# 標準通用標誌語言基本概念

# The Basic Concept of Standard Generalized Markup Language

陳 昭 珍

*Chao-chen Chen*

輔仁大學圖書資訊學系

**Dept. of Library & Information Science**

**Fu-Jen Catholic University**

## 【摘要 Abstract】

本文主要在介紹國際標準通用標誌語言（Standard Generalized Markup Language, ISO-8879）標準之源起、架構、應用及其他文獻處理相關標準，並說明圖書館界了解此標準之意義。

The main objectives of this article is to introduce the background, architecture and application of SGML (Standard Generalized Markup Language). Other related document processing standards are also discussed. We explain why the standards of SGML is a must for working librarians.

**關鍵詞**：標準通用標誌語言　超媒體同步語言　文件格式、語意及規範語言
標準版面描述語言　超文件標準語言　標誌
Standard Generalized Markup Language, (SGML);
Hypermedia/Time-Based Structuring Language, (HyTime);
Document Style Semantics and Specification Language, (DSSSL);
Standard Page Description Language, (SPDL);
Hypertext Markup Language, (HTML);
Markup

## 一、SGML之發展背景

要明瞭SGML之發展背景，首先須說明何謂「標誌」。在傳統出版過程中，版面編輯（copy-editor）必需在草稿中加註，說明希望呈現之版式、字體、空行、縮格等指示，這些加在草稿上的指示，就是傳統的標誌。

六〇年代至七〇年代間，電腦應用普遍，出現了一些處理文件的程式，如IBM的Script，Waterloo Script，及UNIX1的nroff系統等。這些系統都是以批次(batch)模式來處理輸出資料的版面程式，將特殊標誌指令(specific markup commands)加在文件中，以指示需處理之動作。所謂特定標誌指令，是指每一系統皆有其專用之指令語言，若使用不同系統，即須一一學習各種系統之標誌語言。這類標誌方式之缺點有三①：

㈠會遺失文件之屬性資訊：例如，若使用者標註將標題及圖形說明文字置於中央，系統並無法分辨它所處理的是標題還是圖形說明文字，所以這項屬性資訊即無法應用於檢索系統。

㈡程序性標誌缺乏彈性：當使用者決定更改文件之版式或機器時，必須重新標誌。

㈢以控制指令來標誌，相當花費時間，也易出錯，同時，使用者並須受過高度之訓練。

個人電腦普及後，電子文件處理又向前跨了一步，親切且能與使用者互動(interactive)的文字處理系統出現了。這種系統最大的特色稱爲WYSIWYG（What You See Is What You Get)，意即您在畫面上所看到的樣子，也就是您所得到的輸出結果。如：Wordstar, MacWrite, Microsoft Word及Wordperfect等都是屬於這類系統。

WYSIWYG系統雖將很多標誌指令轉爲隱藏式指令，並能即刻呈現標誌後的結果，但它並未記取批次標誌之經驗，甲系統標誌過後之文件轉換到乙系統後，仍會遺失屬性資訊，版面標誌資訊也會失效。因此，高華伯(C. F. Goldfarb)、莫蕭(E. J. Mosher)、勞瑞(R. A. Lorie)等人即提出通用標誌(generalized markup)之構想，這個構想有兩個重要的主張②：

㈠標誌須描述文獻之結構及其他屬性，而非指示如何處理文件，這種描述性標誌(descriptive markup)只須做一次，即可滿足未來之處理需求；

㈡標誌需嚴謹，以便滿足需嚴格定義物件之程式或資料庫系統處理之需求。

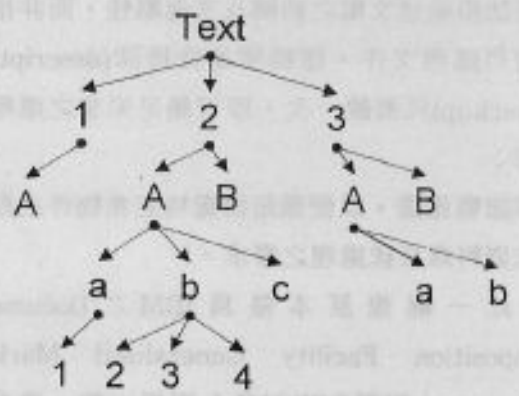
此一構想原本是爲IBM之Document Composition Facility Generalized Markup Language（簡稱DCFGML）而提出的，後來經國際標準組織之文件與辦公室系統委員會(Text and Office Systems Subcommittee of the International Standards Organization)擴充發展爲標準通用標誌語言（ISO 8879－Information Processing－Text and Office systems－Standard Generalized Markup Language (SGML)）。

## 二、文獻結構

何謂文獻結構？若以電腦處理角度言之，可將資訊分成兩種：一爲欄位化資訊，如圖書館的編目資料、統計表格、名錄等；另一則爲非欄位化資訊，這種資訊通常可以樹狀結構表示，故亦可稱爲樹狀結構(tree structure)文獻資料，如期刊論文、出版的專書等。樹狀結構文獻若以電腦建檔，即所謂的電子書，若設計成可檢索之資料庫，則爲全文資料庫。無論是爲電子出版，或爲建立全文資料庫，都必須經過一連串的文件處理(text processing)程序。

沙特(Generad Salton)認爲文件處理過程包括：文字處理(word processing)、文件儲存與檢

索(text storage and retrieval)、應用環境的建立。同時認爲，任何文獻無論其儲存媒體爲何，都可視爲一完整的樹狀結構實體，此實體可以下圖表示之③：




圖一　文獻之樹狀結構圖

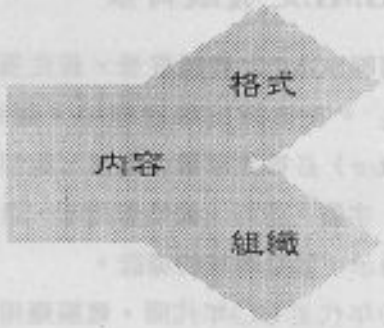
若此實體爲傳統之書籍，則文件大致可分爲：章→節→段→句→字。

1985年，皮爾斯等(Arno J. N. M. Peels)將文件處理過程分爲：建立(creation)→結構化(structurization)→格式化(formatting)→呈現(presentation)；並將文獻結構分爲邏輯結構(logical structure)及實體結構(physical structure)二部份④。

