

## XML 相關技術與下一代 Web 出版趨勢之研究

林信成

### A Study on XML-based Technologies and Next-Generation Web Publishing

*Sinn-cheng Lin*

*Assistant Professor*

*Department of Educational Media & Library Sciences*

*Tamkang University*

*Taipei, Taiwan, R.O.C.*

*<E-mail: sclin@mail.tku.edu.tw>*

#### Abstract

XML was completed in 1998 by the W3C. This paper focuses the attention on the issues of the XML-based Web publishing. First of all, the paper describes the essential primaries of the electronic documents, analyses the situation of the Web publishing, explore the difficulties and bottlenecks of HTML. Furthermore, based on the purposes of document access, integration, delivery, manipulation and display, the paper proposes a 3-tier distributed architecture for document management. Which maps the related technologies of XML family, such as DTD, XML Schema, XML Namespaces, RDF, XLink, DOM, CCS and XSL, etc., to the corresponding tiers. It also reflects that XML will plays an important role in the Web integration. If we say that the HTML was the first revolution of the Web, then the XML could be viewed as the second-generation. XML not only provides a new mechanism for data representation, but also has the potential to extend the Internet beyond information delivery to knowledge management. Expectably, XML will cut a generous swath across various application fields, such as electronic publishing, electronic commerce, electronic library, electronic data exchange, and distance learning, etc.

#### Keywords:

XML; HTML; Web publishing; Electronic document; Electronic publishing

## 前 言

二十世紀是個偉大的年代，電腦、網路、多媒體，各種電子傳播科技的發明與普及，不但消滅了資訊傳播在時間上的藩籬，更縮短了空間上的距離，加速促使全世界變成了一個名符其實的「地球村」(global village)。尤其進入公元兩千年之後，資訊與通訊科技的進展更有如加了順向偏壓(forward bias)的二極體(diode)一般，呈現一發不可收拾的指數成長趨勢(註一)，把整個人類的科學文明推向前所未有的高峰。「指數率」雖是物理上的一種自然現象，套用在人類科技文明的進程上，卻也有幾分道理，莫爾定律(Moore's Law)正好印證了這個趨勢(註二)。在這一波翻天覆地的新科技浪潮中，網際網路(Internet)無疑的是最重大的成就之一。根據「Network Wizards Internet Domain Survey」的統計，Internet的主機數量近年來不斷呈指數成長，於2000年1月已經高達約七千兩百萬部(註三)，儼然成爲一個超強勢的新媒體。而在Internet衆多資訊服務中，最受矚目的當屬全球資訊網(World Wide Web，以下簡稱WWW或Web)。如今，WWW幾乎已經成了Internet的代名詞，甚至對於許多使用者而言，WWW就等同於Internet。

隨著網路的發展，圖書與出版技術也逐漸由傳統印刷形式，朝向全面電子化的方向邁進，藉由網路，人類的文化得以採用數位化型態即時傳播；透過網路，全球的使用者皆可共享資訊、共用資源。從電子出版的角度而言，Web爲一個建構於Internet上的超文件(hypertext)出版與管理系統，Web的發展使得超文件觀念普植於人心(註四)。如今，經由Web產生的電子文件已如同蜘蛛網一樣密密麻麻組成了一個全球性分散式超文件系統(Distributed Hypertext System)。在此系統中，HTML(HyperText Markup Language，超文件標注語言)無疑的扮演了關鍵性的角色。HTML是一種簡化版的標注語言，也是建立超文件的標準規範(註五)。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的確爲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，能夠有效表達網路上各種知識，為資料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。然而，XML基本語法雖然簡單，卻由於其可擴展性，目前已發展完備或正發展中的相關技術，如XML Namespaces、RDF、XLink、XPointer、CSS、XSL、DOM、SMIL、MathML……等如雨後春筍般出現，儼然蔚為一支XML大家族，使得XML的複雜度及學習難度遠遠超越HTML。

本文試圖從電子出版的角度著眼，說明電子文件組成要素，分析目前Web出版的現況，從而深入瞭解HTML所遭遇的困境與瓶頸，接著基於文件存取、整合、傳遞、操縱、顯示等議題，提出一個分散式文件管理架構，並探討XML相關技術在此一架構所扮演的角色及功能，進而描繪出Web未來發展之趨勢與輪廓。

## 二、Web出版的現況與困境

### (一)電子文件的要素與分類

一般而言，電子文件具有三項重要的構成要素：

#### 1. 資料

資料(data)指文件的實質內容(content)，包括該份文件所有的文字、圖形、表格、聲音、影像……等內涵部分。

#### 2. 結構

結構(structure)指文件的組織架構和階層關係，如題名、作者、章節、段

落等。

### 3. 表現

表現(presentation)指文件中的資料經由各種展示媒體(如螢幕、印表機、點字機、喇叭等)所呈現出來的外觀、樣式、版面編排、聲響。

若將文件與人體作一類比，則結構就如同人的骨架，資料則相當於肉體，而表現就形同外貌，三者相輔相成，形成一個完整的資訊描述體系，因此本文特將此三者合稱為「電子文件三要素」。

電子文件依其結構性可概分為三大類(註六)：

#### 1. 完全結構化文件

結構性完整的文件稱為完全結構化文件(fully structured document)。例如，在關連式資料庫(relational database)的關連表(relation table)所儲存的每一筆記錄(record)，都有明確的欄位(field)定義，每個欄位的資料型態(data type)和大小也都清清楚楚，這便是完全結構化文件；又如利用 MARC 所著錄的書目資料、用 Dublin core 所描述的網路資源等，也都屬於此類。由於文件中的結構化資訊是一種 Metadata，記載著有關資料的資料(the data about data)，因此這一類型的文件非常適合進行欄位檢索。

#### 2. 完全非結構化文件

毫無組織的資料或毫無結構可循的文件，稱為完全非結構化文件(fully unstructured document)，如遙測、監控數據、聲音、影像資料、缺乏段落結構的全文資料……等。這一類型的文件在儲存時較為簡單，然而由於欠缺文件結構資訊，使得檢索技術相對困難。

#### 3. 部份結構化文件

大部份的文件都是介於以上兩個極端之間，屬於部份結構化文件(partial structured document)。例如，一般文獻中的題名、作者、摘要、章節段落等都屬於文件的結構化資料，段落中的本文資料則屬於非結構化資料；而 HTML 使用標籤來標注文件內容，也屬於部份結構化文件。

