

XML 在電子出版之應用 — XHTML、SMIL、MathML 與 SVG 初探

Applications of XML for Electronic Publication: An Introduction to XHTML, SMIL, MathML and SVG

林信成

Sinn-Cheng Lin

淡江大學資訊與圖書館學系

摘要 Abstract

「內容」、「結構」和「展現」為電子文件三要素，早期的 HTML 將此三要素混合處理，為造成如今網路文件雜亂無章的主因。而 XML 這個將主導新一代 Web 發展的核心技術，則採取了分離處理的原則，不但能夠有效的組織資訊、表達知識，更能透過各種方式如版面編排或樣式表，將電子文件以多樣化的風貌展現出來。吾人從 XML 於電子出版之應用的角度觀察，已發表若干與電子文件「結構」有關之論述，本文則著重在電子文件的「展現」層面，並根據 W3C 已公告列入推薦標準或候選推薦標準的幾種 XML 應用語言，如 XHTML、SMIL、MathML 和 SVG，說明其在電子文件展示方面的特色及整合方式。

The content, structure and presentation are three essential components of the electronic documents. The HTML dealt with them confusingly. It has caused the Web to be infested with the chaotic documents. However, the XML, which will be the kernel technology of the second-generation Web, dealt them with the principle of separation. By doing so, XML can organize the information and represent the knowledge in a more structural manner. Furthermore, it can present the documents in multiple layouts with the pre-formatted templates or style-sheets. After discussing the issue of "structure" for electronic publication in our previous work, this article is focused on the issue of "presentation" for document presentations. W3C had proposed some XML applications as recommendations or candidate recommendations for these purposes. For example, XHTML, SIML, MathML and SVG. The paper will give an overview about these applications and examine their abilities of integration in electronic document presentation.

關鍵詞 (Keywords): 可擴展標注語言 (Extensible Markup Language) 可擴展超

文件標注語言 (Extensible Hypertext Markup Language) 同步多媒體整合語言 (Synchronized Multimedia Integration Language) 數學標注語言 (Mathematical Markup Language) 可調整向量圖 (Scalable Vector Graphics)

一、前言

XML 和 HTML 都是從 SGML (Standard Generalized Markup Language , 標準通用標注語言) 演變而來的。SGML 發表於 1986 年 (ISO 8879) , 主要目的在於規範文件製作及交換的標準, 以確保電子文件能在不同的電腦系統上進行交換與應用, 它藉由 DTD (Document Type Definition , 文件型別定義) 定義標籤及規範文件結構, 作為標注內容的依據, 因此 SGML 具有可移植、可擴展、結構化 ... 等特性, 樹立了文件共享的基礎。但由於 SGML 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 Web 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, 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, 直到如今, Web 上的大多數文件仍是以 HTML 作為主要的寫作格式。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。其中最嚴重的便是 HTML 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。為了彌補 HTML 的先天缺陷, 近年來發展出許多改善方案, 例如藉助動態 HTML (Dynamic HTML) Script 語言、CGI、ASP 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, 使得近幾年 Web 大放異彩。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係的結構、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。其實, 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。

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。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¹, 當年 11 月提交 XML 初稿, 並於 1998 年 1 月 10

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

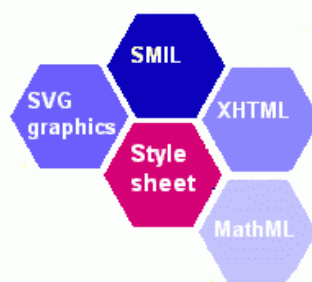
日正式通過 XML 1.0 規範，成為 W3C 的一個建議標準（Recommendation）。由於 XML 具有可擴展性、結構性、自我描述性，並採用資料和樣式分離原則，使其在資料的管理、交換上擁有極為卓越之性能。雖然 XML 和 HTML 一樣都是從 SGML 演變而來的，不過 HTML 是 SGML 的一個應用語言（Application），而 XML 卻是 SGML 的一個精簡子集（Subset）。XML 將 SGML 去蕪存菁，捨棄約百分之二十複雜罕用的部份，承襲了其他百分之八十的特點，是以具備了 SGML 所沒有的簡易性與靈活性，又有著 HTML 所欠缺的擴展性與結構性。因此，稱 XML 為主導「第二代 Web」(Second-Generation Web) 的重要技術實不為過²。一般預料，XML 的發展將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。