侯特(William K. Harton)認爲任何文獻所包含之資訊內容，不外三種類型：1.內容(content)；2.格式(format)；3.組織(organization)。

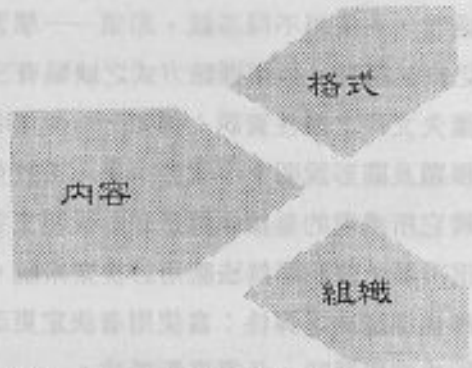
內容是文獻之主體，它可能是文字、圖片、或聲音等。格式則決定內容如何被顯示，它包括字體、版式、色彩等等。組織則說明文獻內之論題(topics)的順序及其交互關係，例如論題X是論題Y的一個子題，論題A應在論題B之前等。

在一紙本文獻中，上述三種資訊內容是不可分離的，如下圖所示⑤：




圖二　紙本文獻中三種資訊之關係圖

但在線上電子版本中，則將三者完全分開，系統設計者必須分別管理這三種元素，但又需使之能組合在一起應用，如下圖所示：




圖三　電子版本中三種資訊之架構圖

若由使用者的角度而言，文獻通常包含兩種結構：一爲明顯結構(explicit structure)，一爲隱晦結構(implicit structre)。明顯結構是文獻的形式結構(formal structure)，例如不同層次之標題(headings)、明晰之參照(cross reference)、及目次等均屬之；隱晦結構則指文獻內容之屬性及相互關係，例如作者由兩個不同的觀點來探討同一主題，正文之注釋具某類別屬性等等。隱晦的部份通常是作者眞正希望讀者了解的部份，同時讀者也需要具備相當之背景知識，才能完全心領神會，不過在轉換紙本資訊爲電子版本時，這種隱晦的資訊內容往往容易被忽略而漏失掉。

## 三、SGML之特性

何謂SGML？簡單言之，它是一套描述文獻結構的超語言(metalanguage)，而非可以執行的電腦系統，SGML四個字所代表之意義分別如下：

**S (Standard)**－表示它是一個國際標準。

**G (Generalized)**－它考慮的是文件在電腦之間的傳輸，適用於任何類型之應用環境。

**M (Markup)**－以開始與結束指標清楚的定義文獻之元素。

**L (Language)**－它是一個可被電腦與人清楚解讀之文獻描述語言。

就特性而言，SGML與其它標誌語言不同之處，有下列四項⑥：

㈠SGML著重於描述標誌而非程序性標誌，使得同一文獻可應用於不同的軟體系統，如，若將附註標示出來，可滿足甲系統欲將附註置於各章之後，乙系統欲將之置於全書最後之不同處理；若將該文獻中之人名、地名標示出來，可滿足系統將之排列爲書後索引，或成爲檢索系統之索引檔的不同需求。

㈡文獻類型概念

SGML提出文獻類型及文獻類型定義(Document Type Definition，簡稱DTD)之主張，文獻之類型乃依其組成部份及結構定義之，例如信件包括：收信人、寄信人、日期、住址、信件內容等；技術報告包括：題名、作者、摘要、正文之篇章段落等。若知道文獻之類型，則可利用剖析器(parser)來處理，並核對該文獻之必備部份。

㈢資料獨立(data independence)

SGML的基本目標之一，是希望根據其規則加上標誌符號之文獻由某一硬體或軟體，傳輸到另一硬體或軟體時，不會遺失任何資訊。在此，SGML以字串取代(string substitution)法達此目標，由此方式定義之字串稱爲實體(entity)。

㈣文件結構

文件因同目的之需，可分爲不同類型、大小之單元，例如，一篇散文可細分爲：節、章、段。這種結構性單元可以用來辨識文件中某一資料之位置，或做爲特定分析之用。文件結構單元只要有意義即可，彼此之間可以重疊。

## 四、SGML之結構

㈠標誌之種類

由上述SGML之發展背景得知，所謂標誌即指加到文獻中以傳達有關此文獻之資訊。在SGML中，加到文獻內之標誌可分爲四類⑦：

1.描述性標誌（又稱爲標誌符號"Tags"）

這類標誌符號旨在定義文獻之結構，是最常用也是最重要的標誌。

2.實體參照(Entity Reference)

在一電腦系統中，往往將較長之文獻分爲幾個部份存檔，這些分別儲存的檔稱爲實體(entity)。分離的實體若要連接，須在文獻中標誌註明，這種標誌即爲實體參照。

3.標誌宣告(Markup Declaration)

這是用來控制應如何編譯(interpret)標誌的敘述，它們可以用來定義實體及文獻類型定義。

4.處理指令(Processing Instructions)

這是指示處理系統做一些特定動作的指令。這種標誌和上述三種不同，它因系統而異(system-dependent)，也因應用而異(application-dependent)，當文獻須在不同之系統處理，或欲改變輸出版式，則需更改此處理指令。

SGML系統需能辨識此四種標誌，並作適當的處理。換言之，它必須有SGML剖析器(parser)，這種剖析器並不需爲專用之剖析器，只要能執行系統所要剖析之資料即可。

㈡文獻結構定義與標誌

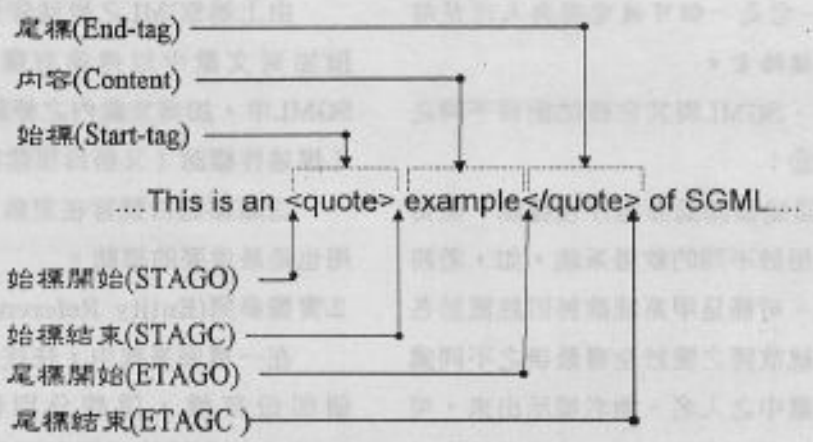
每一類文獻皆有其結構，結構乃由文獻所包含之內容組成。SGML最重要之目的即在界定此文獻結構，其組成內容在SGML中稱爲元素。元素之大小與內涵爲何，使用者可依處理需求定義之，並將元素之位置標示出來，在原始文獻中要標示出各種元素，則需加上標誌符號。各類元素有不同之通用識別語(Generic Indentifier, 簡稱GI)，GI須於DTD中事先宣告才有效，加上通用識別語後之文獻，稱爲SGML文獻。

