Accelon 2017開發後記

這篇文章記錄了自2009年到2017年4月，開發Accelon的歷程，
可視為拙文Accelon, 一個開放的數位古籍平台的後篇。

兩大根本問題

在從事電子佛典的工作以來，有一類問題特別引起我的注意，
這類問題不一定很急迫嚴重，但它或隱或顯，普遍存在於系統的每一個環節，
並且不會隨著技術的發展，例如：網路傳輸速度、運算速度的提昇、儲存容量的擴大而得到解決，
其中有兩個代表性的例子，一是文字編碼缺陷，二是文件編碼缺陷。

文字編碼缺陷（即缺碼問題，俗稱缺字問題）

缺碼問題，隨著Unicode的擴充而得到舒緩，但並沒有得到根本的解決，
缺碼問題的原因和理論上的解法，謝清俊老師已有完整的闡述。
時至今日，主流的作業系統還是沒有漢字構型的制式表達法，
我們依舊無法隨心所欲地創造新字或是表達錯字。
我在1999年就意識到，要徹底解決缺碼問題，最後一塊拼圖是字形產生器，
也就是從一維的漢字表達式，產生二維的字形。

2002年在易符科技的資助下，完成了從IDS到「單線體」的字形產生器，並應用於Accelon3。
但要產生美觀的字形難度極大，就好比教小學生寫字容易，但寫出漂亮的書法就要多年的苦練。
美觀度很大程度取決於部件的比例和佈局，而無論是IDS或是中研院的構字式，
都只規定了部件的相對位置，而沒有包括比例資訊，
當我在2015年想通了這一點，改用「減法」來表達字形，終於解決美觀的問題，
比方說要表達「初」字少寫一點的錯字，表達式是「初ネネ」，
其意義是「基字,減去,替代為」。
字形產生器只要將「初」字裡頭的「ネ」字偏旁替換為「ネ」即可。
由於「初」字中的「刀」已由字形設計師手工微調過（撇往左下角延伸），
比直接用「刀」去拼「ネ」的效果好得多。
「替代為」也可以是式子，如此就可以遞迴地表達如招財進寶的複雜字形。
有興趣的朋友可以看這一段視頻。

接下來的任務就是將這個機制整合到作業系統的圖形介面層，我已將相關技術轉移給一團隊，
希望在不久的將來，無限的漢字得以自由穿梭在所有的電腦及行動裝置上。

文件編碼缺陷

2009年以前，我對這個問題的認識不夠清楚，天真地以為只要有一個像CBETA的組織，
承擔統一經文格式、精細標記的工作，其他人就可以在這個基礎上，自由開發各種應用，
經過多年的實踐和思索，我發現以XML+TEI做為經文的基礎格式，
固然省去了反覆輸入文字和目錄的勞務，但對於經文的進一步加值，反而是一個阻礙。

加值

什麼是對經文的「加值」？「加值」泛指以經文為基礎的延伸創作，
常見的例子有「畫重點」「眉批」「腳注」「注疏」「校勘」等等。
以「畫重點」為例，形式上可以用「畫線」「每個字底下畫小圈」「螢光筆」等等。
「加值」的主要目的是協助自己或他人理解，由於經文是固定的，
而語言一直在變，所以用當代人能夠理解的語言和形式來解釋經文，需要每一代人前仆後繼。

此外，「加值」無法脫離經文而獨立存在，一段重點畫線，只有畫在經文上，才具有意義。
一直以來，我們只關心經文本身的數位化，殊不知「加值」往往比經文更有價值，
就好比同樣的教科書，寫滿筆記的比全新的更有價值。

由於數位化的重心一直都只在經文本身，而沒有充份考慮讀者加值的數位化，
因此電子佛典經過那麼多年的發展，除了全文搜尋和剪貼的便利，
能做的事並不多，除了藏經之外，許多古德先賢的注疏還沒有數位化，更談不上彼此的互相參照，
而今人無數的讀經筆記、講義，個個都困在名叫docx的孤島，老死無法往來。

連結

在所有的經文加值之中，最關鍵也是最難實現的是「互文連結」，
即任意兩段文字之間的連結，比方說「引文」和「出處」就是一個典型的互文連結。
互文連結之難以實現，根本原因在是 HTML/XML 採用樹狀內嵌標記。
樹狀的意思是，XML文件在記憶體中是以樹狀結構形式存在(即DOM)，這也是標記不能重疊的原因。
而內嵌的意思是，標記與被標記的文字，同屬一字串。
同屬一字串的好處是，標記會緊緊黏住所標記的文字。
這對持續編修的文字來說，的確很便利，但對經文這種不變的文字來說，沒有太大的效益。