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，則只要使用適當的「表現」方式來展現出文件的「資料」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。

## (二)HTML的限制

HTML是從SGML (Standard Generalized Markup Language, 標準通用標注語言)演變而來。SGML發表於1986年,是一個電子文件製作及交換的標準(ISO 8879)。它藉由DTD (Document Type Definition, 文件型別定義)來定義標籤及規範文件結構,並以其所定義的標籤來標注文件的內容,以確保電子文件能在不同的電腦系統上進行交換與應用。SGML將文件裡的每個元素區分成邏輯的、可預測的結構,形成一個資訊描述體系,具有跨平台、結構化、可移植、可擴展……等特性,建立了文件共享的基礎。但由於SGML太過龐大複雜、不易學習及使用,因此並未獲得普及。有鑑於此, Tim Berners-Lee於1989年提出WWW概念時,乃根據WWW的需求而以SGML定義了適合超文件使用的HTML語言。由於HTML具備規格開放、易學易用、語法簡潔明瞭等特色,加上獨特的超連結功能,使得HTML在短短數年內迅速普及,成為寫作Web文件的主要格式。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:

1.HTML著重在資料的表現,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、結構和表現三要素交雜在一起,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(cascade style sheets)將外在表現部份獨立於結構化資料之外,可以有效的改善此一問題,但至目前為止似乎成果有限。

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。

由於HTML有這些缺點,所以發展出許多改善方案來提昇其性能,例如藉助動態HTML (dynamic HTML)、客戶端和伺服器端的各類型Script程式、CGI程式和外掛程式(Plug-Ins)等擴增功能,可以完成動態網頁展現、資料庫

存取、通訊協定轉換、網路資源整合等，這是近幾年 Web 大放異彩的主要原因。不過，這些基於 HTML 先天缺陷所提出的解決方案，卻也是促使 HTML 文件愈來愈「不純」的罪魁禍首。現今，在一份 HTML 文件中，極可能除了文件本文和有關的結構、樣式等標籤之外，尚且夾雜了眾多與資料無關的字型、顏色、外觀等標籤及屬性，再加上 Script 程式碼和各家瀏覽器廠商自訂的稀奇古怪功能，怎堪一個「亂」字了得。

資訊化社會所感困擾的不是資訊的匱乏，而是資訊的氾濫。如今，以 HTML 為主的電子文件充斥在網路上，其內容又亂象叢生，不但造成電子資料的氾濫，更引發了資訊檢索的問題。換言之，由於 HTML 過度著重於形式的呈現，而忽略了以結構化方式組織資料，使得在網路上進行資料搜查及資料交換時顯得雜亂而缺乏效率。因此，為了讓使用者在浩瀚的網海中能順利查找到所需資料，資訊檢索技術在這幾年內有許多突破性的進展，而基於資訊檢索技術的搜尋引擎(search engines)，順理成章的也就成了網路使用者查找資料的主要依據。

### (三) 資訊檢索技術的研究

網路所標榜的終極目標，是要使任何人能在任何時間、任何地點，都能獲得所需的任何資訊或服務(註七)，因此提供使用者一個有效的資訊檢索機制，便成為 Web 出版所應考慮的重要課題。

資訊檢索(Information Retrieval，簡稱 IR)技術歷經數十年的發展，累積了不少經驗與成果，如索引(indexing)技術(註八)、自動文件分類(automatic document classification)技術(註九)、全文檢索(full-text retrieval)技術(註十)、相關回饋(relevance feedback)(註十一)、自然語言處理(natural language processing)及跨語資訊檢索(cross-language information retrieval)技術(註十二)、多媒體資訊檢索(multimedia information retrieval)技術(註十三)等。近年來由於電腦科技的發展及各領域研究人員的投入，資訊檢索技術漸漸朝向跨領域整合方向發展，如資訊過濾(information filtering)、資訊擷取(information extraction)、資料挖掘(data mining)、智慧型代理人(intelligent agent)等，使得資訊檢索成了一個多元化的研究術域。學者 Michael Lesk 將資訊檢索技術的發展歷程，從 1945 年起以每十年為一年代劃分，每個年代都有重要的突破與進展，是瞭解近代資訊檢索技術發展的重要文獻之一(註十四)。

圖1為資訊檢索系統架構圖。其中，文件資料庫中的文件集合(document set)，經由文件分析過程，抽取足足以描述文件的特徵(feature)，以簡化後續的檢索處理程序及提昇檢索效能；當使用者輸入檢索條件時，查詢模組則針對使用者的檢索詞句進行查詢分析(query analysis)，並將此查詢映射(mapping)至文件空間(document space)中，以便進一步與文件進行檢索處理；檢索處理模組經由計算「查詢」與「文件」之間的「相似度」(similarity)而完成比對(matching)工作；檢索結果最後輸出至使用者的檢視裝置上；若使用者對系統檢索結果不滿意，可透過互動式人機介面(interactive men-machine interface)對系統進行相關回饋，以便修正檢索策略，逐漸逼近所需的結果。

在文件分析過程所抽取的文件特徵，是否具有足夠的代表性且能充分描述整份文件，對於整個檢索系統的效能有決定性的影響。回現率(recall rate)和精確率(precision rate)是評估檢索系統效能的兩大重要指標(註十五)，兩者經常無法兼得。君不見搜尋引擎動輒成千上百篇的檢索結果，回現率有餘而精確率不足，往往造成使用者資訊需求上的額外負擔；反之，若僅顧及精確率則又往往犧牲回現率。實際上，文件中的結構化資訊，經常是特徵抽取時非常重要的指標，資訊檢索系統在進行文件分析或檢索時，通常可以藉由文件中結構化資訊的輔助，簡化分析過程或提高檢索效能。舉例而言，當對文件進行自動關鍵字擷取(keyword extraction)時，題名的詞彙就比一般本文的詞彙來得重要許多。因此，加強文件的結構性，增加描述資料，對於簡化文件分析過程，提升檢索精確率有極大的幫助。以此觀之，在發展全文資訊檢索技術