SGML之標記符號可用來：

1.表現文獻之元素(element)之間的結構關係；

2.定義每一元素之通用識別語及屬性。

標誌所用之符號，SGML設計了參考性具體語法(reference concrete syntax)供參考，如下例所示⑧：




上述〈quote〉〈/quote〉即爲標誌符號，〈quote〉稱爲始標(start-tag)，〈/quote〉稱爲尾標(end-tag)；'〈' 符號表示始標開始(start-tag open)；'〉' 符號表示始標結束(start-tag close)；'〈/' 符號表示尾標開始。這些符號是SGML之 "reference concrete syntax" 所定義的，而符號內之quote則是由使用者決定，始標與尾標之間的資料即爲文獻之元素(element)。標誌符號內所用之元素名稱，如 "quote" 即爲此元素之通用識別語，同一類文獻元素，皆使用相同之GI，且此GI所界定之元素皆需於DTD中宣告，未經宣告之元素，視爲無效。

DTD具體言之，即爲一組標誌宣告，凡是出現在文獻中之標誌元素，有關此元素之屬性(attribute)，及參照實體(entity)皆需在DTD中加以宣告，是以出現在DTD中之宣告以下列三項最爲重要⑨：

1.**元素宣告**(element declaration)：在此定義將出現文每一元素之GI，及其出現之次序，次數等。如：

**〈!Element textbook (front?, body, rear?)〉**

〈!Element...〉表示是一個元素宣告。

textbook是此元素之名稱。

(front,body,rear)表示textbook這一元素包含之內容，且其順序爲front→body→rear。

？表示front,rear爲可有可無之內含。

2.**屬性表宣告**（attribute definition list declaration）：定義元素之屬性及屬性值。如：

```
〈!ATTLIST memo   status     (final | draft)   "final"
                  security   CDATA             #REQUIRED
                  version    NUMBER            "01"
                  sender     NAME              #IMPLIED〉
```

〈!ATTLIST…〉表示這是一個屬性宣告。

memo是元素名稱

status, security, version, sender皆是屬性名稱，表示此屬性宣告乃在定義此元素之文獻狀態，同一元素有數種屬性，可以一起宣告。每一屬性名稱，皆有其對應之屬性值、預設值。

(final | draft)是屬性值，表示此元素之屬性狀態可以是最後之定稿，或只是草稿。

CDATA亦爲屬性值，表示security之屬性以文字表示。

NUMBER亦爲屬性值，表示version之屬性以數字表示。

NAME亦爲屬性值，表示sender之屬性質必須爲正確之SGML名稱。

"final" 是屬性預設值，此值只能在前面之屬性值中二選一。

#REQUIRED表示屬性值一定要在文獻中指定。

#IMPLIED表示此屬性可有可無，若文獻中不指定屬性值，則由應用環境決定。

元素經屬性宣告後，此屬性即可加在原始文獻中，屬性資料在文獻中之表達方式如下：

**〈memo security= "Internal Use" sender= "LTG" 〉**

Internal Use即爲屬性值，它是文字，且需在文獻中指定。

LTG亦爲屬性值，文獻中指定爲LTG。

3.**實體宣告**(entity declaration)：定義可以應用在此文獻之實體。如：

**〈!ENTITY uta "United Typothetae of America" 〉**

〈!ENTITY…〉表示這是一個實體宣告。

uta爲實體名稱。

"United Typothetae of America" 表示uta所要代表之實體內文，換言之，在標誌文獻時，凡碰到 "United Typothetae of America" 之字眼，皆以&uta取代之。