早在TEI的設計之初，謝清俊老師就指出採用內嵌式標記文件是一個錯誤的方向，
我第一次聽到這個說法是十幾年前在老師家，閒聊電子佛典的前景，
當時我已將整部大正藏的搜尋時間壓縮到一秒以內，也就是使用者一輸入完畢，結果就出來了，
我問：「接下來能做什麼呢」，謝老師答：「做經文的連結」。

這裡講的連結，和一般熟知的網頁連結不同，網頁的連結，只能從一段文字，
連到另一個網頁(或預先設好的錨點)。
由於XML標記是內嵌的，無論是連結的出發地，或者連結的目的地，都必須改變文件本身，
換句話說，只要加一個連結，就要改動兩個檔案。
光是大正藏就有八千多卷，再加上引用大正藏的大量著作，如果每次建立一個連結都要改動一次經文，
檔案的版本控管就是極大的挑戰，即使全部用git管理，產生出來的XML文件也會複雜到難以想象。

因此，標記必須獨立於經文之外，也就是說，經文和標記分開儲存，
以保證增加標記時，不會（也不應）改變經文檔案。

定址系統

為了將標記從經文剝離，經文就必須能夠做到「精確到字」的定址。
而這個定址方式，必須同時對機器和人都有意義，
從反面來理解，大正藏第1234415字，對電腦很有意義（知道是那個字），
對人沒有意義，這就不適合做為定址方式，而必須採用分層編碼。
起初我參考聖經的「書、章、節」，
想要編製一套跨各種佛經版本的多層分段系統，
但光是處理巴利大藏經，就吃足了苦頭。

分層定址

直到2016年中，終於發現經文分段是不可行的，
一來是要分好段，就必須對文意有準確的把握，工作量實在太大，
而即使將經文都分了段，恐怕也需要過一兩百年才會成為標準，
於是我放棄「跨版本」的想法，改為「基於版式，加以擴充」的方案，
剛好得到了菩提乘基金會智翰法師的贊助，讓我可以全力攻克這個難關。
經過大約半年時間，終於開發出「Dhamma Positioning System」，
這是一個四層的定址系統，分別為「冊、頁(欄)、行、字間」。
以大正藏為例，「冊、頁(欄)、行」就是和CBETA採用的行首格式，
而字間就是鍵盤游標可以停駐的位置。
大正藏有100冊，至少要7bits才可容納，
而最大頁數為1464頁，每頁有三欄，就是4392，需要 13 bits，
每頁29行，需要5 bits，每行 19 字，即20字間。也是5 bits，

標記的起點與終點，合理的假設都在同一冊，那麼大正藏剛好可以用 7+13+5+5 13+5+5 = 53bits，
當我算得這個數字時非常高興，直呼菩薩保祐，因為這是Javascript不失精度，所能表達的最大整數，
每個外部標記的位址的表達不必用字串，只須用整數型別，
整數型別是最有效率的處理單元，這對未來運算大量標記有重大意義。

定址系統的原理很簡單，但意義非凡，就好比經緯度座標對地理資訊系統的重要性，
在定址系統的支持之下，非常輕易地實現多年來夢寐以求的高級功能。

Accelon 2017

2008年 Google 推出Chrome，2009 年 node.js 發布並迅速取得成功，
2010 iPad 問世並確定不支援Flash格式，種種跡象表明，
Javascript 大勢已成，將會是跨所有平台的唯一語言，
當時，Accelon 4 核心程式的開發已大致完成，
我做了一個艱難的決定：以Javascript 重寫所有的程式，
這不但意味著推倒重來，還要學習新的程式語言和開發環境。

Javascript的生態和由私人公司主導的環境(如Visual studio, Delphi 之類)有很大的不同，
在初學Javascript 的頭兩年，感覺就像花木蘭「東市買駿馬，西市買鞍韉，南市買轡頭，北市買長鞭。」

得自己拼湊開發環境，往往是剛選定、學好一個套件的用法，沒多久又有更好的套件，
而有學不勝學之嘆。

這五年來幾乎天天寫Javascript，數不清換了幾種套件，選定了 React/React Native + mobx
來發展Accelon 2017前端程式，目前來看，還算滿意。

距離上一代的Accelon3，已過去了將近十五個年頭，Accelon 2017最為關鍵的突破，
就是以「定址系統」為基礎的「文層」以及「逆向連結」。

文層

「文層」是我借Photoshop「圖層」概念而來，要體會圖層有多重要，
只要執行 Flatten Layers (壓平圖層)的功能，再嘗試編輯就會明白。
TEI就像是功能極為複雜，但沒有圖層概念的「小畫家」。
而基於文層的的實作，目前雖然還很原始粗糙，但假以時日，必然大放異彩。
就好比第一代的汽車，時速極低，操作複雜又常出狀況，還得有工程師隨伺，遠不如馬匹方便。

