

第四章 系統分析與實作

本章將介紹本論文提出之網路書籍模式理論中書籍編輯階段所需的兩個工具—書頁下載器與書籍編輯器之實作細節與模組分析，並提出建置符合 SCORM 規範之網路書籍模式環境的相關需求。第一節簡介個別工具的功能以及系統發展的環境；第二節進一步分析各項工具之模組及運作方式；第三節則以實際建立範例書籍的過程來呈現整個模式的運作過程。

第一節 系統規劃

本節分為兩個部分說明本論文之相關系統工具規劃細節，第一部份以網路書籍的製作流程為主軸探討個別階段中所需的工具並簡介其功能，第二部分則說明所提及的各個相關工具和系統模組架構。本論文所提出的書籍模式編輯階段理論中，將網路書籍作者的編輯任務分為書頁下載、教材元件編製、課件打包以及書本上線四個階段，各階段關的系統工具分別是書頁下載器、元資料編輯工具、書籍編輯器以及 SCORM 網路書籍伺服器。各工具與流程的對應關係我們將在下文中繼續進行細部討論。

4.1.1 網路書籍製作

針對書頁下載與課件打包，我們提供兩個工具程式讓網路書籍的作者使用，分別是書頁下載器與書籍編輯器，以下簡單描述兩項工具之外觀與功能規劃：

A. 書頁下載器規劃

作者在決定好所需要的教材內容後，將網址輸入書頁下載器開始進行網頁下載，本研究採行的策略是以維持單一網頁之完整性為主要考量下載相關檔案，將彼此之間的位址關係由絕對參照改為相對參照，並移除不必要的標籤，例如：`<script><style>`等，最後儲存於作者的電腦中，另外也提供加入標籤的功能以滿足作者在書頁中置入隱藏`<term>`標籤的需要。作者在網址列輸入所欲下載的網頁 URL 以後，可以在中間的預覽區看見目標位址的網頁呈現狀況，作者選定所要存入書頁的關鍵字後點選“加入 term 標籤”按鈕，系統會將作者鍵入的關鍵字暫存起來，等到作者完成所有的動作後按下儲存按鈕，書頁下載器會開始執行下載的動作，並在檔案寫入作者電腦硬碟前，把先前暫存的關鍵字以符合`<term>`標籤規格的方式寫入下載的網頁中，再將該網頁的來源位址寫成`<term tag="連結" description="網址">`附加到儲存的網頁中，完成書頁的儲存後則依照前述的策略將完成整個書頁所需下載的其他檔案一併下載至作者電腦中，然後將個別的連結參照全部改為相對位址。

圖 4-1 為書頁下載器之使用畫面，畫面來源為行政院研究發展考核委員會無障礙網路空間服務網【6】所提供之符合網路無障礙規範的範例網頁。



圖 4-1 書頁下載器執行畫面

B. 書籍編輯器規劃

本研究提供的第二個軟體工具為書籍編輯器，為了符合 SCORM 的課程包裹規範，書籍編輯器應與清單檔 `imsmanifest.xml` 的編輯程式緊密結合，且由於兩者的型態和概念相同，所以我們在實作時便以能夠編寫出清單檔邏輯架構且可以進一步對實體資源加以描述並將其含括入課程包裹中為主要的發展目標。本研究採用由英國愛丁堡大學發展的 IMS Editor【23】為基礎，借用其所提供的清單編輯功能來滿足作者編輯網路書籍邏輯架構時的需求，但由於我們的書頁在被下載之後裡面會包含 `<term>` 標籤所記錄的若干隱藏訊息，所以為了符合本研究所提出之理論，我們對該軟體進行部分修改的動

作，作法為在編輯器中加入兩個模組。第一個模組是簡單的剖析器(Parser)，主要的工作在於負責監聽(Listen)作者的編輯動作，當作者將一個新的書頁或者教學元件新增到清單，並將邏輯單元對應到某個資源檔案實體時，Parser 會對加入的書頁進行剖析，找出該份文件中所含的<term>標籤，取出標籤中所隱含的訊息，然後加到 ToT 表格暫存於記憶體中。另一個加入的模組則是一個 HTML 文件的寫入器(Writer)，在作者確定編輯好網路書籍的動作並且將所有的實體資源與清單檔一起包裹起來以後，Writer 會去讀取 ToT 所包含的訊息，並將訊息分成兩類，一類是用來編寫索引頁的內容，另一類則是用來編寫外部參考頁面的相關網址，Writer 分別將兩個類別的訊息寫成索引頁和外部參考頁，並呼叫相關的 API，將兩個檔案一併存入書籍包裹中，至此作者編輯的工作已告完成，最後只要將書本上傳至線上教材(書籍)管理系統就可以開始提供給讀者閱讀了。圖 4-2 為書籍編輯器執行畫面。