若此文獻須參考其它文獻檔，則需作下列宣告：

**〈!ENTITY part2 SYSTEM "usr.stionx3.textfile" 〉**

實體也可應用於DTD中，作爲具有相同元素內容之取代，如：

〈!ENTITY%h1to4 "h1 | h2 | h3 | h4" 〉

〈!ELEMENT body (P | xmp | %h1to4)+〉

〈!ELEMENT rear (P | %h1to4)+〉

在body, rear元素中，都具有(h1 | h2 | h3 | h4) 之內容宣告，故以%h1to4表示之，而此%h1to4必需作實體宣告。

謝清俊教授等曾以心經科文爲例，建立科文之DTD，並將心經標誌爲SGML文獻，茲以該科文之DTD爲例，說明DTD與SGML文獻之關係。心經之DTD如下⑩：

〈!DOCTYPE心經〉

〈!ENTITY%位置 "起始位置NUMBER#REQUIRED" 〉

〈!ELEMENT心經--(總釋題名？，正釋經文，版本轉換*)〉

〈!ATTLIST心經版本CDATA#REQUIRE)〉

〈!ELEMENT總釋題名--（經題，人題）〉

〈!ELEMENT經題--（綱要，融會各家，五重玄義）〉

〈!ELEMENT（綱要，融會各家）--（#PCDATA）〉

〈!ELEMENT五重玄義--（釋名，顯體，明宗，辨用，判教）〉

〈!ELEMENT釋名--（通名，別名）〉

〈!ELEMENT（通名，別名）--(#PCDATA)〉

〈!ELEMENT顯體--（正顯體，辨異同）〉

〈!ELEMENT（正顯體，辨異同）--(#PCDATA)〉

〈!ELEMENT明宗--（正明宗，辨異同）〉

〈!ELEMENT正明宗--(#PCDATA)〉

〈!ELEMENT辨用--(#PCDATA)〉

〈!ELEMENT判教相--（天台，賢首五教）〉

〈!ELEMENT天台--（五時，四教）〉

〈!ELEMENT五時--(#PCDATA)〉

〈!ELEMENT四教--（化法，化儀）〉

〈!ELEMENT（化法，化儀）--(#PCDATA)〉

〈!ELEMENT賢首五教--（#PCDATA)〉

〈!ELEMENT正釋經文--（序分？，正宗分，流通分？）〉

〈!ELEMENT序分--（通序，別序）〉

〈!ELEMENT（通序，別序）--(#PCDATA)〉

〈!ELEMENT正宗分--（顯說般若，密說般若）〉

〈!ELEMENT流通分--（#PCDATA）〉

〈!ELEMENT顯說般若--（因人顯法，正示法空，顯彰妙果，結贊功能）〉
〈!ELEMENT因人顯法--（能修之人，所修之法，，觀行境界，修證功能）〉
〈!ELEMENT（能修之人，所修之法，觀行境界，修證功能）--(#PCDATA)〉
〈!ATTLIST--（能修之人，所修之法，觀行境界，修證功能）--%位置；〉
〈!ELEMENT正示法空--（明蘊空，顯空德）〉
〈!ELEMENT明蘊空--（#PCDATA)〉
〈!ATTLIST明蘊空--%位置；〉
〈!ELEMENT顯空德--（總標，別標）〉
〈!ELEMENT總標--（#PCDATA)〉
〈!ATTLIST總標--%位置；〉
〈!ELEMENT別標--（三科，十二因緣，四諦，智得）〉
〈!ELEMENT三科--（五蘊，十二處，十八界）〉
〈!ELEMENT（五蘊，十二處，十八界）--（#PCDATA)〉
〈!ATTLIST（五蘊，十二處，十八界）--%位置；〉
〈!ELEMENT（十二因緣，四諦，智得）(#PCDATA)〉
〈!ATTLIST（十二因緣，四諦，智得）--%位置；〉
〈!ELEMENT顯彰妙法--（明菩薩得涅槃，明諸佛得菩堤)〉
〈!ELEMENT明菩薩得涅槃--(#PCDATA)〉
〈!ELEMENT明諸佛得菩堤--(#PCDATA)〉
〈!ELEMENT（明菩薩得涅槃，明諸佛得菩堤）%位置；〉
〈!ELEMENT（結讚功能，密說般若）--(#PCDATA)〉
〈!ELEMENT（結讚功能，密說般若）--%位置；〉

以上是根據心經之完整科文爲結構而定的DTD，科文也者，以圖書館學之觀點而言，即好比分類主題一般。它是所有心經版本可共用的主題結構。每一個版本，不一定皆涵蓋科文所有內容，但決不會超出此結構。事實上，若以文獻之組成成份而言，心經也看成是由一段段之文件組成之文獻，同時可據此結構建立心經的另一個DTD，但是其語意功能不若科文，因爲科文幾已成爲佛家研究經典主要的分析方式。

上述之DTD中，所用到之宣告主要有三種，元素、屬性、實體宣告。每一元素均需宣告其內容，直至最小單位之元素爲止，最小單位元素到底有多小，由使用者決定。元素若有屬性，則需作屬性宣告，否則視若無特定之屬性，由於上述心經DTD中元素之屬性，皆在宣告其起始位置、位置以數字表示（NUMBER）、預設值爲REQUIRED（表示此屬性在文獻中必須指定），是以由一實體宣告來取代，以免重複書寫。

在此DTD中，也可以視若它乃根據科文，將心經建爲一樹狀結構，每一元素可以看成是此樹狀結構的節點，而最底層無法再細分之節點，所包含的只有原始資料。這種樹狀結構事實上也就是超文

