

元件型 XML 文件資料庫在臨床文件架構(CDA)上的應用研究

林耀仁

趙家振

育達商業技術學院資訊管理研究所

lyr@ms1.ydu.edu.tw

chao.cloud@msa.hinet.net

摘要

現在已經有許多元件型的 XML 應用標準，例如 HL7/CDA，以高階的觀點定義需求，但現行 XML 資料庫均未能以元件為單位處理資料。本研究依元件型 XML 文件特性，識別出需要能處理文件解構成元件，及元件建構成文件的功能與介面。並開發一個原型系統，定義了「領域維護」、「領域元件集維護」、「類型文件集維護」及「XML 資料庫維護」，來實現解構 XML 文件成為元件，及建構元件成為文件的需求，並以 HL7/CDA 這一個元件型 XML 文件的應用標準，來驗證功能與介面需求。

關鍵字：元件，XML 資料庫，HL7/CDA

Abstract

There are existing component based XML application standards, such as HL7/CDA. They use high-level, component based views to define requirements. But there is no XML database currently can manage component based data. This study is based on feature of component centric XML document, to identify functions and interfaces for decomposing documents into components, and composing components into documents. We develop a prototype system to implement these functions and interfaces, and certify it by the components centric document stand HL7/CDA.

Keywords: Component, XML Database, HL7/CDA.

壹. 緒論

XML 文件常被分類為資料型(資料為主, data-centric)和文件型(文件為主, document-centric)兩大類，前者的 XML 定義必須是完整的格式(well-structured)，類似關聯式資料庫的欄位綱要，大多用來作資料的處理及交換。後者僅包含詮釋資料(metadata)的部份，真正的資料處理是由「內容管理」來完成，而且通常使用的是全文檢索之類的技術。[1][2]

但現在已經有一些介於兩者之間 XML 的應用標準的出現，它有一部份類似文件型的詮釋資料，可以彈性的引用一些資料，但這些資料其實是資料型的元件，被定義得很仔細，例如 HL7 (健康七層組織 Health Level 7) 裡的 CDA (臨床文件架構 Clinical Document Architecture)。

HL7 制定了一系列的臨床資訊的標準，其中以第二版本系列(V2.x)推廣的最為成功。在第二版中，訊息的定義是採「由下往上」的方式定義的，各訊息單位依各自的需求定義訊息，再整合成一整套的格式。第二版也保留了許多的資料欄位供各醫院擴充之用。但這些保留欄位造成互相交換時，即使雙方擴充了相同的功能，也因各自擴充而不能直接交換，必須另外製作一個交換界面。

第三版(V3)就這個問題，及其他的考量，建構了一個標準的參考模型 RIM (Reference Information Model)作為基礎。第三版使用 UML (Unified Modeling Language)方法論和 RIM 來定義訊息。RIM 是第三版中必要的元件，提供精確的語意和文法的表示規則，來表達 HL7 訊息中的所涵蓋的資訊。

臨床文件架構 CDA (Clinical Document Architecture, 或病歷文件架構 PRA, Patient Record Architecture)提供了另一個以病歷交換臨床文件(例如出院病摘)，作為接近實際醫院運作的電子版本模式。

新版本的 CDA 引用 XML，HL7 V3 的 RIM 和語彙，將一份文件視為一群元件的集合，一個元件則相當於一個傳統的訊息。CDA 文件同時包括兩個部分，提供設備讀取的資料，以及提供使用者閱讀的資訊，以便單一文件可供人及設備同時使用。CDA 亦可以在 XML 的瀏覽器上，包括 PDA 和手機，以不同的樣式 (Style Sheet) 方便的瀏覽。[3][5]

目前以資料處理作法的 XML 資料庫無法有效的處理元件型 XML 文件，而是以文件型的 XML 資料庫為主要做法，在資料庫這一個層級並不特意處理元件內部的資料，而是提供儲存、全文檢索、XML 查詢等文件存取的界面，再由應用程式來完成一切的功能及需求。雖然說這些功能與需求有許多是跟應用相關的(application-dependent)，但應用程式仍然要負擔許多文件與元件的拆解、尋找、重組、管理等一般性的工作，相對造成應用開發的障礙。