本文中我們將關注的焦點集中於電子出版方面，介紹幾個基於 XML 所發展出來的新一代電子文件展示技術，如 XHTML、SMIL、MathML、SVG 等，以期讓讀者對於 XML 之應用潛能有個概括的瞭解。

二、XML 與新一代電子文件展示技術

XML 採用內、外分離原則，使得文件的內容、結構與展示可分開處理，它使用 DTD、RDF 等方式來規範文件的結構；至於文件的展示，則除了典型的 CSS 及 XSL 等方式外，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 1 所示。以下針對 W3C 所提出的 XHTML、SMIL、MathML 和 SVG 等作為電子文件展示用途的規範加以說明。

Using XHTML with other W3C tag sets



- XHTML for structural mark-up of documents
- SMIL for Multimedia
- MathML for Math
- SVG for Scalable Vector Graphics

² 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>>.

圖 1 XML 使用者介面語言之整合³

三、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⁴，該標準修正了 1998 年 4 月發佈的 HTML 4.0 規範中的某些 Bug⁵。

在 XML 規範正式成為 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 語法從新定義，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)⁶。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。

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為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)、伺服器端影像映射模組 (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⁷。

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

³ "HyperText Markup Language -- Activity Statement", available at <<http://www.w3.org/Markup/Activity>>

⁴ "HTML 4.01 Specification," W3C Recommendation, 24 December 1999, available at <<http://www.w3.org/TR/html4/>>.

⁵ 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 <<http://www.w3.org/TR/1998/REC-html40-19980424/>>, "HTML 4.0 Specification," W3C Recommendation, revised on 24 April 1998.

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

⁷ "Modularization of XHTML", W3C Working Draft 5 January 2000, available at <<http://www.w3.org/TR/xhtml-modularization/>>

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; 而文件描述檔則紀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 文件和樣式表為何, 當然, 這其中仍須某些轉換程式的輔助, 其運作過程如圖 2 所示。

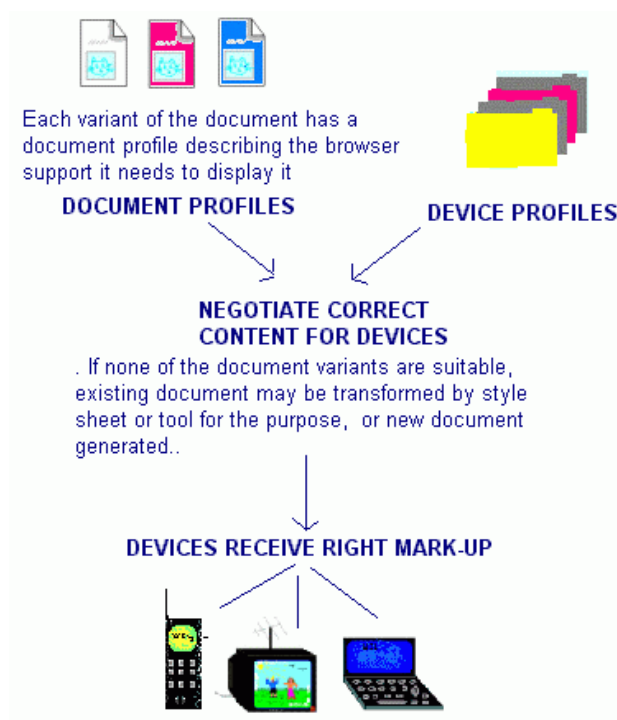


圖 2 文件在不同的設備上呈現出不同的版面⁸

四、同步多媒體整合語言 SMIL

雖然 Web 已經成為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。為了提升 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為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 月正式成為 W3C 的建議標準⁹, 而新版的 SMIL 2.0 目前則仍處於工作草案 (Working Draft) 的階段¹⁰。

⁸ 同註3。

⁹ "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", W3C Recommendation 15-June-1998, available at <<http://www.w3.org/TR/REC-smil/>>

¹⁰ "Synchronized Multimedia Integration Language (SMIL 2.0) Specification", W3C Working Draft 21 September 2000, available at <<http://www.w3.org/TR/smil20/>>.