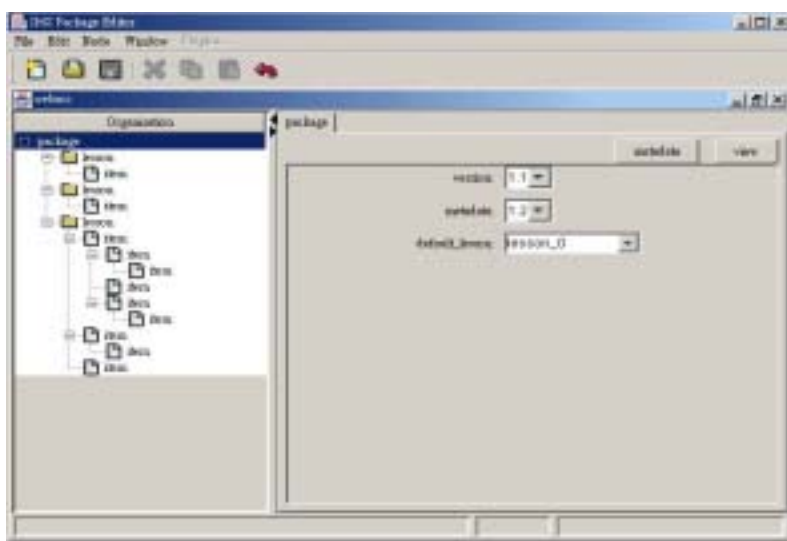


圖 4-2 書籍編輯器執行畫面

C. 教材元資料之編製

另外，如前一章所提到的課程元件的邊製的階段除了所需資源的實體檔案外，所需要的工作便是為各個檔案撰寫個別的 Meta Data 描述檔，不同的課程元件整合程度所需要用到的 XML 標籤雖然不盡相同，但依據 SCORM Content Aggregation Model 【17】 規範的內容所規定的，每一種類的課程整合仍有其個別所需強制使用的相關標籤，以使得該課程元件可以符合 SCORM 規範。我們將 SCORM CAM 標籤的使用時機詳細列表置於本論文的附錄一，表中的 M 代表強制規定使用(Mandatory)；O 則代表非強制(Optional)；R 表示該標籤目前為 ADL 保留(Reserve)，留待後續標準的訂定後才開放給使用者的標籤。

在一個學習資源的 Meta Data 描述檔中使用規範中所有的標籤本是增進該資源自我描述性最佳的方式，但是由於各類資源的本身性質並不相同，某些欄位的資訊並非完全必要，所以目前對於規範中所訂定的 XML 標籤使用方面，本論文建議至少要將符合該整合階層(Aggregation Level)的所有 Mandatory 標籤相關訊息填入 Meta Data 檔案中，對於該資源後續的重用和分享可以有較大的幫助。編寫 XML 檔案的方法很多，可以透過純文字編輯器手動輸入或者使用其他廠商所開發出來的軟體工具來編寫，本研究基於尊重作者的選擇自由之立場，故並沒有實作出類似的軟體提供給書籍作者使用，作者可以依照自己的個人偏好選擇適合的工具來為其所要放進元件庫的元件編寫 Meta Data，而不需受限於本研究提供之工具。

D. 書籍伺服器

網路書籍系統中關於書籍伺服器部分，本研究直接使用 ADL 所提供的範例執行平台(Sample Run Time Environment, Sample RTE)來擔任書籍伺服器的角色。作者完成了書籍編輯階段的前幾個步驟後，只需將編輯完成的網路書籍上傳至 Sample RTE 所在的伺服器，作者便完成了編輯的工作。上傳後的網路書籍由 RTE 伺服器的管理者將該網路書籍匯入(Import)系統中，然後開放給讀者點選閱讀。讀者只需要連線至同一伺服器上，登入後選擇想要閱讀的相關書籍並向系統註冊選擇該書籍，接著就可以開始進行閱讀學習活動了。