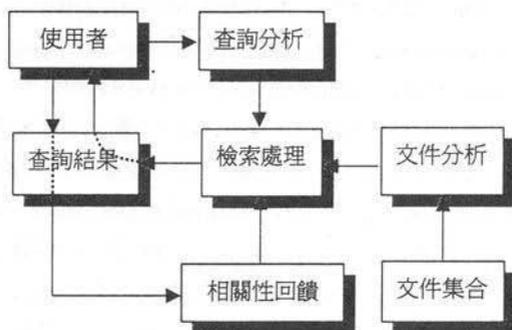


圖1 資訊檢索系統架構圖

之外，加強 Metadata(註十六)的著錄，不失為另一個解決網路上電子文件亂象並提昇檢索精確率的有效方案。

#### (四)Metadata 的發展

Metadata 是個極為普遍的概念，在日常生活中，四處可見 metadata 的蹤影：例如可以用 {CPU 型號、記憶體大小、硬碟機容量……} 這一組 metadata 來描述每部個人電腦的規格，所以我們可以很清楚的知道配備為 {Pentium-II350、64MB RAM、6.4GB HDD……} 的 A 電腦，比起配備為 {Pentium 133、16MB RAM、1.5GB HDD……} 的 B 電腦來得高檔許多；而對於出版品資料則可用 {書名、作者、出版社……} 這樣的 metadata 來加以描述。為了讓 metadata 發揮更大的功效，於是人們開始制訂各種 metadata 標準以供遵循。圖書館長期以來所沿用的機讀編目格式 MARC，就是用來描述書目資料的 metadata 標準。在網路盛行之後，為了因應既多且雜的電子文件，讓使用者都能盡快而且精確的找到所需資料，陸續制訂出來的 metadata 標準也就愈來愈多。

目前已發展成熟或正發展的 metadata 格式眾多，適用於不同領域及用途，例如早已普遍使用的 MARC 家族 (USMARC、UKMARC、CMARC)、PICA+ 等；專為描述科技文獻 BibTex、EELS、EEVL、RFC1807 等；專為描述人文社會科學資源的 ICPSR SGML Codebook Initiative、TEI Headers 等；專為描述政府資訊的 GILS Core Elements；專為描述地理空間資源的 CSDGM (Content Standard for Digital Geospatial Metadata)；專為描述博物館藏品與檔案特藏的 CDWA (Categories for the Description of Works of Art)、CIMI (Computer Interchange of Museum Information)、EAD (Encoding Archival Description) 等；以及專為描述大量網路資源的 Dublin Core (Dublin Metadata Core Element Set)、IAFA/WHOIS++ Templates、LDIF (Lighweight Directory Access Protocol)、SOIF (Summary Object Interchange Format)、URC (Uniform Resource Characteristics) 等(註十七)。

在現成的 HTML 規格中，大部份的標籤都著重於文件外觀及格式呈現，只有極少關於文件結構之描述者(如 <HEAD>、<META>、<BODY>……等)，加以 HTML 並不具備可擴展性，雖然可以使用 <META> 標籤在 HTML 文件中著錄 metadata，但仍不夠理想，XML 的誕生正好提供了一個可行的解決方案。

### 三、XML與下一代Web

#### (一)XML緣起與目標

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(註十八)。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都從SGML演變而來,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20%複雜罕用的部份,承襲了其他80%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」的重要技術實不為過(註十九)。

XML的發展將使得許多理想得以實現(註二十):

- 1.促進國際化媒體獨立(media-independent)的電子出版。
- 2.允許產業間定義平台獨立(platform-independent)的協定來進行資料交換,尤其在電子商務方面。
- 3.以某種形式傳送資訊給使用者代理程式(user agents),使其能在接收之後自動處理。
- 4.讓人們很容易便能使用平價軟體來處理資料。
- 5.允許人們以自己想要的方式顯示資訊。
- 6.提供metadata,不但能幫助人們找到所需資訊,更能幫助資訊生產者與消費者找到對方。

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

- 1.XML文件的作者可自訂標籤(tags)和屬性(attribute);HTML則否。
- 2.XML屬於一般用途的標注語言,而HTML則是一種特殊用途的標注語言。換言之,XML是一種元語言(meta-language),可用以生成其他語言;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。

### (二)XML 的可擴展性

XML 既可視為一種在 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)(註二一)。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。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，例如應用於數學方面的 MathML (Mathematical Markup Language) (註二二)、應用於向量圖的 SVG (Scalable Vector Graphics) (註二三)、應用於多媒體領域的 SMIL (Synchronized Multimedia Integration Language) (註二四)、應用於描述網路資源的 RDF (Resource Description Framework) (註二五)、應用於網路推播頻道的 CDF (Channel Definition Format) (註二六)……等。圖 2 清楚的說明了此一概念，展現了 XML 無限擴展的能力。

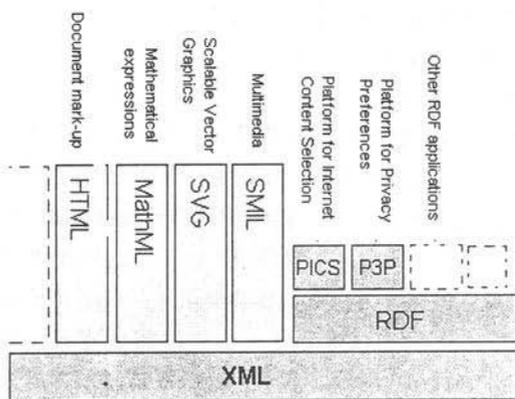


圖 2 XML 與應用語言(註二七)

### (三)XML 的自我描述性

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。圖 3 是一個簡易的 XML 文件，不論就人的或機器的觀點都很容易理解其資料的含意。

```

<?xml version="1.0" encoding="BIG5" standalone="yes"?>
<書籍>
  <ISBN>957-23-721-5</ISBN>
  <書名>精通 Visual Basic 6 程式設計 </書名>
  <作者>
    <姓名>林信成 </姓名>
    </服務單位>淡江大學 </服務單位>
  </作者>
  <出版社>第三波資訊 </出版社>
</書籍>

```

圖 3 一個簡易的 XML 文件

#### (四)XML的結構性

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優勢。這也是促使 XML 迅速成為重要機讀格式的主因之一。

