較有彈性的鏈結方式。 HTML 的文件內部超鏈結必須指定文件中的絕對位置，一旦文件內容有所改變，通常必須更新鏈結；利用文件結構來定址所指定的是相對位置，只要文件結構沒有改變，就算文字內容改變了也不需更新鏈結。

5.支援間接鏈結(Indirect Links)：間接鏈結可以改善目前Web上很普遍的斷裂鏈結（broken links）問題，現在Web上所使用URL（Uniform Resource Locator）是屬於絕對位址，只要主機位置或路徑改變，所有包含這個URL的檔案都必須更新，否則就會造成斷裂鏈結，這對使用者與網站管理者都造成一些困擾，就算能利用程式自動偵測每個鏈結的狀況，產生斷裂鏈結的清單，但要處理這些斷裂鏈結也需耗費不少人力，尤其是一些大的網站，可能定時得處理斷裂鏈結問題，否則便會遭來使用者的抱怨。 XLink 的間接鏈結所採用的位址是間接位址，再透過一個分離的中介檔（intermediate file）來儲存間接位址與實際位址的對應，如果某個檔案改變了在網路的位置，只需更新這個中介檔的對應，至於鏈結的原始檔案與目的檔案都不需更動，這使得鏈結的管理更有效率。

陸、結語

對於電子圖書館等必須大量收集、組織、儲存、傳佈大量數位資料的單位而言，選擇適當的資料格式是極爲重要的，除了必須考慮到長期保存、交換的需求外，這個資料格式應該能支援更精確的檢索及各種不同的再利用，SGML已被幾個知名的電子圖書館相關計劃採用作爲電子文件格式，如TEI、EAD、CIMI、DIAP與American Memory Project等。但無法在Web上廣泛傳佈一直是SGML最大的限制，透過XML的制定，將突破這個限制，使得電子圖書館所典藏的各種結構化資訊也能透過Web廣泛地傳播出去，資訊唯有被廣泛利用才能成就它的價值。此外在文獻傳遞、分散式查詢、資訊過濾等方面，XML也將分別帶來效益，且讓我們拭目以待。

註釋

註1：Yves Marcoux and Martin Sevigny, "Why SGML? Why Now?," Journal of American Society for Information Science 48:7 (July 1997): 584.

註2：Lou Burnard and Richard Light, "Three SGML metadata formats: TEI, EAD, and CIMI - A study for BIBLINK Work Package 1.1", Dec. 1996, <http://hosted.ukoln.ac.uk/biblink/sgml/toc.html>.

註3：Edward Gaynor, "From MARC to Markup: SGML and online Library Systems", May 1996,<http://www.lib.virginia.edu/speccol/scdc/articles/alcts_brief.html>.

註4：Peter D. Verheyen, "SGML, Metalanguage with no limits?", IST 667 Literature Review.

註5：同註3。

註6：Arbortext,"SGML:Getting Started," <http://www.arbortext.com/Think_Tank/SGML_Resources/getting_started_with_sgml.html>.

註7：Jon Bosak, "XML: The new standard for Web data," SGML Solution World, July 1997, <http://ftp.cycu.edu.tw/UNIX/sun- info/xml/pres/9707ja>.

註8：Richard Lander, "XML: The New Markup Wave" May 1997, <http://www.csclub.uwaterloo.ca/u/relander/>.

註9：鄒宏德，「SGML文件資料庫系統」（碩士論文，國立台灣大學資訊工程系研究所，民國84年6月）。

註10：曾士熊，「認識SGML㈠—㈢」，計算中心通訊12:24-26（民國85年11-12月）。

註11：陳昭珍，「標準通用標示語言基本概念」，圖書與資訊學刊（民國85年3月），頁37-56。

註12：陳昭珍，「SGML與電子出版」，資訊傳播與圖書館學 3:3（民國86年3月），頁49-63。

註13：許麗娟，「SGML、HTML及VRML之比較研究」，在資訊科學與技術專題論集（台北市：文華，民國86年），頁131-182。

註14：同註8。

註15：Arbortext, "XML for Managers", <http://www.arbortext.com/Think_Tank/XML_Resources/xml_for_managers.html>.

註16：同註8。

註17：Richard Light, Presenting XML (Indianapolis: Sams.net, 1997), 20.

註18：Lois Patterson 原著；孫昱編譯，HTML 最新版教戰手冊（台北市：文魁資訊，民國87年），頁1-2。

註19：同註17: p.10-14.

註 20 ： Ronald C. Turner, Timothy A. Douglass and Audrey J. Turner, README.1ST: SGML for Writers and Editors (New Jersey: Prentice Hall PTR, 1996): 81.

註 21 ： 同前, p.119-120.

註 22 ： 同註 15。

註 23 ： 梁中平、徐千惠，「取SGML之長，補 HTML 之短－新一代標示語言XML」，民國86年11月，CALS Web Site<http://www.cals.org.tw/files/cals1-4.htm>.

註 24 ： 同註 15 。

註 25 ： Todd Freter, "XML: Mastering Information on the Web," (SUN) <http://www.sun.com/980310/xml/>.

註 26 ： W3C XML Working Group," Extensible Markup Language (XML)1.0," Feb 1998, <http://www.w3.org/TR/1998/REC-xml-19980210.html>.

註 27 ： 同註 15 。

註 28 ： Jon Bosak, "XML, Java, and the future of the Web," (Sun Microsystems: Mar. 1997)<http://sunsite.unc.edu/pub/suninfo/standards/xml/why/xmlapps.htm>.

註 29 ： W3C XML Working Group, " Extensible Stylesheet Language (XSL)," Dec. 1998, <http://www.w3.org/TR/WD-xsl>.

註 30 ： W3C XML Linking Working Group, "XML Linking Language (XLink)," Mar. 1998, <http://www.w3.org/TR/WD-xml-link>.

註 31 ： W3C XML Linking Working Group, "XML Pointer Language (XPointer)," Mar. 1998, <http://www.w3.org/TR/WD-xptr>.

註 32 ： 同註 15 。

註 33 ： 同上 。