4.1.2 系統架構

本節所討論之各個相關軟體工具的平台如下：書頁下載器和書籍編輯器是用 JAVA 語言編寫完成，基於 JAVA 語言跨平台的特性，可在各種支援的平台上直接執行，故不限定使用的作業系統；而 ADL Sample RTE 則是安裝在 Windows 2000 的作業系統中，以 Tomcat 為主要的伺服器軟體，內部的網頁是由 JSP 與 JAVA Servlet 的技術寫成，API Adaptor 則是由 JAVA Applet 技術來編寫。Sample RTE 中另外附上一個簡單的資料庫，是由 Microsoft

Access 軟體實作。Meta Data 的編寫軟體視個別網路書籍作者個人的選擇而定，在此不多加限制。整體系統中的各模組關係如圖 4-3 所示。

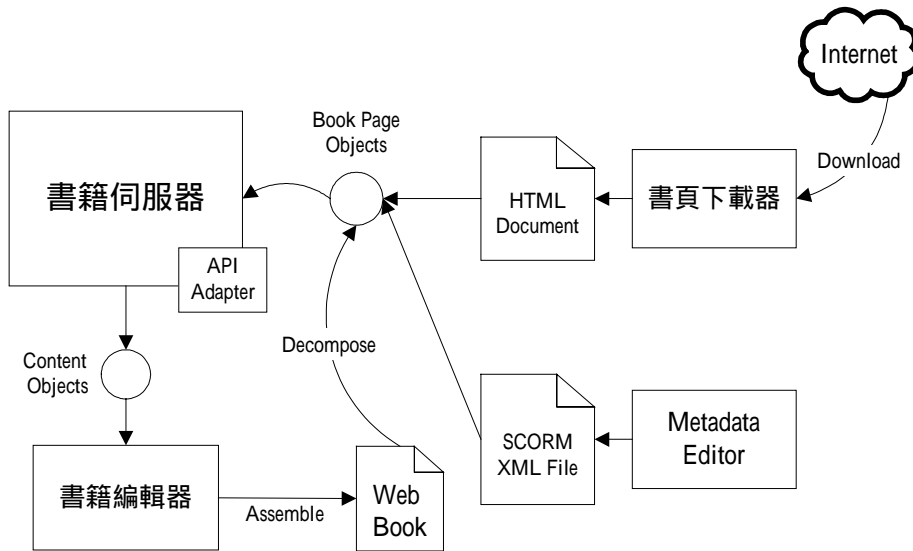


圖 4-3 SCORM 網路書籍系統模組架構關係圖

第二節 系統模組分析

本節將探討 SCORM 網路書籍模式的相關系統模組。我們將系統分為兩大部分：書籍編輯模組和書籍伺服模組兩大類，而使用者角色方面則分為作者、讀者和管理者三種，圖 4-4 為系統的細部概況和各類型使用者與 SCORM 網路書籍系統間的互動關係示意圖。