XML 文件可根據其結構性概分為兩類(註二八)：

1. 完構的(well-formed) XML 文件。
2. 有效的(valid) XML 文件。

一個 XML 文件只要合於 XML 規範所制訂的語法規則，便可以很容易的具備「完構性」；不過，要達到「有效性」，則除了滿足語法規則外，尚必須遵循某些額外定義的語意才行。XML 處理器會根據 XML 規範所定義的完構性和有效性來查核文件內容，一旦發現不合規定的文件便會拒絕處理並告知使用者，這對於在網路上進行資料交換與資訊共享是非常必要的——凡不符合公認標準的資料可以不被接受。「文件類型定義」(Document Type Definition, 簡稱 DTD)和「XML 綱要」是 XML 處理器據以確定文件有效性的兩大重要機制。

#### (五)資料和樣式分離原則

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 文件的內在、外貌(即資料、樣式)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，依據不同需求來展現。如此一來，同一份文件或資料，將可在不同場合呈現出不同的風貌。

### (六)XML 所帶來的效益

總而言之，XML 至少將帶來如下效益：

1. 就資訊檢索方面而言，由於 XML 資料具有自我描述性質，因此可以提供語意層次的搜尋，避免全面性的盲目搜尋，進而提昇檢索結果的精確率。這在網路文件氾濫成災的今日尤其重要。

2. 對於系統的開發而言，由於 XML 具備可擴展性、資料與樣式分離等特色，各個系統可根據自身的需求，對 XML 資料進行其他加值處理。這使得 Web 應用程式(Web Application)發展更具彈性。

3. 由於 XML 文件高度結構化，使得 XML 可以很容易的和目前已經發展成熟的各種資料庫管理系統(DBMS)進行資料交換。這意味著來自不同管道的資料將可以輕易的藉由 XML 加以整合。

## 四、分散式文件管理模式

### (一)三層式架構

由於網路與資訊科技的發展，Web 出版逐漸地朝向「分散式系統」的方向邁進，而分散式系統又可區分為「同質性分散式系統」(homogeneous distributed system)和「異質性分散式系統」(heterogeneous distributed system)。在「同質性分散式系統」中，同一個機關、行號或企業組織可以跨越空間限制，如省、

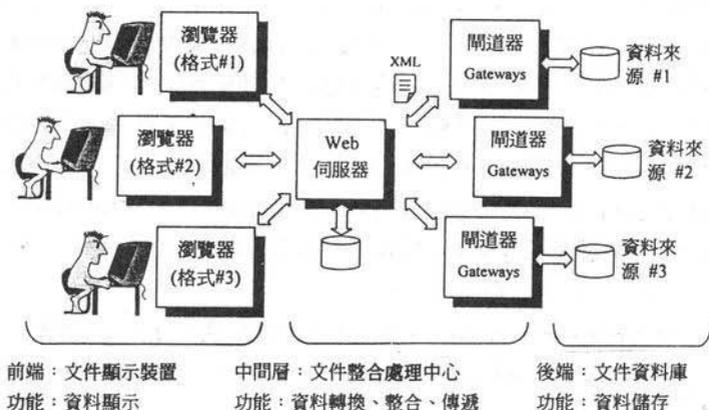


圖 4 分散式文件管理架構

市、縣，甚至國界，將其文件資料庫分散在網路上。它們能獨立作業完成區域性的應用，也能彼此關聯完成整體性的異動。在「異質性分散式系統」中，不同的機關、行號或企業組織的異業合作，使得原本各自獨立的異質性資料庫得以跨組織的完成整合處理，以達成整體資源共享的目標(註二九)。

一個基於XML的分散式文件管理模式，其架構如圖4所示。在此三層模式中，結構化的XML資料可以被分別儲存、處理和顯示。如同資料庫管理系統朝向分散式架構發展一樣，分散式文件管理系統勢必成為網路電子文件整合管理之標準架構。

以下便來探討XML相關技術在此一架構所扮演的角色及功能。

## (二)文件資料庫

文件資料庫是文件的儲存場所。這些文件來自作者、發行者或其他來源，分別以各種不同的檔案格式存放在各處的伺服器。這些伺服器本身可以透過文件的原始格式對其進行各種處理。原始文件可採用封閉式或私有性格式，雖然不影響本地處理，但若欲與遠端系統互相傳遞、交換、分享，則必須轉換成彼此可接受的格式。XML的標準化與開放性，正好提供一個電子文件交換(document interchange)的可行機制。XML文件必須是完構或有效的，而DTD或XML綱要正是定義文件結構的重要機制，也是驗證XML文件有效性的藍圖。DTD是沿襲自SGML的技術，主要用途在於定義文中的元素型態、結構、範圍、順序等，以作為撰寫XML文件的藍本。不過，由於DTD採用的語法規則與XML不同，不支援Namespace，所提供的資料型態相當有限，並且擴充能力不足，於是有必要發展一個類似DTD又能跨越其功能界限的機制，以作為定義文件結構的全新規範。是以W3C便在1999年2月15日發佈一份「XML綱要需求書」(XML Schema Requirements)(註三十)，說明DTD不足之處及新制訂XML綱要規範所必須符合的要求，並在1999年5月6日，首度發佈XML綱要工作草案，歷經多番修訂，預定在2000年中期提交W3C審議，以便成為建議標準(註三一)。

DTD比起XML綱要，具有如下優點，因此短期內仍有一定優勢：

1.廣泛的支援：所有SGML和大多XML工具程式都支援DTD，許多文件格式也都支援DTD。

2.豐富的經驗：DTD使用多年，已經累積許多寶貴經驗。

3. 眾多的應用：數以千計應用於各領域的 DTD，早已普遍存在供人使用(註三二)。

然而就長期趨勢而言，XML 綱要因具有許多 DTD 所不及的特色，預料將逐漸成為定義 XML 文件結構的重要技術：

1. 具有豐富的資料型態，如整數、布林、日期……等，並可由使用者自訂資料型態。

2. 支援 Namespace。

3. 使用 XML 語法規則，因此文件作者不必另學一套特殊規格。

4. 設計支援 XML 綱要的軟體，遠比 DTD 容易。

5. 具有強大的擴展能力。

### (三) 文件整合、傳遞與操控

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 在文件整合方面，扮演了一個通用集成器(universal hub)的角色(註三三)，而 XML 的名稱領域、RDF、XLink 等正是文件整合不可或缺的重要技術。