SMIL (唸作 smile) 語言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 文件同樣可以力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為，精確規劃螢幕上的布局，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 的主要特點為：

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 雖然被標榜為一個多媒體環境，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、圖片轉換呈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。

五、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 月提出專為出版數學公式的標注語言 MathML (Mathematical Markup Language)，1.01 版建議標準 (Recommendation) 於 1999 年 7 月修訂完成¹¹，目前 MathML 2.0 版已經進入候選建議 (Candidate Recommendation) 階段¹²。MathML 的標籤群可概分為展示標籤 (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，作為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。

目前市面上的瀏覽器大都未支援 MathML，不過 W3C 有提供了一個具備編輯及瀏覽 MathML 能力的瀏覽器，稱為 Amaya¹³，可在 W3C 的網站上自由下載使用¹⁴，執行結果如圖 3 所示。

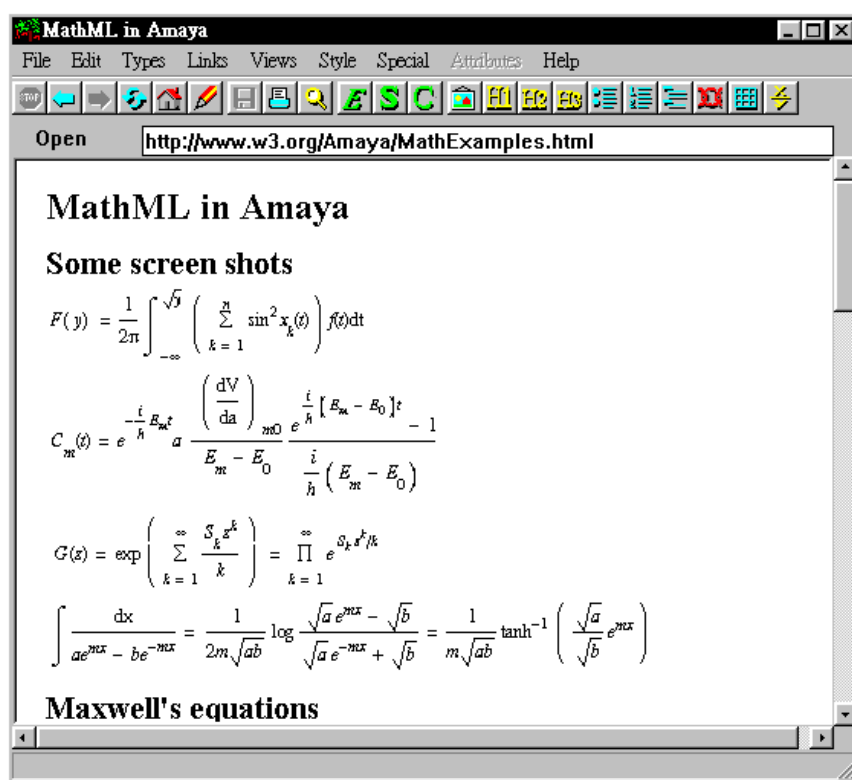


圖 3 Amaya 執行畫面

MathML 也是 XML 的一個應用，因此具備高度結構化的特性，有助於機器

¹¹ "Mathematical Markup Language (MathML™) 1.01 Specification", W3C Recommendation, revision of 7 July 1999, available at <<http://www.w3.org/TR/REC-MathML/>>.

¹² "Mathematical Markup Language (MathML) 2.0 Specification", W3C Candidate Recommendation, 13 November 2000, available at <<http://www.w3.org/TR/MathML2/>>.

¹³ Vincent Quint and Irene Vatton, "An Introduction to Amaya", W3C NOTE, 20 February 1997, available at <<http://www.w3.org/TR/NOTE-amaya-970220.html>>.

¹⁴ "Welcome to Amaya -- W3C's Editor/Browser", available at <<http://www.w3.org/Amaya/>>.