從直觀上來說，元件型 XML 文件像是一個文件夾，而不是像傳統 XML 的觀念是一個檔案一個文件。例如每位病友每次就診都有一個病歷紀錄，在 CDA 上線後，每個病歷紀錄即為一個 CDA 文件，內容包含病友之基本資料及診斷治療所相關的紀錄，如主訴、檢驗及醫令等。如果我們需要某位病友在醫療過程中，某項檢驗(例如心肺檢查)的歷次檢驗的摘要報告，不只是把所有的 CDA 從資料庫裡調出來，而是希望有一份就只有該項檢驗資訊的新的 CDA 文件。

本研究即是以元件管理的精神，分析元件管理的需求、功能及界面，並以開放原始碼之XML資料庫Exist為基礎[4]，實作具元件管理能力的元件型XML資料庫雛型。除了文件型XML資料庫的傳統功能及元件管理的能力外，並提供對架構文件及元件的內容資料處理的能力，以及全文件與元件之間的組合關係的操作。

本研究以對標準化推動較積極，有較多公開的標準測試資料，並且比較容易區分應用相關技術及一般資訊技術之元件型XML文件標準CDA為對象，分析元件型XML資料庫所應提供的功能及界面，並且根據開放原始碼的XML資料庫及Web伺服器，挑選合適的系統，加以包裝所需的機能，並且根據應用建立使用範例，作為示範之用。

本研究將提供未來元件型XML文件在資料交換上進一步的發展，使得未來資料交換的規劃可以更偏重於大架構及 metadata，而元件化的資料交換將使得資料交換更具彈性，交換項目的變動只要更動元件，而不必造成整個標準的變動。

貳. 引用詞彙

本文所謂的元件，是類似於XML片段(fragment)，都是XML文件中的部份，但是元件有更多的意涵，如重複使用性、符合XML定義的良好格式(Well-formed-ness)，是為了編輯、格式化而截取的適當區塊等。

元件之所以異於片段，在於它有一個描述元件的Metadata稱為領域元件，它除了記錄資料庫存取的位置，也記錄當載入XML文件時，系統如何識別那個部份是屬於那個領域元件管轄。例如「出院病摘報告」中，有段「入院診斷」敘述，會被系統定義成為「出院病摘報告—入院診斷」的領域元件，當載入「出院病摘報告」文件時，有關於「入院診斷」的元件，就會被集中管理。

除元件外，本研究基於元件管理的觀點，定義了許多詞彙，茲說明如下：

- 集合(Collection)：XML資料庫用來承載子集合或資源的虛擬容器。
- 資源(Resource)：可以儲存在XML資料庫中的物件，以eXist而言，它支援XML文件及二進位物件。
- 領域(Domain)：特定的範圍。領域用來區隔不同的XML應用標準制定組織所提出的標準(如HL7/CDA)，並約束領域內的XML實例文件，需符合該標準的定義。
- 元件(Component)：有意義可以重複使用的XML子樹。
- 領域元件(Domain Component)：描述同性質元件特性的詮釋資料，它記錄了給人看及給系統看的資訊，其中給人看就是特性的說明，而給系統看的是元件存放位置。
- 領域元件集(Domain Components)：領域元件的聚集體，以XML文件型態儲存在資料庫內。

- 類型文件(Class Document)：堆砌元件成為文件時，系統記錄的資訊。資訊大體可分為領域元件、結構及條件等資料。
- 類型文件集(Class Documents)：類型文件的聚集體，以XML文件型態儲存在資料庫內。

參. 研究內容

本研究其中會引用XML文件應用標準HL7/CDA，雖然元件型XML文件應用並不侷限於HL7/CDA，但引用的目的在避免研究結果與實務應用無法配合。研究重點分述如下：

1. 領域畫分：因為不同的XML應用標準制定組織，所提出的標準(如HL7/CDA)個有特色，若資料庫不予適當區隔，會造成不必要的衝突，也會影響未來擴充性，所以用領域畫分不同。
2. 文件驗證(Validation)：XML應用標準制定組織提出的規範，太多會提供W3C XML Schema (XSD)或Document Type Definition (DTD不在本研究討論)，來驗證XML文件是否正確合法(Validity)，驗不驗證會影響系統分割、查詢文件及組合元件。
3. 元件的定義：切割XML文件的方式不同，會影響XML介面與功能，什麼方式適合本研究，必需分析研究。最後結果的XML文件如同一棵樹狀結構，元件是樹中的子樹，是由元素節點切下來的，單獨的子樹必需符合良好格式。元件不能被亂切，它必需遵某個規範，由人介入設定切割規則，且只要設定一次。
4. 描述同性質元件的詮釋資料：元件只是符合良好格式的XML，系統不會讀取它而知道應該做什麼，人也不會想要直接讀它，必需有個抽象的敘述，來告訴兩方有關於同性質元件的特性，而這個抽象的敘述，內容要如何表達，才能支援研究目的。
5. 文件解構成元件：當有了文件切割設定資訊，及描述同性質元件的詮釋資料的定義，系統如何執行切割並將元件存入資料庫端，如何維護同性質元件的詮釋資料，才能充份表達文件切割成元件。
6. 元件建構成文件：有了描述同性質元件的詮釋資料，提供人易懂的資訊，使用者可以直接操作詮釋資料，排列新文件的結構，另外可再賦予條件使結果更符合需要。當使用者完成新文件的設定，系統如何取得符合條件的元件，依前設定的文件結構，產生新的文件。