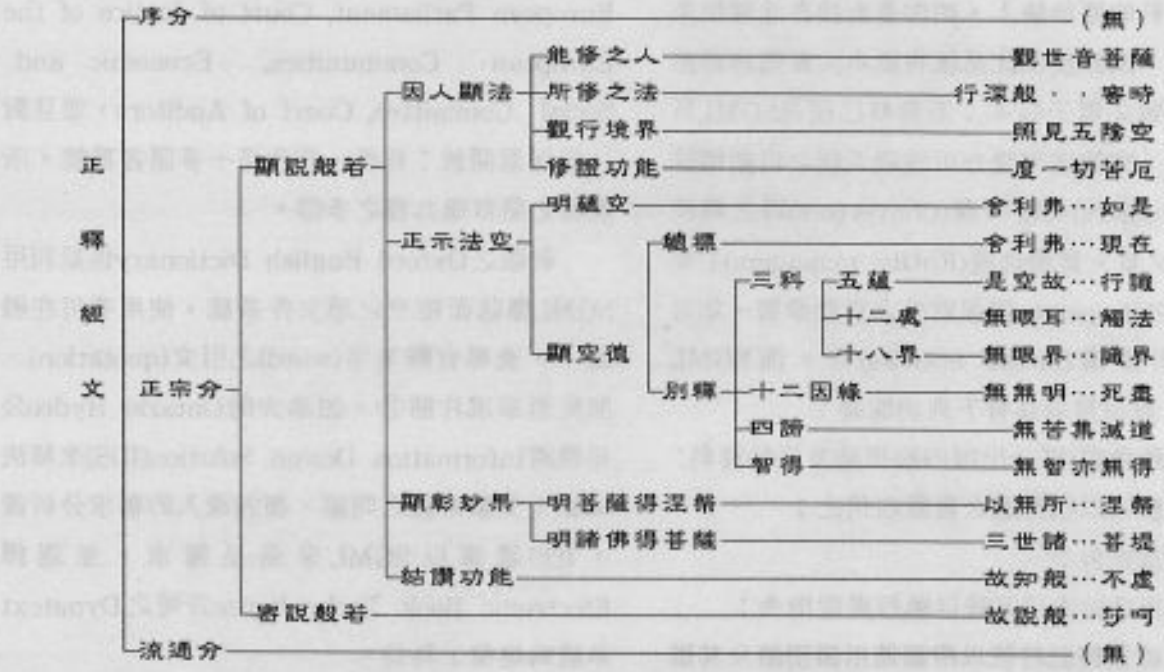
件之結構，DTD之最小元素單元，就是超文件的終端節點。

將上述宣告過之元素名稱及屬性加到心經原文後，即成下例：

## 經標誌後之鳩摩羅什版心經

〈心經 版本＝"鳩摩羅什"〉〈正釋經文〉〈顯說般若〉〈因人顯法〉〈能修之人 起始位置＝"1"〉觀世音菩薩。〈/能修之人〉〈所修之法 起始位置＝"6"〉行深般若波羅蜜時。〈/所修之法〉〈觀行境界 起始位置＝"14"〉照見五陰空。〈/觀行境界〉〈修證功能 起始位置＝"19"〉度一切苦厄。〈/修證功能〈/因人顯法〉〈正示法空〉〈明蘊空 起始位置＝"24"〉舍利弗。色空故。無惱壞相。受空故。無受相。想空故。無知相。行空故。無作相。識空故。無覺相。何以故。舍利弗。非色異空。非空異色。色即是空。空即是色。受・想・行・識亦複如是。〈/明蘊空〉〈顯空德〉〈總標 起始位置＝"88"〉舍利弗。是諸法空相。不生。不滅。不垢。不淨。不增。不減。是空法非過去。非未來。非現在。〈/總標〉〈別釋〉〈三科〉〈五蘊 起始位置＝"120"〉是故空中無色。無受・想・行・識。〈/五蘊〉〈十二處 起始位置＝"131"〉無眼・耳・鼻・舌・身・意。無色・聲・香・味・觸・法。〈/十二處〉〈十八界 起始位置＝"145"〉無眼界。乃至無意識界。〈/十八界〉〈/三科〉〈十二因緣 起始位置＝"154"〉無無明。亦無無明盡。乃至無老死。亦無老死盡。〈/十二因緣〉〈四諦 起始位置＝"172"〉無苦・集・滅・道。〈/四諦〉〈智得 起始位置＝"177"〉無智。亦無得。〈/智得〉〈/別釋〉〈/顯空德〉〈/正示法空〉〈彰顯妙果〉〈明菩薩得涅槃起始位置＝"182"〉以無所得故。菩薩依般若波羅蜜故。心無罣礙。無罣礙故。無有恐怖。遠離一切顛倒夢想苦惱。究竟涅槃。〈/明菩薩得涅槃〉〈明諸佛得菩堤 起始位置＝"222"〉三世諸佛。依般若波羅蜜故。得阿耨多羅三藐三菩堤。〈/明諸佛得菩堤〉〈顯彰妙果〉〈結讚功能 起始位置＝"243"〉故知般若波羅蜜。是大明咒。是無上明咒。是無戰等明咒。能除一切苦。眞實不虛。〈/結讚功能〉〈/顯說般若〉〈密說般若 起始位置＝"274"〉故說般若波羅蜜咒。即說咒曰。揭帝。揭帝。波羅揭帝。波羅僧揭帝。菩薩僧莎呵。〈/密說般若〉〈/正釋經文〉

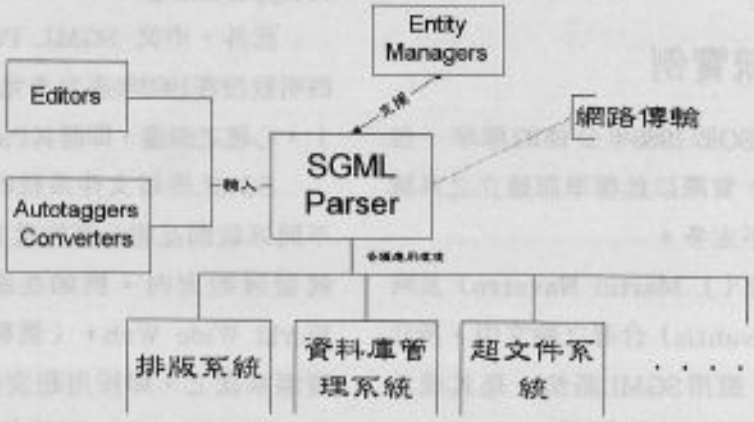
上述標誌後之心經中，在〈元素名稱〉〈/元素名稱〉之標記符號內之元素名稱即所謂之GI，加上之GI都已在DTD中宣告過，但在DTD中宣告過之元素，正文中未必皆包括，鳩摩羅什版之正文只有正釋經文中之正宗分；元素名稱後可加上已宣告過之屬性。標誌文獻之結構與內容順序與DTD是相對應的，若以樹狀圖表示，則心經之科文（亦爲DTD中之元素名稱，或加在文獻中之GI)與心經原文，可以對應如下圖：




圖四　心經科文與原文對照圖

## 四、SGML系統模式

SGML只是一套描述文件結構的超語言(metalanguage)，若要應用之，則須建立SGML系統來處理標誌、剖析、核對等相關步驟。一個SGML系統主要之模式可以下圖表示⑪：




圖五　SGML系統模式圖

在上述SGML系統模式圖中，Editors系統主要負責資料的原始輸入，如作者直接在此編輯系統上創作，或直接在此系統將紙本文獻轉換爲加了標誌符號之電子版本；若資料已在非SGML系統上建立，傳輸進來後亦可透過系統之自動標誌器(Autotaggers)或轉換器(Convertors)將之轉換爲SGML文獻。實體管理(Entity managers)主要在支援SGML parser管理對外之實體參照，如有那些公用實體(public entities)等。而SGML parser一般而言須具有下列功能⑫：

1.掃描文件內容，區分出這四種標誌及原始資料；
2.將實體參照以相對應之實體取代之；
3.編譯標誌宣告；
4.送控制指令給處理系統以執行處理指令；
5.編譯描述性標記符號以辨識通用識別語及其屬性，並根據文獻類型規則執行下列動作：(1)檢查每一GI及其屬性是否正確；(2)探知其在文獻結構中之位置。
6.送控制指令給處理系統以執行和GI相連結之程序(procedure)。

SGML文獻剖析後，可做各種利用，如用在排版系統、資料庫管理系統、超文件系統等等，若要作交換，亦可透過Parser中之標準包裝（採SDIF標準）送出去。

## 五、SGML系統實例

雖然SGML是ISO於1986年公佈的標準，但至今仍在推廣階段。實際以此標準而建立之系統，見諸於文獻者並不太多。

1991年，雷瓦若（J. Martin Navarro）及阿里方廷（P. E. Alevantis）合著之論文中，所介紹的CELEX系統，使用SGML語法，是其成功的關鍵。該系統爲歐洲共同體（European Community）所擁有，它是一個跨機構系統，使用的單位包括：Commission of the European Communities, Council of Ministers, European Parliament, Court of Justice of the European Communities, Economic and Social Committee, Court of Auditors，並且對一般民眾開放；此外，它也是一多語言系統，所處理之語言達九種之多⑬。