理解與自動處理，但一般科技文獻的作者若要以 MathML 直接編寫數學方程式卻不是件容易的事，現在就以電磁學中常見的馬克士威爾方程式（Maxwell's equations）為例來說明：

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0}; \nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}; c^2 \nabla \times \mathbf{B} = \frac{\partial \mathbf{E}}{\partial t} + \frac{\mathbf{j}}{\epsilon_0}; \nabla \cdot \mathbf{B} = 0$$

這一組馬克士威爾方程式若以 MathML 編寫，其原始碼則如下所示¹⁵：

```
<?xml version="1.0"?>
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <mrow>
      <mi fontstyle="normal">&dtri;</mi>
      <mo>&CenterDot;</mo>
      <mi fontstyle="normal">E</mi>
    </mrow>
    <mo>=</mo>
    <mfrac>
      <mi>&rho;</mi>
      <msub>
        <mi>&epsiv;</mi>
        <mn>0</mn>
      </msub>
    </mfrac>
  </mrow>
  <mtext>&thickspace;;&thickspace;</mtext>
  <mrow>
    <mrow>
      <mi fontstyle="normal">&dtri;</mi>
      <mo>&times;</mo>
      <mi fontstyle="normal">E</mi>
    </mrow>
    <mo>=</mo>
    <mrow>
      <mo>-</mo>
      <mfrac>
        <mrow>
          <mi>&PartialD;</mi>
```

¹⁵ "MathML in Amaya -- Some screen shots", available at <http://www.w3.org/Amaya/MathExamples.html>.


```

        <mo>&ApplyFunction;</mo>
        <mi fontstyle="normal">B</mi>
    </mrow>
    <mrow>
        <mi>&PartialD;</mi>
        <mo>&ApplyFunction;</mo>
        <mi>t</mi>
    </mrow>
</mfrac>
</mrow>
</mrow>
<mtext>&thickspace;;&thickspace;</mtext>
<mrow>
    <mrow>
        <msup>
            <mi>c</mi>
            <mn>2</mn>
        </msup>
        <mo>&InvisibleTimes;</mo>
        <mrow>
            <mi fontstyle="normal">&dtri;</mi>
            <mo>&times;</mo>
            <mi fontstyle="normal">B</mi>
        </mrow>
    </mrow>
</mrow>
<mo>=</mo>
<mrow>
    <mfrac>
        <mrow>
            <mi>&PartialD;</mi>
            <mo>&ApplyFunction;</mo>
            <mi fontstyle="normal">E</mi>
        </mrow>
        <mrow>
            <mi>&PartialD;</mi>
            <mo>&ApplyFunction;</mo>
            <mi>t</mi>
        </mrow>
    </mfrac>

```

```

</mfrac>
<mo>+</mo>
<mfrac>
  <mi>j</mi>
  <msub>
    <mi>&epsiv;</mi>
    <mn>0</mn>
  </msub>
</mfrac>
</mrow>
</mrow>
<mtext>&thickspace;;&thickspace;</mtext>
<mrow>
  <mrow>
    <mi fontstyle="normal">&dtri;</mi>
    <mo>&CenterDot;</mo>
    <mi fontstyle="normal">B</mi>
  </mrow>
  <mo>=</mo>
  <mn>0</mn>
</mrow>
</math>

```

顯然，要產生結構化文件是需要付出代價的，一個簡單的方程式必須大費周章，才能成為符合有效性的文件（Validated Document）！幸好 Amaya 不但具備如同一般編輯器的所見即所得（WYSIWYG）的使用者介面，可將畫面上所編輯的公式直接轉換成 MathML，更提供樹狀結構檢視畫面，如圖 4 所示，使得編輯數學公式時更能輔以結構編排，有助於結構化文件之產出。