肆. 系統架構

元件型XML文件的特徵有三，一是，它有一個由元素組成的結構(簡稱結構元素)，結構得被定義成數個層級，例如款、項、目、節，或頭文件、主體。其二，結構元素會被定義可不可以掛載元件。其三，元件會被資料模型清楚定義。

當有了這些特徵，就可以從XML文件中，分析出元件的特性。而這些特徵需要由使用者告訴系統，在此稱為「領域參數設定」。當處理分割XML文件成為元件時，其分割順序，是依據參數中結構層級設定切割，而切下來的子樹只是結構的表示，並不足以表達元件，需從子樹中找到元件類型相關資訊。此等相

關資訊在此稱為「領域元件」，可以提供系統儲存元件的指示，及方便使用者判讀的資訊，且會被系統儲存管理。

當處理組合領域元件成為新的 XML 文件時，需要提供階層式的排列順序，如此才能產生樹狀結構的 XML。為了產生更有意義的結果，排除非需要的部份，得加入條件。必為了能夠重複使用這些設定，得命名儲存，在此稱為「類型文件」。

系統主軸就是要能處理文件分解成為元件，元件重組成為文件，而要能處理元件，先要設定前述的資料並儲存（設定領域參數），使能夠達成「一次設定無限再用」的目標。其次要有載入 XML 文件功能，當系統執行此功能時，會參考設定接續執行分割 XML 文件動作，再將元件存入資料庫，另外也會將元件的特性記錄在領域元件中。而當重組文件時，首先操作排列領域元件，使系統了解到何處處理取元件，及取回後排列方式。其次決定是否要加上條件，使取得的資料更為所需。最後交由系統，產生新的 XML 文件。

為使系統更加週延，加入領域維護，使系統不因多領域而產生衝突，且可以在領域內載入 XSD 文件，對爾後載入的 XML 文件可以進行驗證。另外因為系統延用 eXist 的安全管理機制，使用者必需設定連結上資料庫，才能使用其他功能。

系統區分為四大子系統，分別為領域維護、領域元件維護、類型文件維護及 XML 資料庫維護。其主要功能分別為劃分領域、切割 XML 文件成為元件、重組元件成為文件及資料庫管理。其系統架構圖如下