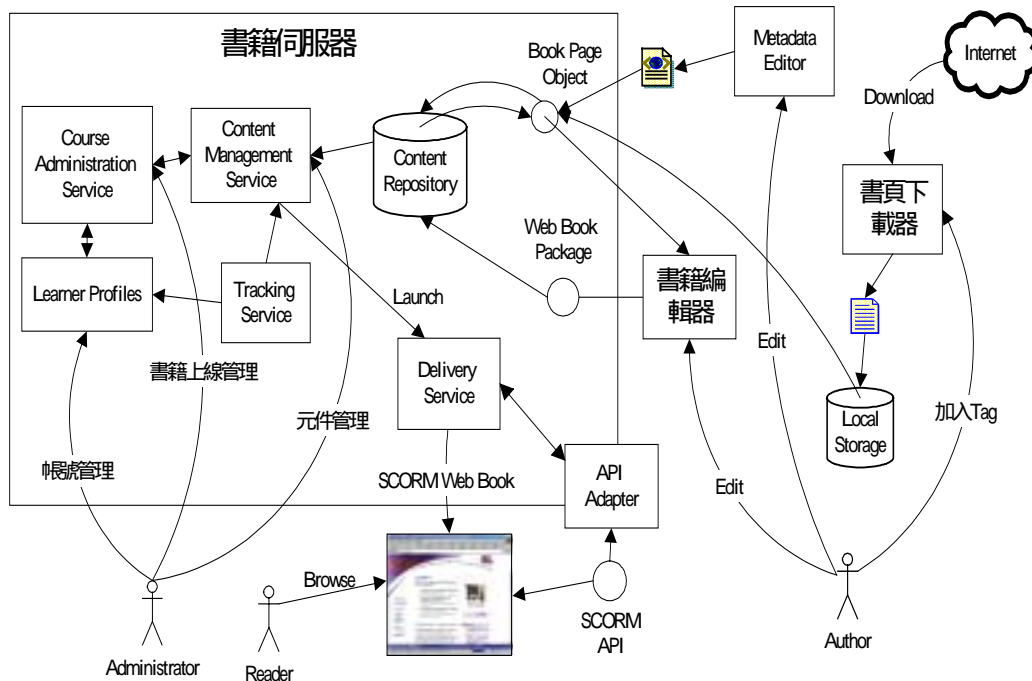


圖 4-4 SCORM 網路書籍系統細部架構圖

以下將以書籍編輯和書籍伺服器兩大相關模組為主軸進一步分析探討其個別運作方式以及與不同角色使用者間的互動關係。

4.2.1 書籍編輯模組

根據 SCORM 網路書籍模式理論，本研究將網路書籍編輯階段可分為書頁下載、課程元件編製、課件打包和書本上線等四個部分，針對書頁下載的部分我們提供的工具軟體為書頁下載器，書籍編輯器主要是幫助完成課件打包的工作，而課程元件編製所需的 Metadata 編輯工具為一般文字編輯器，最後作者再將完成編輯的書籍透過 FTP 上傳至伺服器中。書頁下載器、

Metadata 編輯工具和書籍編輯器合稱為書籍編輯模組，主要的使用角色為網路書籍的作者，作者的身份並不限定特定的使用者才可以擔任，只要是透過編輯模組完成網路書籍寫作並將成品上傳後與其他使用者分享的人皆可視為網路書籍的作者。書頁下載器和書籍編輯器都是以 JAVA 語言編寫而成，由於 JAVA 語言具有跨平台特性，故而實作出來的工具理論上可以在任何支援 JAVA 執行環境的系統平台上使用，以下針對書頁下載器和書籍編輯器等兩個子模組進行個別分析。

A. 書頁下載器子模組

本論文所提供的書頁下載器使用 JAVA 語言所提供的基礎物件(JAVA Foundation Class, JFC)編寫而成，圖 4-4 為書頁下載器的外觀架構圖。



圖 4-5 書頁下載器外觀架構圖

以下針對書頁下載器各區的使用介面及功能作進一步的說明：

1. 功能表下拉式選單：

本區包含的功能選項分別為 File、Go 以及 Themes 三個部分。

File 下拉式選單中，使用者可以選擇開啟新視窗(New Window)、開啟舊檔(Open)、列印文件(Print)、關閉視窗(Close)以及結束程式(Exit)等功能。Go 選項內的功能與瀏覽控制區的四個 Button 相同，Themes 則提供使用者選擇不同的視窗外觀。

2. 瀏覽控制區：

本區含有四個 Button，提供使用者類似瀏覽器的向前(Forward)、向後(Back)、重整(Reload)和回到首頁(Home)的功能，在作者預覽所要下載的網頁時，提供類似於瀏覽器的操作情境。首頁預設為本實驗室的首頁(<http://xpplab.ice.ntnu.edu.tw>)。

3. 網址輸入區：

本區為作者輸入所要下載網頁位址的欄位，如作者點選該網頁中的某個連結後，則改以顯示連結後的位址。經系統判定輸入的為合法 URL 後便可在預覽區看見目標網頁的輸出概況。

4. 網頁預覽區：

本區可以讓作者確認是否輸入的網址為所要下載的網頁，以 JAVA Swing 中的 JeditorPane 元件製成，由於該元件的 HTML 支援版

本以 HTML 3.2 為主，較新的語法並不能保證完全正確的顯示，故本論文將此區定義為預覽性質，作者在下載網頁前，還是要先透過一般瀏覽器確認目標畫面的正確性後再使用書頁下載器執行下載動作。

5. 加入 term 標籤：

在課件打包階段中，本研究提出書籍自動索引機制，在下載的網頁中置入隱藏的<term>標籤供書籍編輯器擷取作為自動產生索引頁和外部連結頁兩種書頁的來源。在書頁下載器視窗中點選此按鈕後將彈出對話框，讓作者選擇所要加入網頁的標籤術語類型以及選定的關鍵字，收集完成的相關資訊則以 JAVA Container 中的 Vector 暫存之。

