

# Valid XML Editor -- 不懂 XML 也能編輯 XML 資料

施乃誠, 曾俊雄, 許杏如, 郭譽申

中央研究院資訊所

{ncshi | lendle | minitaco | yskuo}@iis.sinica.edu.tw

## 摘要

隨著資訊科技的演進，資料交換的需求大增，促成 XML 標準的興起，市面上雖已提供不少 XML 編輯軟體，但對各領域而言，介面無法客製化及使用者須具備 XML 基礎知識是一大障礙，本計劃將以一般使用者為目標群眾，透過演算法降低 XML 入門門檻，並以可客製化使用者介面為特點，希望在計劃完成後，能讓不懂 XML 的使用者也能輕鬆編輯 XML 資料，則各領域將能順利推廣 XML 標準，促進產能進步。

關鍵字：Valid XML Editor、XML 編輯器、Valid

## 1. 簡介

XML (Extensible Markup Language) [9]的跨平台、跨網路、跨程式的特性，讓它成為通行各領域的標準語彙，並組織公會來訂定與推動，例如：BizTalk RosettaNet XML/EDI ebXML、HL7 等國際產業流程標準，隨著各種 XML 語彙、標準的訂定與產生，益發需要適合編輯 XML 文件的軟體。

不論是 XML 資料或文件，皆需遵守以下兩項基本規則，Well-Formed 及 Valid：

Well-Formed，意即格式良好，元件必須合乎 XML 通用語法規則，可能是元件須有與起始標籤相對應的結尾標籤，例如：

`<author></author>`，也可能是一種空白元件標籤，例如：`<img/>`，元件與元件間不能錯置。

Valid，意即「確證」；一份 XML 文件只要 Well-formed，瀏覽器就能讀取，但 XML 文件是否「確證」，僅有 Well-Formed 是不夠的，還要看它的標籤用法是否符合某個 Schema/DTD [10][11]，換言之，一份 Valid 的 XML 文件一定是 Well-Formed 且符合語法。

現今已有的 XML 編輯軟體有以下幾種特點：確認 XML 文件 Well-Formed 或 Valid 與否、提供語法檢查及校正建議、需要具備基礎的 XML 知識 提供不精確的建議及指示、無法對 XML 資料產生足夠的規範、難以客製化使用者介面。

本計劃將在研討會場提供實機展示以 Forms-XML 元件為核心所研發的 Valid XML Editor，期能適用各種語彙，並可專為特定語彙客製化，希望一般無 XML 知識的使用者亦能依此編輯 Valid XML 文件。

## 2. 相關工作

以下針對 XML 編輯器的「編輯模型」與「使用者介面」兩方面加以分析，藉以瞭解現今編輯器的特點，作為本計劃研發 XML Editor 的參考，以便發揮 Valid XML Editor 的最大功效。

## 2.1 由編輯模型來看

現今的 XML 編輯器多會提供基本的確認機制，讓使用者確認編輯出來的 XML 文件是否 Well-formed 或 Valid，在編輯過程中，部份程式提供編輯建議及指示給不熟悉 XML 的使用者，但卻不是很精確的幫助，導致編輯器須另外提供語法檢查或校正建議，但在缺乏 XML 知識的情況下，即使提供校正建議，使用者往往無法得知該如何依照建議將文件編修至正確狀態，這表示使用者需要具備基礎的 XML 知識，才能使用現有的 XML 編輯器。

另外，由於現有的編輯器不是專為某種 XML 語彙所設計，故無法對 XML 資料內容做足夠的規範，允許使用者過度靈活的運用，太多的功能容易令使用者困惑；也因為如此，若產業要使用現有 XML 編輯器，也難以達到客製化使用者介面的要求。

## 2.2 就使用者介面而言

從 UI 的表現來看，目前的 XML 編輯器大約分為三大類：Text View、Tree View、Presentation View。

### Text View：

此類軟體介面將 XML 結構與內容以純文字的形式表現出來，適合熟悉 XML 的使用者。代表軟體包括：XRay XML Editor [2]。

### Tree View：

此類軟體介面將 XML 的層級觀念以樹狀結構表示，能清楚的展現 XML 的元件關係，適合具備 XML 知識的使用者，代表軟體包括：XMLSPY [1]。

### Presentation View：

此類軟體介面能轉換 XML 結構，將內容顯現出來，較適合 XML 初學者，大略還可分

成以下兩種：

### 表單式：

此類軟體介面能轉換 XML 結構，將 XML 內容以表單形式表現出來。代表軟體包括：Office InfoPath [7]。

### Word Processor：

此類軟體介面能轉換 XML 結構，將 XML 內容以文字表達出來，類似 Microsoft Word。代表軟體包括：Corel XMetal [3]。

## 3. Valid XML Editor

XML 既已發展為各界標準的基礎語彙，在訂定標準後，各業人士勢必需要以此標準來編修 XML 文件，由上得知，現今使用 XML Editor 或多或少皆須具備 XML 知識，才能順利編輯一份 Valid 的 XML 文件；Valid XML Editor 以 Forms-XML 元件為核心，針對一般使用者，將使用者難懂的架構轉換為表單，透過演算法計算合法元件進而產生 Well-Formed 且 Valid 的 XML 文件，藉以下幾項特點提出設計：