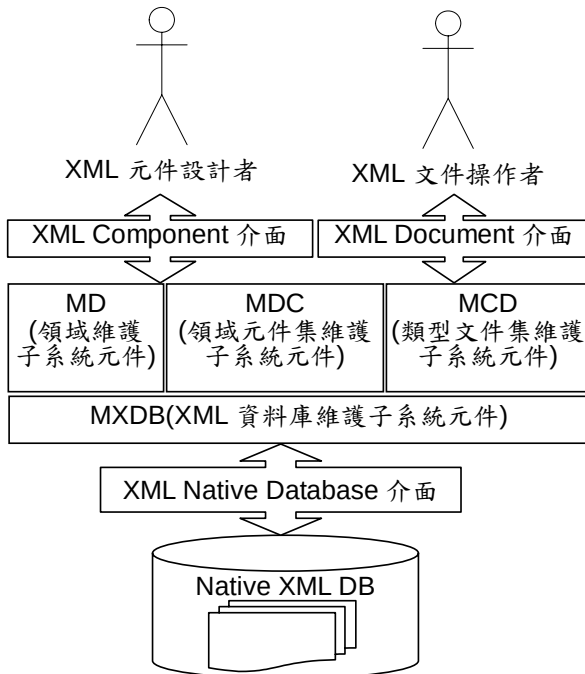


圖 1: 系統架構圖

伍. 研究結果

本章節主要說明投入資料，及經過系統操作後的產出，來說明在元件型 XML 資料庫中扮演的角色，並分別按照以下小節說明

一. 領域維護

1.1 產生領域

產生領域視窗畫面如圖 2，當按下新建按鈕，就會產生新的領域，而資料庫端也會產生必要的集合，用來承載後續作業所需子集合或文件。

當選取一個領域後，如 CDA，系統會視為進入領域，來提供後續作業一個進入點。

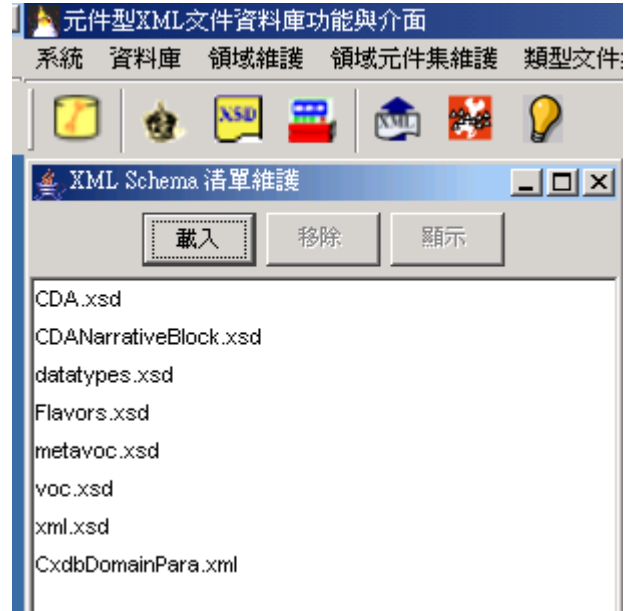


圖 2: 載入 XSD 視窗畫面

1.2 載入 XSD

當從外部載入 XSD 後，即賦予系統對該領域有驗證的能力，載入 XSD 視窗畫面如圖 3。

1.3 參數設定

使系統得以對後續載入的文件，進行切割產生元件。當完成編輯儲存後，它會在資料庫端建立領域參數文件，達成一次設定的目的。

二. 領域元件集維護

領域元件集維護視窗畫面如圖 4，當載入 XML 文件時，視窗畫面會顯示處理狀況等資訊。其主要功能是将載入的文件，按照參數設定執行元件分割的工作。

三. 類型文件集維護

類型文件維護視窗畫面如圖 5，提供元件建構成為文件的介面，當從選單中選取特定類型文件時，會將之前存檔的類型文件取出，並顯示在建構區及條件區。圖 5 為當要產生一個診斷段落，並限制病友名字為 Henry 的文件時，按下產生 XML 文件按鈕，會得到組合之 XML 檔案(為 CDA Body 部份)。