#### 1. XML 名稱領域

不論同質性或異質性的文件資料庫，都可透過 XML 定義不同的詞彙集(vocabulary)加以整合，然而當分散資源透過不同詞彙集，在同一份 XML 文件中被引用或參照時，若是不同詞彙集使用了相同的元素名稱，此時便產生衝突了。因此，XML 進行資源整合時必須注意元素名稱的唯一性問題。XML 名稱領域(XML Namespaces)便是為解決元素命名衝突所發展出來(註三四)。根據 W3C 的定義：一個 XML Namespace 是使用 URI (Universal Resource Identifier，一致性資源識別碼)識別的名稱集，用於 XML 文件中作為元素型別和屬性名稱。由於 URI 是唯一的，所以 Namespace 可以保證文件所使用的名稱是獨一無二的。

XML Namespace 的宣告是由 xmlns：伴隨著一個自訂的前導字(namespace prefix)和一個 URI 名(namespace name)所構成，亦即：

xmlns：namespace\_prefix = namespace\_name，例如：

```
<x xmlns：edi='http://ecommerce.org/schema'>
<!--edi 為前導字-->
<!--http://ecommerce.org/schema 為 URI-->
<!--x 為元素名稱-->
</x>
```

## 2. RDF：資源描述架構

文件整合為資源整合一環，而資源整合的重大議題之一是：如何使得所有被整合的事物不但是「機器可讀的」(machine-readable)，且更進一步是「機器可理解的」(machine-understandable)。這和人工智慧(artificial、intelligent)領域長久以來致力於提昇機器智能的宗旨不謀而合。人類的思維過程不外乎藉由各種管道獲取知識(knowledge)，然後進行推理(reasoning)，最後做出決策(decision)。由此可見，知識乃是人類智慧中極重要的一環。然而，知識的獲得需要經過資料處理及資訊管理的過程才能粹取，如圖5所示。大量的資料經過組織、整理之後成為有用的資訊；而眾多的資訊經過歸納、演繹之後，才能構成知識；有了足夠的知識才能做出正確的推理與決策，此乃人類思維的基本體系。因此，要讓機器如同人類一樣能「理解」事物，就必須發展能夠具體表達知識的方式。於是，知識表達(knowledge representation)這一門探討如何利用不同模式有效描述人類知識的學問，遂逐漸成為人工智慧一個極重要的研究領域。目前，已被提出的知識表達模式眾多，較重要的有：語意網路(semantic nets)、框架(frames)、生產法則(production rules)、劇本(scripts)、邏輯與集合(logic and sets)、物件向量表示法(object-oriented representation)等(註三五)。



圖5 知識的構成

網路資源若能予以有效整合，將建構出人類有史以來最大的知識庫(knowledge base)，輔助人類進行推理與決策。然而，如今Web資料雜亂、資訊氾濫，若要借助機器之功，完成全自動化的資源整合與知識粹取，以目前之資訊科技而言，實仍有許多困難；若要完全利用人工方式，對龐大資料進行組織與整理，又力有未逮。因此，在撰寫Web文件時，由作者或專業人士著錄metadata，提供足夠機器理解之資訊，實為一可行方案。而此種粹取文

件中 metadata 的行為，實際就是一種知識擷取的過程，所產生的 metadata 即是知識的具體表現！有鑑於此，W3C 乃大力倡導「資源描述架構」，簡稱 RDF，作為處理 metadata 的標準模式，使得 Web 上的應用程式可據以交換「機器可理解的」資訊(註三六)。若是從人工智慧的角度觀之，RDF 實可視為一種知識表達，其資料模型將資源表示成特性(property)與值(value)的關係，相當於「物件—屬性—值」三元表示法(object-attribute-value triples) (註三七)，而用來定義資源與屬性關係的 RDF 綱要(RDF Schema)(註三八)，則類似實體關係圖(entity-relationship diagram)(註三九)。然而，RDF 的宗旨並不在於建立一個專家系統或人工智慧系統，因此，在 RDF 中並未定義任何的推論機制(reasoning mechanism)，而僅藉助知識表達技術來描述 metadata、建構知識庫，使得系統開發者可以很容易的自行在此知識庫上建立所需的推論引擎。

在 RDF 資料模型中，所有被描述的事物一律稱為「資源」，每個資源都擁有一些「特性」(properties)作為描述其外觀、特徵或屬性之用。描述資源與特性的表示式則稱為「敘述」(statement)。一個敘述包含「主詞」(subject)、「述詞」(predicate)和「受詞」(object)三部份，可表示成一個三元式 {predicate, subject, object}，或簡寫成 {p, s, o}。此三元式等同於一個起始於端點 s 經由弧線 p 指向端點 o 的有向圖(directed graph)，如圖 6 所示。其含意為「資源 s 有一個屬性 p，其值是 o」或更簡潔的說法：「s 的 p 是 o」。

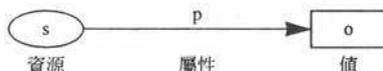


圖 6 以有向圖表現 RDF 敘述

例如，一個描述網站資源的句子：「<http://mail.tku.edu.tw/sclin> 的 creator 是 Sinn-Cheng Lin」，其對應的 RDF 三元式敘述為：

{Creator, [http://mail.tku.edu.tw/sclin], "Sinn-Cheng Lin"}

其資料模型如圖 7 所示。



圖 7 RDF 敘述之簡易資料模型

至於更複雜的敘述，例如「<http://mail.tku.edu.tw/sclin>的creator是一個Person，他的Name是Sinn-Cheng Lin，Email是sclin@mail.tku.edu.tw」，則可表示成如圖8的結構化資料模型。

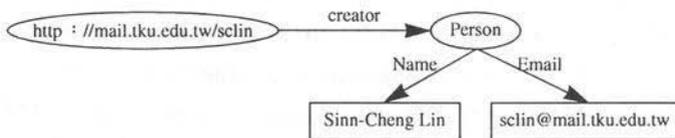


圖8 RDF敘述之結構化資料模型

由於XML文件具有標準化、開放性、結構化及自我描述等特性，因此極適合作為RDF的實作語言。在RDF中，為了使文件讀寫雙方都能瞭解敘述的含意，必須定義彼此能理解的語彙，稱為RDF綱要。然而，不同的RDF綱要極可能定義了同名的詞彙集，以至於一旦同時在同一份文件被引用或參照時，將造成衝突。為了避免此一問題，RDF採用了XML的名稱領域作為解決方案。茲以上述範例說明以XML實現RDF敘述之方式，如圖9所示。

```

<?xml version="1.0"?>
<rdf : RDF
  xmlns : rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns : s="http://description.org/schema/"
  <rdf : Description about="http://mail.tku.edu.tw/sclin">
    <s : Creator>Sinn-Cheng Lin</s : Creator>
  </rdf : Description>
</rdf : RDF>