Valid XML 文件、符合 DTD 語法 (DTD-Compliant) 的 XML 編輯器、產生 HTML 表單的使用者介面、自動產生合法元件(element/attribute)、在鄰近區域內加入元件、可客製化使用者介面、以樣本檔為基礎的編輯行為。

### 3.1 Valid XML 文件

一份 XML 是不是有用，端看它是否符合該領域的語法規範，故 Valid XML 文件為 Valid XML Editor 的製作目標；一般編輯器提供編輯 XML 功能，但 Valid 與否還是要經由使用者自己確認、更改，Valid XML Editor 透過自動產生元件的特性，在編輯過程中或許還

未達到 Valid 的條件，但能保證最後一定是一份 Valid XML 文件，換言之，Valid XML Editor 專注於處理及產生 Valid XML 文件，不 Valid 的 XML 文件不在 Valid XML Editor 的作用範圍，本研發軟體亦因此不需提供 Valid 與否的確認機制。

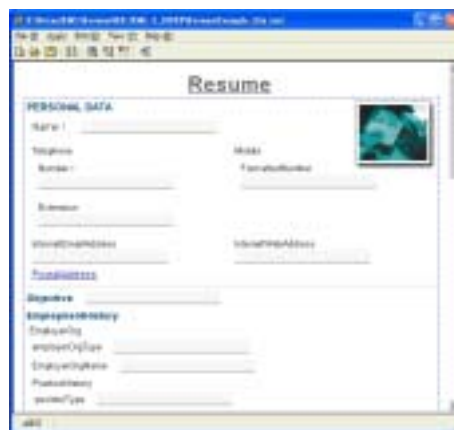
### 3.2 符合 DTD 語法(DTD-Compliant)的 XML 編輯器

Valid XML 文件是由許多符合 DTD 語法(DTD-Compliant) [5]的結構集合而成，當達到 Valid 的必要因素且符合 DTD 語法才是一份 Valid 的 XML 文件；Valid XML Editor 核心元件能由 Glushkov automaton [4]中建立演算法，計算出 XML 文件從開始符合 DTD 語法到結束的完整路徑，從該路徑分析語法規則的所需要素，進而精確建議編輯過程中的可新增元件，因此使用者不須具備 XML 相關語法知識即能產生符合 DTD 的 XML 文件，使用者也不會有語法錯誤的煩惱。

### 3.3 產生 HTML 表單的使用者介面

Valid XML Editor 既然以一般使用者為開發目標，相對於純文字與樹狀結構的表現方式來說，Presentation View 的形式是較為簡單的，為了降低 XML 入門門檻，採用簡易的 HTML 表單作為使用者介面 [6]，畢竟表單為各領域工作者最常使用的格式之一，使用者可依平常填寫一般的表單的習慣生成 XML 文件；進階使用者亦可依 HTML 語法輕易的客製化使用者介面。

如圖一，為 Resume 的 XML 文件，經由 HTML 表單的使用者介面，是否為 XML 格式對使用者而言並無太大的差別，可輕易的透過此介面填寫其內容。



圖一 Resume 的 XML 文件

### 3.4 自動產生合法元件(element/attribute)

Valid XML Editor 可由 Schema/DTD [10][11] 中推論該份 XML 文件若要 Valid，則必須出現哪些元件，進而自動產生 Valid 所需的元件 (圖二)，若 Valid 所需元件不只一個或須多個擇其一，則以下拉選單將集合選項 (圖三)，供使用者選擇，直到 Valid 所需元件皆已新增完成，除此之外，還可依原理推算出元件與元件之間，可合法新增哪些元件，由以上可判斷出：在 Valid XML Editor 中若將必須出現的元件新增完成，則可保持編輯中 XML 文件的 Valid 狀態。



圖二 產生必須出現的元件



圖三 下拉選單集合所須元件



圖五 After 與 Child 無元件可加入

### 3.5 在鄰近區域內加入元件

一般使用者在不瞭解語法結構之前，往往不知從何處新增元件，透過演算法推算出元件與元件間的可新增元件後，Valid XML Editor 集合相關資訊，針對作用元件產生鄰近區域內可新增元件的對話視窗，分「Before」、「After」、「Child」三方面供使用者快速選擇（圖四），省卻尋找元件的時間，並可依演算法迅速計算出作用元件是否無鄰近區域元件可供新增，進而提供相關訊息（圖五）。