6. 儲存：

作者點按此按鈕後，便開始執行網頁下載的動作，下載的任務主要是透過一個稱之為 PageSaver 的非視覺化物件完成之。PageSaver 繼承自 JAVA 基礎物件中的 HTMLToolkit，主要的工作方式首先為目標 URL 開啟一個輸出串流(Output Stream)，然後呼叫 JAVA 提供的 HTML 剖析器(Parser)分析目標網頁所含的內容，將網頁中的<script>及<style>標籤移除；對於等與該頁顯示有直接相關的標籤區段則進一步讀取該 tag 的屬性欄位(Attribute)，取得該標籤所連結的檔案之後產生一個新的線程(Thread)將該檔案下載至 Local 端，並將連結內容改寫為相對位置參照；對於連結描述用的<a>標籤則忽略不寫入

儲存後的檔案中。經過上述處理後，將內容寫到使用者指定的路徑之下，完成下載的動作。

B. 書籍編輯器子模組

書籍編輯器的主要任務是提供作者方便的邏輯架構編輯環境，同時可以處理邏輯物件與實體資源間的對應關係，另外本研究所定義的書頁內容允許作者置入隱藏的 Term 標籤以供自動產生索引頁和外部連結網頁之用。本研究採用英國愛丁堡大學提供的開放源碼 IMS Editor 編輯器套件為基礎，並在該軟體中加入額外的元件類別，以滿足前述之相關需求。加入的兩個分別為儲存 Term 標籤內資訊用的表格資料結構 ToT 和網頁寫入器(Writer)，書籍編輯器之內部組態及作者編輯書本時的運作流程分別以圖 4-6 及圖 4-8 表示，圖 4-7 為資料流程圖的使用圖例說明。