```

圖9 以XML實現RDF之敘述

在RDF架構中採用都柏林核心集(Dublin Core，簡稱DC)是描述Web文件或網路資源的有效方式之一(註四十)。其用法如圖10所示。

### 3. XLink：XML連結機制

資源整合的另一個重大議題是：如何將資源連結起來。眾所周知，HTML的錨元素(anchor element)和連結元素(link element)是目前Web文件的兩個主要連結機制，已經廣泛使用許多年了。然而，連結的種類繁多，如基本

```

<?xml version="1.0"?>
<rdf : RDF
  xmlns : rdf="http : //www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns : dc="http : //purl.org/metadata/dublin_core#">
  <rdf : Description about="http : //www.dlib.org">
    <dc : Title>D-Lib Program-Research in Digital Libraries</dc : Title>
    <dc : Description>The D-Lib program supports the community of people
      with research interests in digital libraries and electronic publishing.
    </dc : Description>
    <dc : Publisher>Corporation For National Research Initiatives</dc : Publisher>
    <dc : Date>1995-01-07</dc : Date>
    <dc : Subject>
      <rdf : Bag>
        <rdf : li>Research: statistical methods</rdf : li>
        <rdf : li>Education, research, related topics</rdf : li>
        <rdf : li>Library use Studies</rdf : li>
      </rdf : Bag>
    </dc : Subject>
    <dc : Type>World Wide Web Home Page</dc : Type>
    <dc : Format>text/html</dc : Format>
    <dc : Language>en</dc : Language>
  </rdf : Description>
</rdf : RDF>

```

圖 10 RDF 與 Metadata：以 Dublin Core 為例(註四一)

連結、多向連結、索引連結、節點內註釋連結、IS-A 連結、HAS-A 連結、蘊含連結、執行連結、自動連結、類型連結等(註四二)。HTML 的連結只不過是基本連結的一種，還有許多其他類型的連結是無法使用 HTML 完成的。因此 XML 便肩負了增強連結功能的重任，不過在 XML 規格中並未定義任何連結方法來改變 HTML 的現狀，而是以原有 HTML 的連結概念為出發點，延伸出更強大的方式來處理連結資訊。此一延伸機制稱為「XML 連結語言」(XML Linking Language)，簡稱 XLink(註四三)。XLink 目前仍是 W3C 的一個工作草案(working draft)，尚未成為建議標準。為了與 HTML 的簡易連結相容，並提供更強的延伸功能供人們使用，在制訂 XLink 規格時，簡易性、可讀性、延

伸性、實用性等，便成了首要考量的幾個重要設計目標(註四四)。XLink定義了兩種主要的連結：簡易連結(simple links)和延伸連結(extended links)。簡易連結雖然提供類似HTML的錨元素功能，只需要一個定址器(locator)，只能連結到單一資源，只能作單向連結，不過也具備HTML所沒有的能力，例如可以定義連結的語意、展現、動作、行為等。延伸連結可以連結到不定數量的資源，可以作相關連結的過濾，但需要將定址器元素另外置放在與連結元素不同的集合裡。此外，XLink還提供了延伸連結群組(extended link group)，以便建立一個核心文件來集中管理各類連結(註四五)。

#### 4. DOM：文件物件模型

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文件資料內容進行其他加值處理或操控(manipulation)，則可透過W3C制訂的介面標準「文件物件模型」(Document Object Model，簡稱DOM)為之(註四六)。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，電腦程式可輕易的建立文件，可在文件結構中來回穿梭，甚至可新增、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)，可在不同作業系統中，使用任何程式語言加以實現。