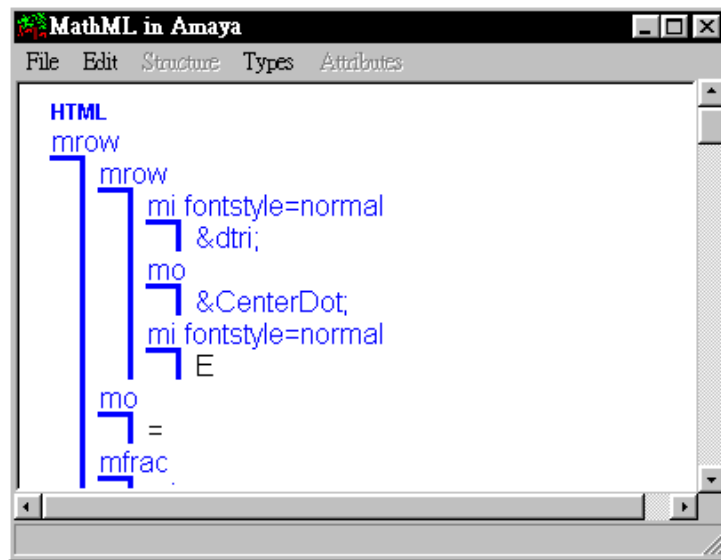


圖 4 Amaya 之樹狀結構檢視畫面

六、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。為了能增強 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，而最新的候選建議 (Candidate Recommendation) 則於 2000 年 11 月公布¹⁶。SVG 也是 XML 的一個應用，主要是作為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。SVG 不但提供靜態的向量圖形，更可以製作向量式動畫，雖然 Macromedia 公司強力主導的 Flash 技術也有類似的功能，但是 SVG 符合 XML 規範，未來之整合性及相容性上應高於 Flash。

目前已有許多應用軟體開始支援 SVG，例如，繪圖軟體市場上的兩大巨頭 Adobe 公司和 Corel 公司，不約而同的在 2000 年 8 月宣稱其新版的繪圖軟體 Adobe Illustrator 9 和 Corel DRAW 10 已經開始支援 SVG^{17, 18}。使用者若想順利瀏覽 SVG 網頁，則需要在瀏覽器上安裝 SVG 檢視器 (SVG Viewer) 或 SVG 播放器 (SVG

¹⁶ "Scalable Vector Graphics (SVG) 1.0 Specification", W3C Candidate Recommendation 02 November 2000, available at <<http://www.w3.org/TR/SVG/>>.

¹⁷ "Corel Launches CorelDRAW 10 Graphics Suite at Seybold!", Ottawa, Canada - August 29, 2000, available at <http://www.corel.com/news/2000/august/august_29_2000.htm>.

¹⁸ "Adobe Illustrator 9 New Features -- Flash 和 SVG output", available at <<http://www.adobe.com/products/illustrator/keyfeature1.html>>.

Player)，如 Adobe SVG Viewer¹⁹。由於 SVG 也是一個 XML 應用語言，因此具有結構化、可擴展、自我描述等特性，如圖 5 是一個簡易的 SVG 原始碼，能產生一個藍色矩形，而由於筆者的瀏覽器有安裝 Adobe SVG Viewer，所以可以正常顯示此段 SVG 原始碼所呈現出來的結果；對於未安裝 SVG 檢視器或 SVG 播放器的瀏覽器，則無法正常顯示藍色矩形。