圖四 在鄰近區域內加入元件

### 3.6 可客製化使用者介面

使用者介面的產生來自：XML Schema XML 資料、CSS (Cascading Style Sheet) [8]、UIC (User Interface Customize)，其中 Schema 與 XML 資料為文件本身，Valid XML Editor 由 Schema 動態產生表單，依資料的增加改變介面外觀，再配合 UIC 與 CSS 的使用，將可針對各種 XML 語彙客製化使用者介面，UIC 作用於文件排版（圖七），如：顯示層級、欄位數量、顯示狀態 等，CSS 則可針對文件細部美化（圖八），包括段落、字形、背景、框線 等。

現今 XML 編輯器若能客製化使用者介面，則須透過系統工程人員撰寫程式，Valid XML Editor 透過 UIC 與 CSS 客製化使用者介面，不須依靠程式，一般使用者僅須具備相關基礎知識即能完成客製化作業。



圖六 未客製化前的使用者介面



圖七 加上 UIC 排版的使用者介面



圖八 再配合 CSS 的使用者介面

### 3.7 以樣本檔為基礎的編輯行為

當使用者必須從一無所有開始建立一份 XML 文件，即便系統提供「正確」且「可選擇」的元件給使用者，對一般使用者而言，仍是一個陌生的領域，因此 Valid XML Editor 提出「Templates」的觀念。

Valid XML Editor 完整的樣本檔包括 Schema、XML、CSS 及 UIC 文件，樣本檔有幾個特點，一是配合語義的使用，建立人性化的樣本檔，二是由於樣本檔已加入常用元件，故介面較少變動；基於以上兩點，加以 UIC 排版與 CSS 美化後，即是一份完美的「Templates」。

由於製作 UIC 與 CSS 須具備基礎的 XML 知識與 CSS 能力，故 Valid XML Editor 將使用者區分成一般使用者與 UI 設計人員，讓較熟悉的 UI 設計人員可依該領域的標準建立相關樣本檔，則使用者不只可由 Schema 開新 XML 文件，亦可由 Template 進行簡單的編輯行為，不須更動架構，只要編修內容即可完成一份內外兼具的 XML 文件。

### 4. 結語

隨著 XML 應用越來越廣，使用者也不再侷限於資訊人員，要如何讓一般使用者也能無障礙的編輯 Valid XML 文件是重要的課題，透過 Valid XML Editor 的編輯模式，相信能讓缺乏 XML 基礎知識的一般使用者，亦能輕鬆編輯 XML 文件。

本計劃未來將朝以下兩方面努力：

開發更多的樣本檔：針對各領域已訂定完成的 Schema 建立樣本檔，應用在各領域上將可減少各界導入 XML 所需時間。

開發客製化使用者介面的編輯程式：畢竟 UIC 與 CSS 需要具備基礎的相關知識，本計劃將透過與 Valid XML Editor 相似介面的模

式，降低需要專門知識的障礙，讓客製化使用者介面更簡易。

## 5. 答謝

感謝美商遙軟股份有限公司對本計劃（05T-890701B-CN）的資助。

## 6. 參考文獻

- [1] Altova, XMLSPY, <http://www.xmlspy.com/>
- [2] Architag International, XRay XML Editor, <http://architag.com/xray/>
- [3] Corel, XMetal, <http://www.softquad.com/>
- [4] V. M. Glushkov, “The abstract theory of automata”, Russian Math. Surveys, 16, 1961, pp. 1-53.
- [5] Y. S. Kuo, Jasper Wang, and N. C. Shih, “Handling Syntactic Constraints in a DTD-Compliant XML Editor”, Proc. ACM Symp. Document Engineering, Grenoble, France, Nov. 2003.

- [6] Y. S. Kuo, N. C. Shih, and Lendle Tseng, “Forms-XML: generating form-based user interfaces for XML vocabularies”, in preparation.
- [7] Microsoft, Office InfoPath, <http://www.microsoft.com/office/infopath/prodinfo/default.mspx>
- [8] W3C, Cascading Style Sheets, Level 2, W3C Recommendation, May 12, 1998, <http://www.w3.org/TR/REC-CSS2/>
- [9] W3C, Extensible Markup Language (XML) 1.0, W3C Recommendation, Feb. 10, 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>
- [10] W3C, XML Schema Part 1: Structures, W3C Recommendation, May 2, 2001, <http://www.w3.org/TR/xmlschema-1/>
- [11] W3C, XML Schema Part 2: Datatypes, W3C Recommendation, May 2, 2001, <http://www.w3.org/TR/xmlschema-2/>