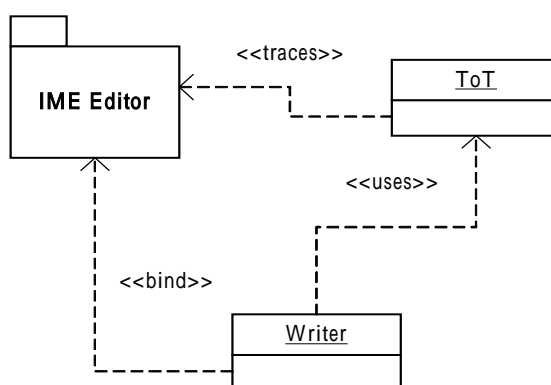


圖 4-6 書籍編輯器內部模組關係圖

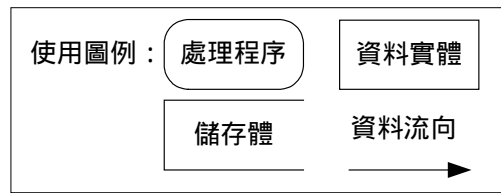


圖 4-7 資料流程圖圖例說明

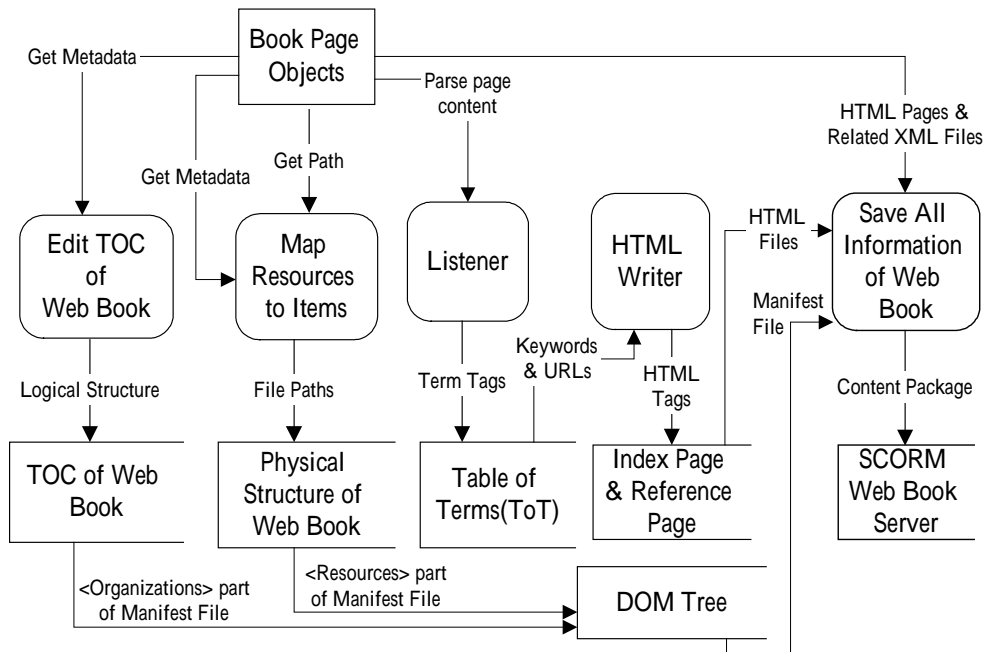


圖 4-8 書籍編輯器內部模組資料流程圖

作者在使用書籍編輯器時的工作流程主要可分為三個部分：編寫邏輯架構、選擇對應實體資源以及儲存課程包裹。在編寫內容架構的視覺化編輯環境中，如圖 4-2 所示，在視窗的左半邊作者可以新增 Item 到清單檔中，並調整其架構。點選需要和實體資源互相對應的 Item 後會在視窗的右半部出現讓作者可以填入實體檔案資源路徑資訊以及對應的 Metadata 等資訊的欄位，作者也可以透過 File Chooser 來選擇加入 Local Storage 中的檔案，並 Import 已經存在的 XML 檔案來作為該實體資源的對應 Metadata，或者直接使用書籍編輯器提供的介面填入所需的資訊。Listener 會聆聽作者進行實體

檔案與邏輯元件的對應這個動作，如有此動作發生，則透過 Parser 將實體資源檔案中的 Term 標籤內的資訊填入 ToT 中暫存起來，最後當作者確定儲存的動作後，系統會先呼叫 Writer 將 ToT 寫成索引頁與外部連結頁，將兩個檔案的相關資訊加入 Manifest 檔中，最後將實體資源以及 Manifest 檔案打包壓縮到同一課件包裹中，於是作者便完成了書籍編輯的動作。

4.2.2 書籍伺服模組

本研究將書籍伺服模組分為兩個部分，分別為書籍伺服器和範例知識庫，書籍伺服器採用 ADL 所提供的 Sample RTE 為主，而範例知識庫則是由本實驗室所規劃設計。以下將針對兩部分進行個別的分析：

A. 書籍伺服器

本研究所採用的書籍伺服器為 ADL 所提供的 Sample RTE 程式套件，該套件以 JSP 為主要的實作技術並結合 JAVA Servlet 與 JAVA Application 技術來提供教材管理(Content Management)、元件管理、學習歷程(Learner Profiles)以及教材編序(Sequencing)等服務，並附一個 Microsoft Access 編製完成的資料庫，用來管理使用者帳號以及課程元件等相關訊息，本論文附錄二為此資料庫的實體關係圖(Entity-Relation Diagram, E-R Diagram)。此外，伺服器還提供一個由 JAVA Applet 實作的 API Adapter 作為教材 SCO 和伺服

器溝通的管道，使用者透過瀏覽器連上伺服器後，LMS 便將 API Adapter 傳送使用者端的電腦上，而當使用者開啟一個課程開始學習活動之後，傳至使用者電腦中 HTML 格式的教材便利用 JavaScript 函式呼叫的方法將相關訊息透過 API Adapter 回傳至伺服器，並記錄在資料庫中。書籍伺服器與使用者間的溝通狀態如圖 4-9 所示。

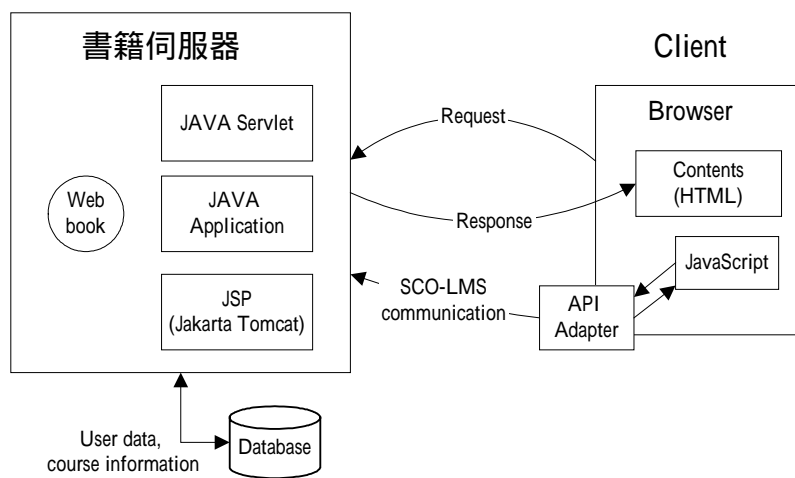


圖 4-9 書籍伺服器端與瀏覽器端溝通狀態示意圖

使用者身份在伺服器中分為兩種，分別是系統管理者和一般使用者，本研究將網路書籍的讀者(Reader)和作者(Writer)定義為一般使用者，課程元件以及伺服器管理的任務則由管理者擔任。一般使用者可以使用的功能分為：註冊課程(Register for a Course)、閱讀已註冊課程(View Registered Course)和修改密碼(Change Password)。系統管理者除了上述的功能外還可以匯入/刪除課程(Import/Delete a Course)、新增/刪除使用者(Add/Delete a User)以及清除資料庫(Clear Database)等管理選項可以選擇。關於上述功能與不同角色的