新版之Oxford English Dictionary也是利用SGML標誌而建立之超文件系統，使用者可在視窗下，查尋有關某字(word)之引文(quotation)、歷史沿革或片語⑭。加拿大的Ontario Hydro公司聘請Information Design Solution(IDS)來解決其線上文獻系統之問題，經過深入的需求分析後，IDS建議以SGML來滿足需求，並選擇Electronic Book Technologies公司之Dynatext系統爲建檔工具⑮。

另一使用SGML之實例，是Silicon Graphics之IRIS InSight系統，這是第一個由電腦代理商的觀點來看線上文獻系統，且首先採用SGML者，Glushko在論文中主要由四個階段來檢討此系統⑯。

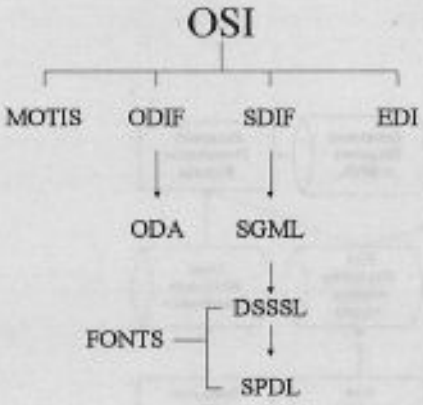
國內在這方面的研究，則尚處於起步階段，元智工學院電機與資訊工程研究所周亞民之碩士論文，乃使用SGML與物件導向資料庫轉換古書爲Hypertext⑰。

此外，中文SGML Parser由元智工學院朱四明教授在1992年底宣告完成第一版（現至第二版），心經之表達，即經其Parser解析、除錯過。

SGML與超文件系統的結合，是超文件可在不同系統間互相分享的重要依據，也是超文件系統發展的方向，例如在網際網路Internet上之World Wide Web，（簡稱WWW或W3），網路資源系統上，即採用超文件標誌語言Hypertext Markup Language(HTML)標誌資料，Internet上另一可處理視聽、影像、全文之Mosaic系統，其文件也是以SGML爲標誌之標準。

## 六、SGML相關標準與計畫

SGML並非一單獨的標準，而是在國際標準組織（International Standard Organization）所制定之開放性系統互連（Open System Interconnection，簡稱OSI）環境中，由文獻之原始創作至最後產品傳輸之完整文獻處理過程中的一環而已，SGML在這套標準中的角色，主要在建立基本資料，它需要配合上其它的標準，才能完成整個出版過程，相關標準可以下圖表示之：




圖六　OSI有關文獻處理之相關標準

| | |
|---|---|
| OSI | Open System Interconnection |
| MOTIS | Message Oriented Text Interchange System |
| ODIF | Office Document Interchange Format |
| ODA | Office Document Architecture |
| SDIF | SGML Document Interchange Format |
| SGML | Standard Generalized Markup Language |
| FONTS | Font and Character Information Interchange |
| DSSSL | Document Style Semantics & Specification Language |
| SPDL | Standard Page Description Language |
| EDI | Electronic Data Interchange |

以下即就SGML這一系列相關標準說明如下：

㈠相關標準

1.SGML文件交換格式標準(SDIF)

SDIF是SGML Document Interchange Format(ISO-9096)之簡稱，它是將SGML文獻加以封包（package）成在OSI（Open System Interconnection）環境中交換的標準。

由於SGML語言最大特徵乃在建立實體結構(entity structure)，換言之，SGML不需將完整的文獻存放成單一檔案或資料結構，其圖形檔與文字檔通常會分開，文字檔也可分成數個部份；而SDIF提供機制以匯集同一文獻之實體，將它們組合成適合OSI交換的單一資料結構，接收端收到後也可根據SDIF在將之分解爲實體結構。

和OSI其它元件一樣，SDIF也是抽象語法標記（Abstract Syntax Notation, ASN.1, ISO-8824, 及ISO-8825）二元編碼（binary encoding）標準的應用⑱。

2.文件格局，語意及規範語言(DSSSL)

SGML設計之初，設計者雖已建立起文獻之內容與形式(form)須分開的共識，但卻認爲他們也應處理有關形式之問題。因此，同時擬定一些排版語言標準。但當SGML完成後，設計者又體認到，市面上的排版系統已經夠多了，不需要再有新的排版語言，目前應該做的，是如何將SGML的非程序性標誌，擴充爲處理環境所須之描述。同時，他們也認爲，所謂的處理並不僅於排版系統，也包括其它功能，如資料庫檢索等。

DSSSL(Document Style Semantics and

Specification Language，DIS 10179)就是以SGML爲基礎，描述SGML文獻之處理的標準語言。DSSSL主要由兩個模式組成：location model及general language transformation process，前者在建立SGML元素中一段資料之絕對位置與相對位置；後者在連結前面建立起位置之項目，描述其處理方式⑲。

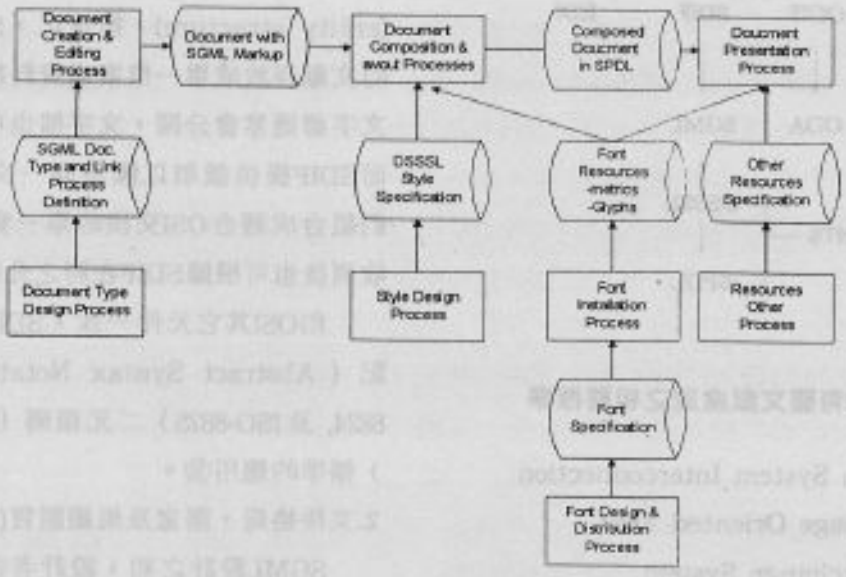
3.標準版面描述語言(SPDL)