由於網路上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)都被視為一個一個的節點(node)，而整份文件則被描述成樹狀結構(tree structure)。例如，圖11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12所示的樹狀結構。圖中的矩形方塊表示文件中的節點，而橢圓形則為該節點的內容。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。

```

<?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 11 以 XML 撰寫之圖書目錄

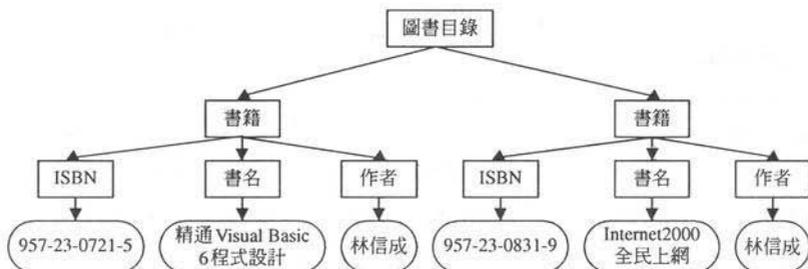


圖 12 以 DOM 展開之樹狀結構

#### (四)文件顯示

前曾述及文件的三要素為資料、結構和表現。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。

一般而言，展示 XML 文件最簡便的方式是透過樣式表(stylesheets)，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 可擴展樣式語言)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。

### 1. CSS：層級樣式表

層級樣式表規格分為level 1(簡稱CSS1)及level 2(簡稱CSS2)兩種，目前都已正式成為W3C的建議標準。CSS1是1996年12月完成，並在1999年1月增修(註四七)；而CSS2則是1998年5月完成(註四八)。目前的瀏覽器都已經有支援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分為外部樣式、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。

在HTML文件中引用外部CSS的語法為：

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

在XML文件中引用CSS的語法為：

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

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。

### 2. XSL：可擴展樣式語言

XSL目前仍是一個正在制訂中的規格，W3C於1998年12月提出了1.0版的工作草案，最新修訂版則在2000年1月公佈(註四九)。雖然最終的正式規格可能與目前的草案有所差異，但其理念是一脈相承的。

XSL是由XML所定義語言，它提供遠超過CSS的強大功能，雖然CSS發展較早，並可作為HTML文件和XML文件的排版樣式，但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。

XSL處理XML文件的過程如圖13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文件轉換為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。



圖 13 以 XSL 顯示 XML 文件(註五十)

在XML文件中引用XSL的語法為：

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

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。表1基於幾項要點，列出兩者的差異(註五一)：

表 1 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

### 3. 以程式剖析XML結構

對於無法使用CSS或XSL完成的工作，可能需要直接以程式剖析XML文件的方式來處理；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現畫面即可。

## 五、結論

XML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。因此，XML將促使網路從資訊處理階段跨越到知識管理階段。

從電子出版的角度來看，目前Web上為數眾多的網站及互相糾結的HTML文件，在寫作之初由於缺乏有效的資料著錄，造成資料雜亂、不易檢索的問題，因此亟需資訊檢索技術與方法來解決；此外，除了致力於技術的發展，在寫作及出版過程中加強metadata的著錄，亦不失為另一個解決網路上電子文件亂象並提昇檢索精確率的有效方案。XML與生俱來的結構化及自我描述特性，能有效的組織資料、描述資料，以進行語意層次的有意義檢索；而其無限擴展的能力，更使其跳脫既定的框架，而在版面編排、多媒體出版、文件共享……等方面發展出獨特的應用，在在使XML成為網路出版內、外兼顧的有效方案。

從資訊傳播的角度而言，傳播的目的在於建立收送彼此雙方的共同性(commonness)，亦即設法共享資訊、觀念、態度或知識(註五二)。然而，傳播的對象可能是人，也可能是電腦，當網路上兩部電腦透過通訊協定(protocol)進行通訊時，只不過具備了訊息傳遞的能力；至於訊息的含意以及如何把這些訊息轉換成另一種形式呈現給人們，通訊協定是無能為力的，但這正是XML的專長。在未來幾年，為因應資訊需求從大眾化到個人化的趨勢，網路上預料將出現更智慧型的搜索引擎，能根據訊息意涵和使用者需求，提供資訊過濾、資訊擷取、智慧型代理等功能。XML技術正是實現此一目標的基石。

從電子商務的角度觀之，不同商業系統採用相同的XML詞彙集，便能隨時自動更新來自各供應商的商品目錄和價格資訊，並根據商品類別為客戶提供最直觀的購物環境。商家之間更可透過金融機構共通的資料交換格式(如以XML制訂的OFX)來交換金融資訊，例如電子支付、財務分析、諮詢文件等；圖書館更可利XML達到自動化的館際合作、館際互借、線上採購等工作。

有鑑於XML將為Web帶來第二波革命(第一波為HTML)，目前，產、官、學界無不積極合作，致力於XML相關技術之研究及工具軟體、應用程式之開發(註五三)。我們何其有幸生活在這個科技發達的時代，得以享受新科技

所帶來的新產品及新服務，大大提昇我們的生活品質。當然，科技是多變的，未來更難以準確預測，我們也可能成為新科技潮流中被實驗的白老鼠。不過，從 XML 受產、官、學各界關注的程度而言，XML 成為下一代 Web 的整合技術殆無疑義，並將在電子出版、電子商務、電子圖書館、電子資料交換等領域展現其強大的應用潛能，究竟能發揮多少效能，只侷限於人們的想像力及創造力。相較於國外 XML 技術正如火如荼的發展，國內的研究仍處於萌芽階段，不但中文化的開發工具和應用程式嚴重不足，連參考文獻、研究報告也都寥寥可數，這對提升國家競爭力無疑是一大阻礙。因此，國內亦應儘速結合各界力量，投入 XML 相關技術之研發，積極從事推動學術、教育、企業各領域早日採用 XML 標準，以因應未來多變的網路應用環境。

## 附 註

註一 二極體(diode)是最基本的半導體元件，當外加的電壓超過門檻值時，內部流通的電子數量會產生如滾雪球般的累增效果，電流便會呈現指數上升，積體電路的主要元件「電晶體」，就是利用這種電壓—電流特性組合成類比或數位電路。

註二 Gordon Moore 是英代爾(Intel)公司的創辦人，曾於 1965-1975 年間，經由長期觀察電腦晶片功能價格比的變動而預言電腦晶片的容量每一至兩年(另一說法為平均 18 個月)將增加一倍。這個預測後來被工程界稱為「莫爾定律」。

註三 Network Wizards 利用搜尋領域名稱系統(Domain Name System)的方式，以網路程式自動探詢並統計 Internet 上所裝設的主機數量，統計結果每半年會公佈一次，歷年來詳細的統計資料可在 <http://www.isc.org/ds/> 找到。

註四 超文件是一種依人腦的聯想思維方式，非線性地儲存、管理和瀏覽資訊的技術，文件之間彼此以超連結方式組成文件網路，便於人們「聯想」搜尋，比起傳統文字的線性模式更符合人類的思維方式和習慣。

註五 “HyperText Markup Language”, available at <<http://www.w3.org/pub/MarkUp/>>

註六 邱炯友、林信成，「各國政府出版品電子化策略及措施之研究」，研考會研究報告(民國 88 年 6 月)。

註七 林盈達，多媒體網路：趨勢、技術、應用(台北：松崗，民國 86 年)，頁 1-1。

註八 Salton Gerard, “A Comparison Between Manual and Automatic Indexing Methods,” *American Documentation*, 20: 1(1969): 61-71.

註九 K. A. Hamil & A. Zamora, “The Use of Titles for Automatic Document Classification,” *Journal of American Society for Information Science*, 43: 2(1992): 130-148.

註十 Salton, Gerard, *An Introduction to Modern Information Retrieval* (New York : McGraw-Hill, 1983).

註十一 Salton Gerard & C. Buckley, "Improving Retrieval Performance by Relevance Feedback," *Journal of the American Society for Information Science*, 41(1990) : 288-297.

註十二 陳光華, 〈超越資訊檢索的語言藩籬〉, *大學圖書館*, 2 : 1 (民國87年1月)。

註十三 曾元顯, 〈多媒體資訊檢索技術之探討〉, 21世紀資訊科學與技術的展望國際學術研討會(民國85年9月), 得自<<http://blue.lines.fju.edu.tw/~tseng/papers/mir.htm>>.

註十四 Michael Lesk, "The Seven Ages of Information Retrieval," Bellcore, available at <<http://www.lesk.com/mlesk/ages/ages.html>>.