使用者間的關係，我們使用 UML 使用案例(Use Case)圖形來說明，如圖 4-10 所示。

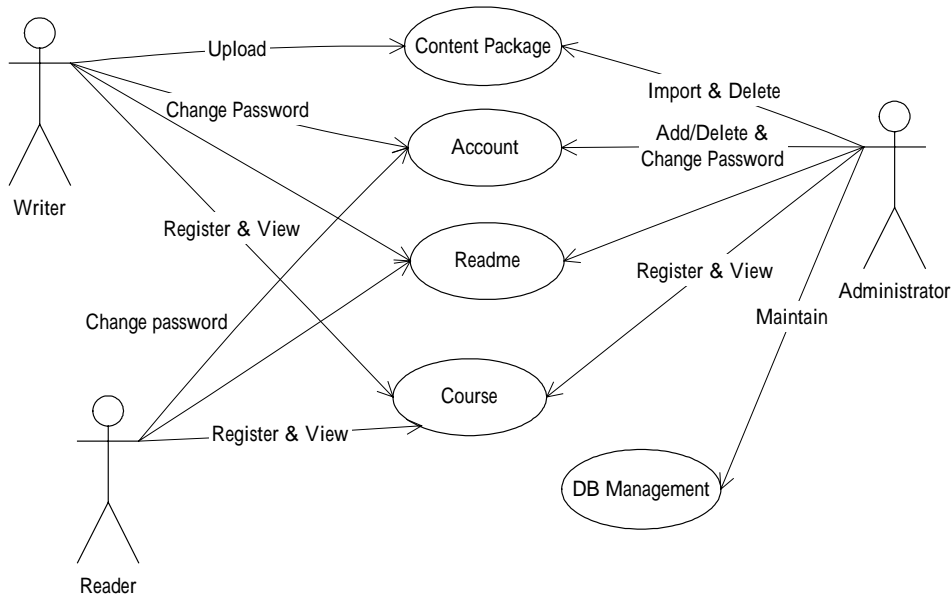


圖 4-10 書籍伺服器使用案例圖

B. 範例知識庫

除了以 ADL 的 Sample RTE 擔任書籍伺服器的工作外，本研究在書籍伺服器模組方面還提供了一個範例知識庫，作為作者編輯書本所需的基本素材。本研究選定由行政院研究發展考核委員會所主導的網路無障礙專案計畫為主要的領域進行知識管理架構規劃，並將規劃出來的分類法則以知識樹的形式表示，知識樹的詳細架構圖請參閱本論文附錄三。接著將本實驗室發展的無障礙網路空間規劃與設計手冊【7】以符合前述知識管理架構的方式切割成知識單元(Knowledge Unit, KU)，最後為個別的 KU 撰寫符合 SCORM 規範的 Meta Data，建置出教材元件庫。

無障礙網路空間規劃與設計手冊中所涵蓋的內容相當廣泛，由無障礙網路導論、全球網路無障礙趨勢與上網相關輔具、無障礙網站的規劃及設計、標準檢測碼與無障礙示範網站實例說明等相關議題組成。不同的 KU 雖然各隸屬於知識樹的不同分支，但由於編製成 SCORM 教材元件後彼此已自成獨立個體，故作者在編輯成網路書籍時，可依照個別不同的需求或者想要傳達的學習經驗特色來決定書本的邏輯架構，將元件動態組成網路書籍而不受限於知識管理架構原本的分類，在編輯網路書籍教材時可以有更大的彈性和自由，如此亦能落實本研究邏輯與實體架構相互獨立的基本精神。

第三節 實例說明

這一節中我們將實際呈現 SCORM 平台上之網路書籍模式，並以實例來說明書本建置與分享的程序。我們根據前文所述的步驟，自書籍伺服器資料庫中取出一本網路書籍，編寫對應的清單檔後，在 ADL 範例執行平台上展示出來，以驗證我們所提出的方法之可行性。範例之來源為身心障礙者服務網【25】之舊版網頁，下載後存於我們的書籍伺服器資料庫中。建置過程略述如下：