雷射印表機的發展，使得文件及圖形資料可以較低的成本印刷。這種發展也引起學者專家們發展標準印表機介面之興趣。此列表機介面後來即成爲「版面描述語言」（Page Description Languages，簡稱PDL's）。

後來國際標準組織承繼原來SPL的工作，將之發展成爲一國際標準，稱爲「標準版面描述語言」(Standard Page Description Language，簡稱SPDL)。

SPDL定義一套不因設備而異(Device-independent)的語言，以描述電子文件，如黑、白、灰或彩色文件，影像，圖形等資料，使其適於呈現在印表機、螢幕或任何媒體上。

作過SGML標誌的文件，很容易的就可以交由DSSSL、SPDL處理輸出工作，SGML、DSSSL、SPDL三者間之關係及處理流程，可由下圖說明之⑳：



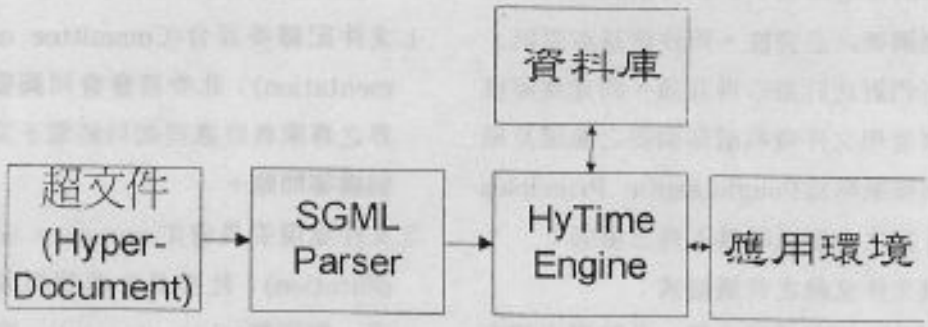

圖七　SGML，DSSSL，SPDL關係流程圖

4.超媒體同步描述語言(HyTime)

HyTime全名爲Hypermedia/Time-Based Structuring Language (ISO/IEC 10744:1992)。Hytime源起於SGML應用於音樂資料之描述及處理計畫，此計畫稱爲Standard Music Description Language，簡稱SMDL(cd 10743)。此計畫進行中，當設計者檢討如何連結樂譜、錄音、及演奏之評論時，突然領悟到，這也就是描述Hypertext的機制，同時，他們也由分析音樂中時間的概念，因此了解任何可以計量之事物的暫時性關聯，所以，這些設計者決定延緩音樂計畫，先發展較通用的語言，HyTime因而誕生㉑。

HyTime是表達多媒體(multimedia)、超文件、超媒體(Hyper-media)、時間(time-based)、空間(space-based)等文獻之標準語言。它使得超文獻(hyperdocument)不需具有標準的多媒體物件(objects)、標記方式(notations)、修飾語(modifiers)、或連結類型語意，即可交互作業(interoperability)。

HyTime和DSSSL相同的是，它們都是以SGML爲基礎，描述文獻處理的工具；不同之處在於，HyTime可應用於非SGML文獻，只要它的核心文獻是SGML文獻即可。要了解SGML與HyTime的關係，可以超文獻處理模式圖說明如下㉒：




圖八：SGML與HyTime之關係圖

SGML Parser之功能已如前述，而Time engine之功能主要有二㉓：

(1)將SGML Parser處理完之文獻傳達給應用環境；

(2)根據由Parser端接收到之資訊，建立引擎內部資料結構。

5.Registration

如前所述，SGML文獻可參考到公用實體，而幾乎所有的SGML建構（如其宣告、文獻內容）也都可定義爲公用實體，供他人參考。此公用實體定義之標準即：Registration Procedures for Public Text Owner Identifiers(ISO-9070)。此標準可以定義文件的「擁有者」，此擁有者可爲個人、團體機關、或只是個電腦系統。本標準採用ISBN（Internation Standard Book Number）之方法，由各國自行登錄，以建立起國際之登錄權威檔；同時，它也提供本標準之登錄號與ASN.1之物件識別號（Object Identifiers）相對應之方法，因爲物件識別號經常應用於OSI之應用環境。此外，本標準亦可擴充爲一種通用之登錄標準，並不僅限於SGML文件及其宣告之使用㉔。

6.HTML

在Internet網路上，World Wide Web開啓連結全球相關資訊之先聲。HTML(Hypertext Markup Language)即在提供連結資訊一簡單格式。HTML以SGML觀點定義並連結文獻，它代表一種文獻類型(即超文件)，也是表達這種文獻類型之實體的標誌語言。Internet網路上所有W3相容之系統都必須能處理HTML㉕。

㈡相關計畫

TEI（Text Encoding Initiative）是一個由電腦與人文學會（Association for Computers and Humanities，簡稱ACH）、計算語言學會（Association for Computational Linguistics，簡稱ACL）、文學與語言計算學會(Association

for the Literary and Linguistic Computing，簡稱ALLC) 聯合贊助的國際計畫，其主要之工作乃在發展並傳播研究者使用之機讀文件交換綱要，以及建議新文件之編碼方式。

TEI之成立源起於ACH在1987年12月12至13日於紐約Vassar College, Poughkeepsie所舉行的會議，此次會議中有31位來自於北美、歐洲、以色列、日本的學者專家，他們討論到建立一套機讀文件編碼綱要之必要性、彈性與基本原則，會議中，學者們對此計畫取得共識，同意遵照幾個基本原則來管理文件資料編碼綱要之範圍及組織，這些原則後來稱爲Poughkeepsie Principles。並界定此計畫之主要目的爲下列三項㉖：

1.提出一機讀文件交換之共通格式；
2.提供一套新文件資料編碼建議，此建議主要在界定文件的那些特徵須要編碼，及這特徵應如何呈現；
3.考證既存的編碼系統，並發展一種超語言(metalanguage)以描述之；

隨後，該計畫並決定以SGML爲描述文件之基本語法，SGML本身並非編碼系統，而只是一個架構，可依此架構發展編碼系統。

TEI即遵照這些原則發展電子文件編碼與交換綱要（Guidelines for Electronic Text Encoding and Interchange）。此綱要須能滿足下列設計目標㉗：