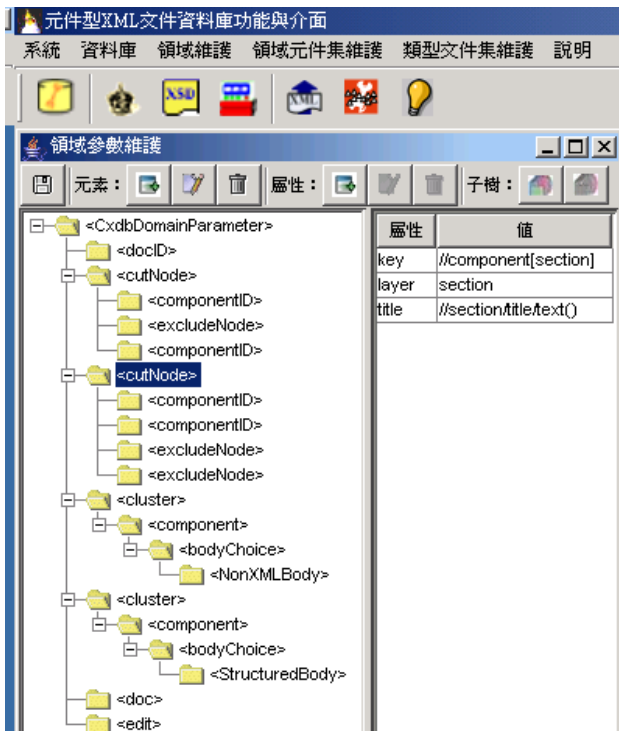


圖 3: 參數設定視窗畫面

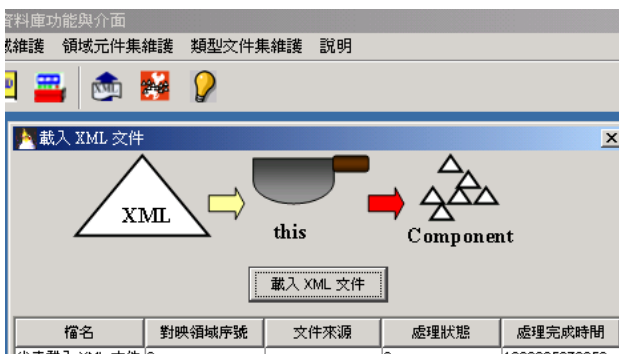


圖 4: 領域元件集維護視窗畫面

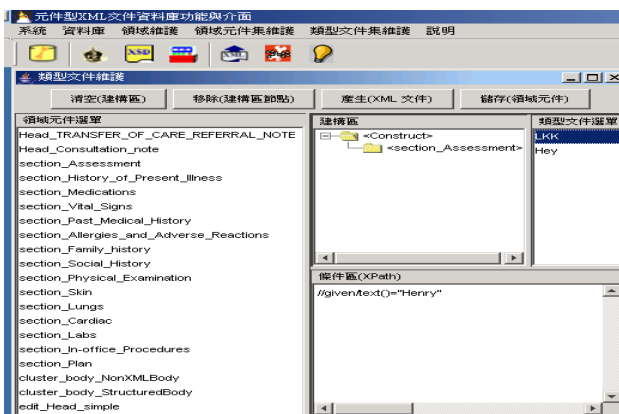


圖 5: 類型文件維護視窗畫面

陸. 結論

目前實務上對的 XML 準則的制定，不僅要用資料模型定義元件，也要有足夠的彈性架構來掛載元件，元件型的 XML 應用已是時勢所需。但從高階觀點定義

元件的應用，如果實作上還是停留舊方法，不是存取整份文件，再篩選重組，就是徹底支援文件，再由元素、元件併成文件，都不是適當處理方法。

本研究以另一個角度思考，只要一次設定，就可以按人的角度，將文件分割成為元件，而且只要分解一次。重組元件成為新文件時，是以元件為單位建構文件。且當環境改變準則修改，只要修正設定不用改程式，其所帶來的彈性與及時性，足以跟上應用環境進步的腳步。

本研究依元件型 XML 文件特性，識別出需要能處理文件解構成元件，及元件建構成文件的功能與介面。並開發一個原型系統，定義了「領域維護」、「領域元件集維護」、「類型文件集維護」及「XML 資料庫維護」，來實現解構 XML 文件成為元件，及建構元件成為文件的需求，並以 HL7/CDA 這一個屬於元件型 XML 文件的應用標準，來驗證功能與介面，並證明符合系統規劃時需求，達成一次設定無限再用的目標。

柒. 討論

本研究在元件型 XML 文件應用中，尚屬基礎階段，著重於 XML 文件的元件的分解與組合操作。未來展望有以下三點：

- 提供格式化元件的功能，如此可以配合人或裝置，提供適合的視覺效果，與操作方式。
- 提供資料交換的功能，當大家都遵循特定規範，經由系統定義的領域元件也會一樣。如此資料交換可以用元件的角度，組成交換文件，而更能符合需要。
- 在應用層提供更多的事件處理，使文件能依商業邏輯(business logic)重組元件，產生適當文件。

捌. 致謝

本研究承蒙國科會自由軟體專案計畫「元件型 XML 文件資料庫的元件管理介面之研究」(計劃編號 NSC-92-2218-E-412-001)支持，特此致謝。

參考文獻

- [1] Bourret, Ronald. "XML Database Products," Internet article of <http://www.rpbouret.com/xml/XMLDatabaseProds.htm>, 2004.
- [2] Chaudri, Akmal B., Rashid, Awais., Zicari, Roberto., "XML Data Management: Native XML and XML-Enabled Database Systems", Addison Wesley Professional, 2003.
- [3] Health Level Seven Structured Documents Technical Committee, "Documents and Presentations," Internet Article of http://www.hl7.org/lib_admin/docs.cfm?dir=library\committees\structure&comm=structure.
- [4] Meier, Wolfgang M., "eXist: Open Source XML Database," Internet article of <http://exist-db.org/>, 2004
- [5] Piggott, David. "Reporting Experiences from Using the HL7 Clinical Document in the PICNIC," Technical report of Professionals and Citizens Network for Integrated Care, 2002.