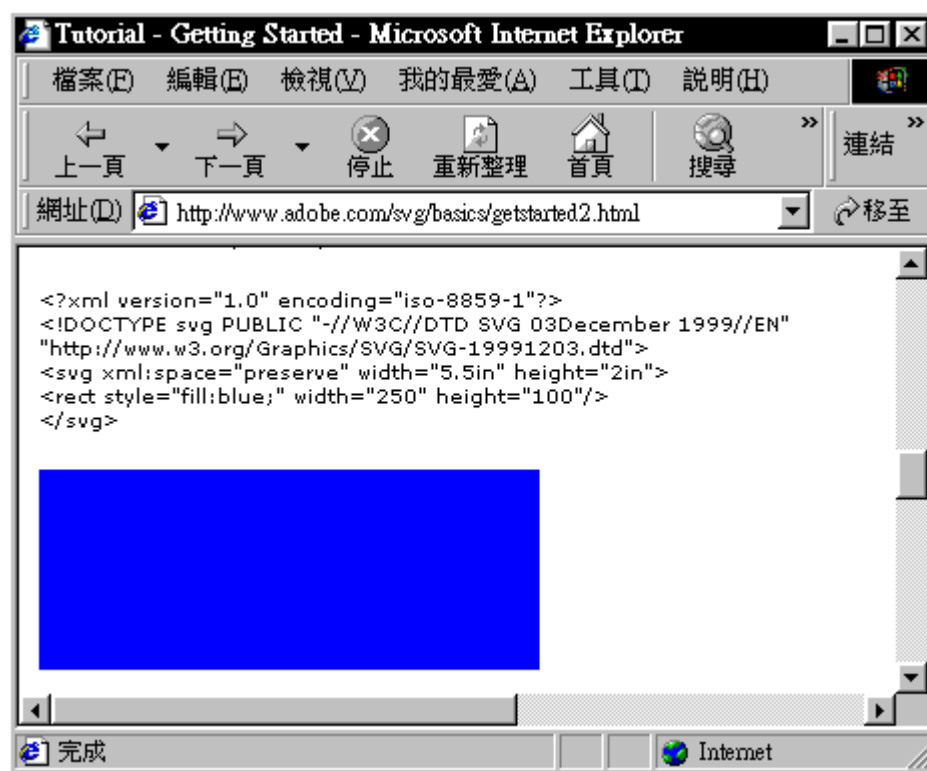


圖 5 簡易 SVG 原始碼及檢視結果

七、結論與建議

從電子出版的角度來看，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。有鑑於此，目前各界無不積極合作，致力於 XML 相關技術之研究及工具軟體、應用程式之開發²⁰。不過，這些應用程式大都各自獨立，尚未能有效整合，亦缺乏支援中文系統之能力，使得以 XML 為核心之電子出版無法有效的推展。因此，有必要發展一套適合於電子文件的作者、管理者或出版者皆能輕鬆使用的「整合性 XML 文

¹⁹ "Adobe SVG Viewer", available at <<http://www.adobe.com/svg/viewer/install/>>.

²⁰ 在 James Tauber 和 Linda van den Brink 所維護的網站「XMLSOFTWARE」中 (URL: <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。

件管理系統」(Integrated XML Document Management System)。此系統不但要具備中文化介面、簡單易用等特性,而且對於不具備任何 XML 技術素養的使用者,只要經由一般性的編輯流程,即可產出符合 XML 規範的文件。如此一來,不僅可以解決檢索、管理等問題,更使得電子文件在出版的同時,也具備了結構化資料庫的特性。

為了使此一系統能基於 XML 規範,對電子文件的結構、內容、展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,進而建構一個適用於網路的電子文件整合出版環境,吾人藉由系統分析過程,建議此一整合系統可劃分為三個子系統,分別是:

(1) 編輯子系統:含文件結構編輯模組、文件內容編輯模組和文件樣式編輯模組。其中,文件結構編輯模組的主要任務是提供一個定義文件結構的途徑,最好具備圖形化的使用者介面(Graphic User Interface),並將使用者在畫面上所定義的結構轉換成 DTD 或 XML 綱要,作為驗證 XML 文件完構性及有效性的依據;而文件內容編輯模組的任務則是提供內容編輯的必要工具,如新增、刪除、複製、剪下、貼上 ... 等,若是文件中尚包含多媒體資料,則還須具備支援編輯多媒體資料的能力;至於文件樣式編輯模組的主要任務則在於提供版面和樣式的編排,如字型大小、段落間距、排列方式、字體樣式 ... ,及有關該文件外觀的設定等。然而,由於 XML 文件不似 HTML 有內定樣式,必須透過 CSS 方式或 XSL 來顯示,當然亦可藉由 XSLT 轉化成 XHTML、SMIL、MathML、SVG 或其他各種專為展示用途而發展的應用語言來展現。

(2) 核心子系統:含剖析模組、轉換模組和檢索模組。其中,剖析模組的任務是依據 DTD 或 XML Schema 的定義,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;而轉換模組主要任務是將文件的詮釋資料,依據所定義的結構轉成資料庫方式儲存,其優點是可藉助現有資料庫管理系統(DBMS)強大的功能,對文件進行查詢、存取、管理、傳遞等處理;至於檢索模組是提供一個文件檢索的機制。由於轉換模組已經將文件轉存成資料庫格式,因此只要藉由通用的資料庫檢索方式,如結構化查詢語言(Structure Query Language, SQL),即可很容易的檢索文件內容。

(3) 出版子系統:含發行模組和排版模組。其中,發行模組的主要目的是將文件以 Web 方式送到伺服器上;而既然要以 Web 方式發行,排版模組的任務便是依據使用者在編輯子系統中,透過文件樣式編輯模組所制訂的格式來展現文件內容,展現的方式則可以排版樣本或標準 XHTML 為主,再視資訊的內容特色,搭配 SMIL、MathML、SVG 或其他應用語言為之。

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。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。相較於國外 XML 技術正如火如荼的發展,國? 的研究仍處於萌芽階段,應儘速結合各界力量,投入

相關技術之研發，以因應未來多變的網路環境，提升國家之競爭力。