1.足以呈現研究所須之文件特徵；
2.此綱要必須簡單、清晰、具體；
3.不須任何特殊目的之軟體，即能易於爲研究者使用；
4.可作嚴格之定義，並使文件能被有效的處理；
5.可讓使用者自行擴充；
6.符合現存的標準。

此外，綱要中需包含下列明顯之建議㉘：

1.新文件之編碼方式（文件特徵需掌握，最低限度應說明文件應如何標明）；
2.如何增加或更正資訊到已編之碼中；
3.已編碼者如何交換；
4.編碼之檔案文獻；
5.爲書目控制之文件及編碼文獻；
6.建議之標誌系統文獻。

目前TEI有四個工作委員會，分別是㉙：

1.文件記錄委員會(Committee on Text Documentation)：此委員會會同圖書館和檔案管理界之專家負責處理如何將電子文件之編目資料編碼等問題。
2.文件呈現委員會(Committee on Text Representation)：此委員會處理機讀形式之呈現問題，如字集(character set)、文件之邏輯結構、原始資料之版式及其它外在特徵等。
3.文件分析及編譯委員會(Committee on Text Analysis Interpretation)：此委員會主要須提供適合特定學科用來分析文件處理程序之標記符號(tags)。
4.超語言及語法問題委員會(Committee on Metalanguage and Syntax Issues)：此委員會及制式語言(formal language)專家很快即認爲以SGML作爲TEI之編碼語言是最好的選擇，並出版如何使用SGML指南，及爲配合TEI，應對SGML作那些修訂之建議，同時指出TEI與其他編碼系統間轉換時將遭遇之問題。

## 七、圖書館界了解SGML等標準之意義

圖書館界對於電腦的應用有長久的歷史，但主要以處理書目資料爲其最初之目標，並且爲此建立了很多相關標準，但目前電腦技術與應用的進步與普及，使得邁進圖書館電腦系統之資訊，除了書目資料外，尚有很多全文資料，其傳輸形

式除離線方式外，網路上的擷取更爲便利。換言之，傳統圖書館自動化必須漸漸的由書目資料的處理擴展到文獻的處理(document processing)，所以圖書館界也必須對文獻處理之相關標準有所了解，甚至會利用它來編寫交換文件，例如目前網路上很熱門的Mosaic系統，其文件就是以HTML來編寫，這是SGML的部份應用(subset)，圖書館員若要利用Mosaic系統就必須了解SGML，HTML等標準，本文只是對此標準作了初淺的介紹，希望藉此拋磚引玉，在資訊科技瞬息萬變之際，與同界共勉，更希望國內相關行業及圖書館界也能效法國際TEI計畫，針對中文各類文獻之特性擬定相關標誌語言。

（收稿日期：1994年12月10日）

## 註　釋

註①：Charles F. Goldfarb, The SGML Handbook (Oxford: Clarendon Press, 1990), p.7.

註②：同上註，pp.7-8。

註③：Gerard Salton, Automatic Text Processing (Reading, Massachusetts: Addison-Wesley,1989), p.530.

註④：Arno J. N. M. Peels, Norbert J. M. Janssen, and Wop Nawijn, "Document Architecture and Text Formatting", ACM Transactions on Office Information Systems, v.3(Oct. 1985), pp.347-369.

註⑤：William K. Horton, Designing and Writing Online Documentation: Hept Files to Hypertext (New York: John Wiley, 1990).

註⑥：C. M. Sperberg-McQueen and Lou Burnard eds."A Gentle Introduction to SGML", In Guidelines For Electronic Text Encoding and Interchange, chapter 2, Draft Version 2 (Chicago Text Encoding Initiative, 1993), pp.4-5.

註⑦：同註①，p.21。

註⑧：同上註，p.24。

註⑨：同上註，p.21。

註⑩：謝清俊等著，「電子佛典中處理中文版本的方法」，The 5th CJK DocP Meeting (Taiwan, Taipei, May 23-24, 1994).

註⑪：David James Mason, "Outside Industry Research: Standards Backgrounds, Implementations, and Products, with Emphasis on SGML", Document Interchange Symposium Proceedings Appendices D-F (25-26 March, Airlie, Virginia), pp.32-45.

註⑫：同註①，p.22。

註⑬：J. Marin-Navarro and P.E. Alevantis, "Alice in the Wonderland of SGML: Streamlining Text Entry in the CELEX Databases", The Electronic Library, v.9 (June 1991), pp.155-159.

註⑭：Heather Fawcett, "The New Oxford English Dictionary Project", ACM Technical Communication, third quarter (1993), pp.379-382.

註⑮：Ann Rockley, "Ontaria Hydro and SGML", ACM Technical Communication, third quarter (1993),

pp.383-386.

註⑯：Robert J. Glushko, 'Silicon Graphics' IRIS InSight: An SGML Success Story', ACM Technical Communication, third quarter (1993), pp.394-402.

註⑰：周亞民，「使用SGML與物件導向資料庫轉換古書爲Hypertext」（碩士論文，私立元智工學院電機與資訊工程研究所，民82年6月），頁140。

註⑱：International Organization for Standardization, Information Processing-SGML Support Facilities-SGML Document Interchange Format (SDIF) (ISO, 1988).

註⑲：International Organization for Standardization, Information Technology-Text and Office Systems-Document Style Semantics and Specification Language (DSSSL) (ISO, 1991).

註⑳：蘇克毅，「電子文件描述與處理語言之國際標準簡介」，文件處理標準研討會，台北，民國82年4月12日。

註㉑：International Organization for Standardization, Information Technology-Hypermedia／Time-based Structuring Language (HyTime) (ISO, 1991).

註㉒：Steven R. Newcomb, Neill A. Kipp, and Victoria T. Newcomb, 'Hytime :Hypermedia／Time-based Document Structuring Languge', Communications of the ACM, v.34 (Nov. 1991), p.80.

註㉓：同上註，pp.79-80。

註㉔：同註⑪，p.13。

註㉕：Tim Berners-Lee, 'Hypertext Transfer Protocol: A Stateless Search, Retrieve and Manipulation Protocol', Internet Draft, (Expries 5 May, 1994) subscribe www-talk-request@info.cern.ch.

註㉖：Susan Hockey, The ACH-ACL-ALLC Text Encoding Initiative: An Overview (June 1991, Rev, Feb. 28, 1993), Document no. TEI J16.

註㉗：同上註，p.6。

註㉘：同上註，p.7。

註㉙：同上註，pp.3-4。