A. 教材元件編製

我們為原網頁中所用到的每張圖片附加 Meta Data 描述檔，將圖片編製成 Asset。每個獨立的網頁則視為書籍中的章節，以 SCO 的方式加以描述。為所有網頁檔案編寫完個別所需的 Meta Data 後，便可編寫清單檔，將所需的元件加入包裹中，使成為一完整的書籍。

B. 課件打包

我們比對該網路書籍的原始架構，編寫出一個具有同樣邏輯形式的 SCORM 教材架構，並打包起來，圖 4-11 與圖 4-12 為兩個版本的網路書籍之內容架構比較，SCORM 版本的內容架構是取自於 ADL 範例平台顯示畫面中的左子視窗。



圖 4-11 原始網路書籍內容架構

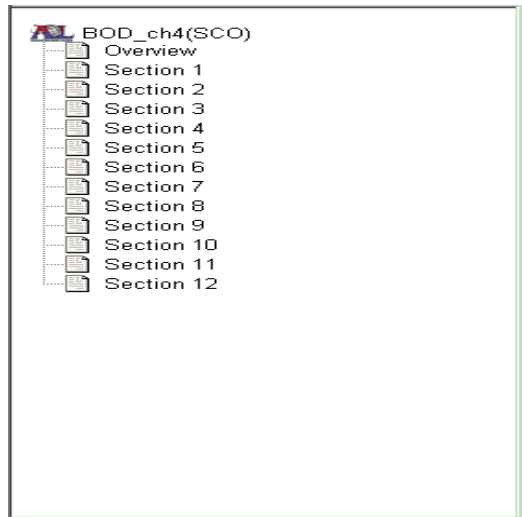


圖 4-12 SCORM 版網路書籍內容架構

C. 書本上線

將教材打包好以後，將檔案上傳至 SCORM 範例平台伺服器上，並由系統管理者將課程包裹匯入 ADL 範例執行環境中，讀者在登入系統後便可以依照自己的需求來挑選想要閱讀的書籍，完成選取的手續後便可以開始正式閱讀網路書籍。圖 4-13 和圖 4-14 分別展示原網路書籍與 SCORM 版的閱讀畫面。



圖 4-13 原始網路書籍閱讀畫面

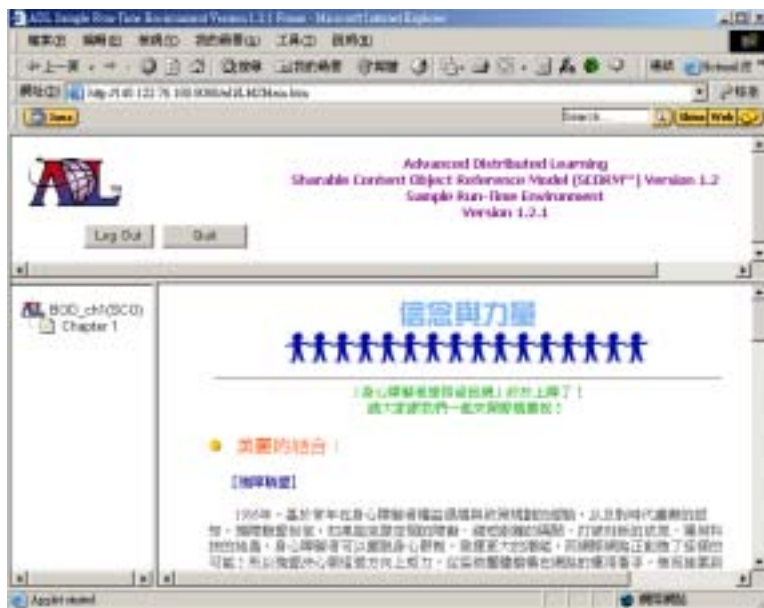


圖 4-14 SCORM 版網路書籍閱讀畫面

此外，結合前一小節所述的知識管理架構，我們也用切割完成的知識單元為書籍素材編寫了一本主題為關於無障礙網路規範的範例網路書籍，從以下範例中可以看出雖然知識單元有自己專屬的知識管理架構，但依照網路書籍模式理論所編輯而成的書籍仍然可以擁有個別的邏輯架構，瀏覽畫面如圖 4-15。



圖 4-15 SCORM 版本無障礙網路規範手冊瀏覽畫面