註十五 Robert R. Korfhage, *Information Storage and Retrieval* (New York : Wiley Computer Publishing, 1997), pp.196-199。

註十六 Metadata是「用來描述資料的資料」(data describes other data)或「關於資料的資料」(data about data)。其譯名有「元資料」、「描述資料」、「詮釋資料」、「後設資料」……等,並不統一,因此本文直接採用原文而不使用譯文。

註十七 陳雪華, 〈網路資源與Metadata之發展〉, *圖書館學刊*, 12期(民國86年) : 23-28。

註十八 「XML工作小組」最初稱為「SGML編審委員會」(SGML Editorial Review Board)。

註十九 Jon Bosak & Tim Bray, "XML and the Second-Generation Web," (*Scientific American*, May 1999), available at <<http://www.sciam.com/1999/0599issue/0599bosak.html>>.

註二十 "Extensible Markup Language (XML) Activity," available at <<http://www.w3.org/XML/Activity.html>>.

註二一 同註二十。

註二二 "Mathematical Markup Language (MathML)," available at <<http://www.w3.org/Math/>>.

註二三 "W3C Scalable Vector Graphics (SVG)," available at <<http://www.w3.org/Graphics/SVG/>>.

註二四 "Synchronized Multimedia," available at <<http://www.w3.org/AudioVideo/#SMIL>>.

註二五 "Resource Description Framework (RDF)," available at <<http://www.w3.org/RDF/>>.

註二六 "Channel Definition Format (CDF)," available at <<http://www.w3.org/TR/NOTE-CDFsubmit.html>>.

註二七 同註二十。

註二八 XML 1.0規範中對於完構性(well-formedness)和有效性(validity)有嚴謹

的定義，可由以下 URL 取得詳細說明 <<http://www.w3.org/TR/1998/REC-xml-19980210>>。

註二九 曾守正編著，資料庫系統之理論與實務(台北：儒林，民國87年)。

註三十 “XML Schema Requirements,” available at <<http://www.w3.org/TR/NOTE-xml-schema-req>>。

註三一 「XML 綱要」共有兩部份，第一部份為“XML Schema Part 1 : Structures”，可在以下 URL 取得 <<http://www.w3.org/TR/xmlschema-1>>；第二部份為“XML Schema Part 2 : Datatypes”，可在以下 URL 取得 <<http://www.w3.org/TR/xmlschema-2>>。

註三二 Robin Cover, “SGML : General Introductions and Overviews,” in *The SGML/XML Web Page* (28 June 1999), available at <<http://www.oasis-open.org/cover/general.html#overview>>。

註三三 Jon Bosak, “XML, Java, and the future of the Web,” (10 March 1997), available at <<http://metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.html>>。

註三四 “Namespace in XML,” available at <<http://www.w3.org/TR/REC-xml-names>>。

註三五 Giarratano and Riley, *Expert Systems : Principle and Programming* (Boston : PWS-KENT, 1989), pp.63-102。

註三六 W3C Recommendation, “Resource Description Framework (RDF) Model and Syntax Specification,” (22 February 1999), available at <<http://www.w3.org/TR/REC-rdf-syntax>>。

註三七 同註三五，頁73。

註三八 W3C Recommendation, “Resource Description Framework (RDF) Schema Specification,” 3 March 1999, available at <<http://www.w3.org/TR/REC-rdf-schema>>。

註三九 姚步愼，人工智慧與專家系統導論(台北：基峰，民國80年)，頁3-11~3-13。

註四十 都柏林核心集創始於1995年3月，當時OCLC(Online Computer Library Center)和NCSA(National Center for Supercomputing Application)爲了制訂一套網路資源的描述格式，乃聯合舉辦了一個研討會，邀集來自電腦、網路和圖書館領域的52位學者專家，共同針對Metadata的中心議題，希望制訂一個簡易有彈性並且連非圖書館專業人員也能輕易使用的Metadata格式，以便有效描述網路上多樣化電子文件。基於以上共識，與會學者們主張先建立一套最小規模的核心資料項，同時擁有意義明確和擴充彈性等特色，於是在1995年第一次研討會中訂定了13項，而在1996年第三次研討會時，修訂爲15項。

註四一 同註三六。

註四二 吳樂南，多媒體系統：軟硬體、原理及應用(台北：儒林，民國85年)，頁11-16~11-20。

註四三 “XML Linking Language (XLink),” available at <<http://www.w3.org/TR>>。



/xlink>, W3C Working Draft, Dec.20, 1999.

註四四 William J.Pardi 著, 鄒正平編譯, 實戰XML (台北: 微軟, 民國88年7月), 頁180~182。

註四五 同註四四, 鄒正平, 頁196~198。

註四六 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," (1 October 1998), available at <http://www.w3.org/TR/REC-DOM-Level-1>。

註四七 Hakon W. Lie & Bert Bos, "Cascading Style Sheets (CSS1) Level 1 Specification," (17 December 1996, revised 11 January 1999), available at <http://www.w3.org/TR/REC-CSS1>。

註四八 Bert Bos, Hakon Wium Lie, Chris Lilley and Ian Jacobs, "Cascading Style Sheets, Level 2 (CSS2) Specification," (12 May 1998), available at <http://www.w3.org/TR/REC-CSS2>。

註四九 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 January 2000), available at <http://www.w3.org/TR/xsl>。

註五十 Microsoft Web Workshop, "XML Architecture," *MSDN Online*, available at <http://msdn.microsoft.com/xml/general/architecture>。

註五一 Jakob Nielsen, "Web Style Sheets," available at <http://www.w3.org/Style/>

註五二 傳播(communication)一詞源自拉丁文Communi, 意思是「共同」, 欲了解傳播的正式定義, 可參考傳播學者Frank Dance的論文"The Concepts of communication," *Journal of Communication*, 20 (1970): 201-210。其中整理了各家學說, 每種皆以短語標明出不同的主旨。本文對於傳播一詞, 則引用徐佳士教授在*大眾傳播理論*(正中書局, 民國76年)一書所說明的概念。

註五三 在James Tauber和Linda van den Brink所維護的網站「XMLSOFTWARE」中(<http://www.xmlsoftware.com/>), 羅列了數以百計由各家廠商所開發的XML軟體, 依功能區分為XML Browsers、Conversion Tools、Database Systems、Document/Content Management Systems、DTD Editors/Tools、XML Editors、XML Parsers/Processors、Publishing Systems、Utilities/Tools/APIs、XLink/XPointer Tools、XSLT editors、XSL formatters、XSLT engines、XSLT utilities、XPath utilities、Search engines等十餘大類, 並可連結至原開發廠商獲取更詳細資料。