在Accelon 2017 中，底文是不變的文字，而不同類型或是不同作者的標記，
儲存在互不干擾的文層，除了底文之外，其他文層都可以自由開關，
Accelon 2017 會將選定的文層，合併渲染成HTML。

逆向連結

在Accelon3 就有了逆向連結的想法：從印順導師的著作的引文，跳到大正藏經文，這是順向連結，而從大正藏經文，回到導師引用之處，是逆向連結。Accelon3 只能知道某一個大正藏的頁，被那些導師著作引用，精度不高，用處不大。

在Accelon 2017，正向連結是存在於引用端資料庫中，也就是導師全集資料庫中，
而逆向連結是即時計算而得，也就是說，大藏經資料庫並沒有記錄被誰連，
而是在打開導師全集資料庫時才計算，這個機制的威力在於，
大藏經的檔案不必更新，隨著連結的增加，
我們慢慢會現在大藏經中那些是連結佛學著作的樞紐。

樞紐

假設這個世界有一千個機場，若要直接連結每個機場，需要近50萬條航線，
設置了轉運中心，即樞紐機場，僅須要幾千條航線，所有機場都可以互連。

想像每篇經文或注疏就像一座座的機場，若沒有樞紐，彼此的連結、參照極為不便，
而大藏經中存在某些經文，具有樞紐之功能，我們的任務是將它發掘出來。

由於引文和大正藏經文經常略有出入，必須人工逐一確認，
目前已完成了導師全集和大正藏經文的互文連結。

從這些互文連結，很容易知道那些大正藏的經文段落是常常被引用的，
其中引用次數最多的是這一段，共27次：（大正2，67a05-06）
「此生故彼生，謂緣無明有行，乃至生、老、病、死、憂、悲、惱、苦集；」

導師經常引用經文，就是樞紐，並提供了理解導師思想的鑰匙。

什麼是全文檢索？就是將相隔很遠的相似詞組，彙集到視野之內（搜尋結果）。
而互文連結，可以串連相似語義和主題，這是基於「語形」的全文檢索無法企及的境界。

目前我們只建立了導師到大正藏不到兩萬個連結，
閱讀某本著作，引到某處經文，跳到該處經文，
系統會自動顯示另一本著作也引用同一段經文，讀者可以快速從經文跳到該著作。
未來隨著更多祖師的著作的加入，互文連結會慢慢連成一個綿密的網絡，
利用大數據和圖形視覺化等工具，很多過往難以偵知的規律和關係將會被揭示，
佛典的閱讀和理解，也將會有完全不一樣的風貌。

資料加工鏈條

工業化的本質，就是從初級原始材料，經過一系列的加工，製造出高級產品。
數位化也是如此，原始資料必須經過一系列的加工，才會產生有用的資料庫。

Accelon 2017 的加工鏈條，主要有三個階段：
一）提煉 corpus-refinery
　　加上符合原書的頁碼和換行。
　　文字對照原書圖版校對。

二）鍛造 corpus-forge
　　加入內建標記，產生 cor 檔案。
　　既有的XML/TEI，則只須撰寫轉換程序，通常不必重新標記。

三）連結 corpus-connect
　　建立互文連結。（未完成）

格式

　　在數位世界，時空幾乎沒有距離，距離主要是由格式造成的。
　　大數據時代，數據量不是問題，格式不統一才是問題。
　　數位化，工具的選擇其次，最關鍵的是格式。
　　好的格式須同時滿足兩個條件：
　　　　「人容易編輯」
　　　　「機器容易剖析」

　　純文字格式，docx 等格式，滿足前者，不滿足後者。
　　TEI 滿足後者，但不滿足前者。
因此我創造了htll 格式 (hypertext label language)，有點像markdown，但更簡單也更靈活。請參考[十三經豎排資料庫]() 。

htll 的設計理念：
　　1)讓熟悉內容的人方便編輯，記憶負擔少，技術門檻低，不容易出錯。
　　2)以古文沒有用到的半形符號和abc做為標記符號，方便輸入，占的視覺空間也少。
　　3)方便豎排，以利與原書圖版對照。

4)完全自由標記語法，使用者可以根據資料的特性自由創造語法，只要regular expression 可以無歧義地轉換為XML即可。

github: https://github.com/accelon/accelon2017

延伸閱讀：
TEI 的種種問題，學者 Desmond Schmidt <link to="https://jtei.revues.org/979">總結</link>得很好，值得一讀。

超連結：Ted Nelson 所著文章。