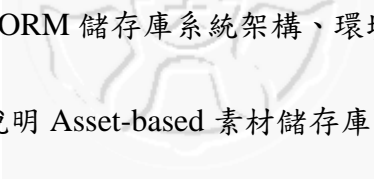


第五章 SCORM 儲存庫系統

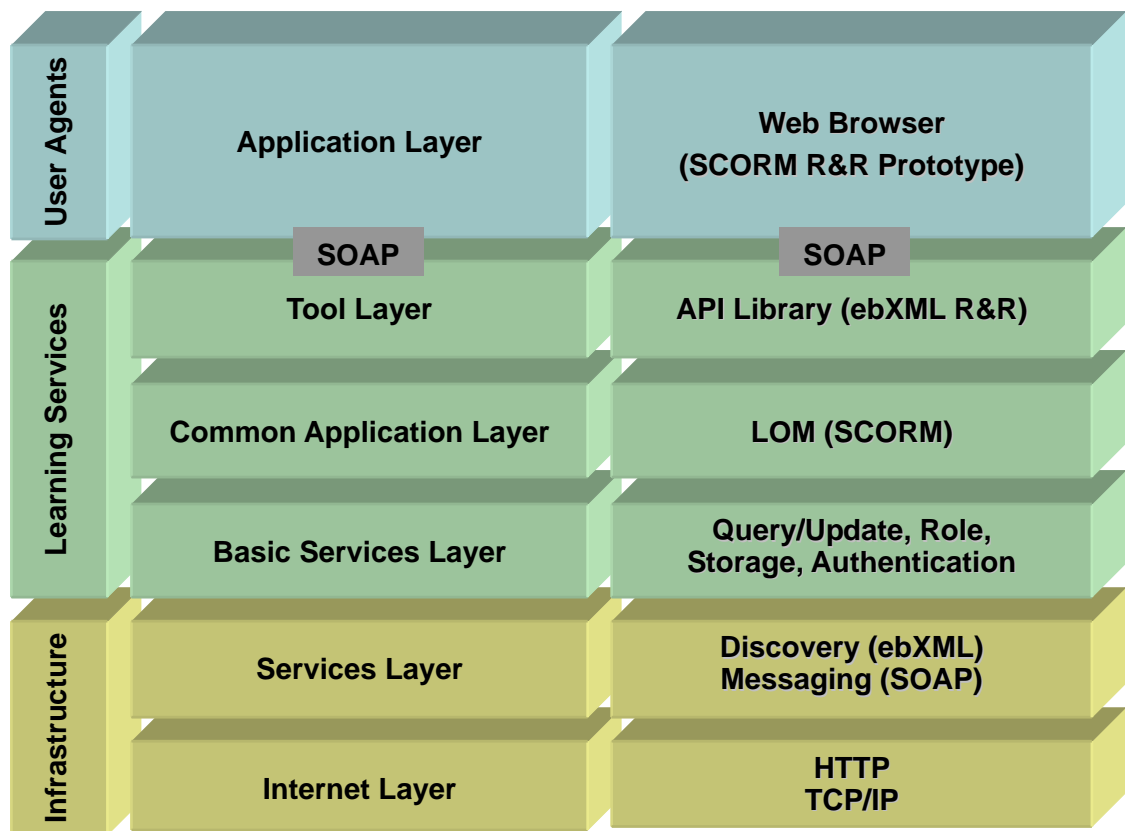


本章第一節將說明 SCORM 儲存庫系統架構、環境架設需求與系統的整體功能規劃。第二、三、四節說明 Asset-based 素材儲存庫、SCO-based 學習物件儲存庫、Content Aggregation-based 教材儲存庫三種模式的資料新增過程。第五節說明依物件類型查詢、依分類查詢、查詢 Metadata 等三種搜尋方式。

第一節 系統架構

5.1.1 SCORM 儲存庫 Prototype 三層式架構

本研究參考美國卡內基·梅隆大學，學習系統架構實驗室(LSAL)所提出的學習服務架構(Learning Services Architecture)【15】，對應 User Agents、Learning Services、Infrastructure 三層 Layers 特性，引入 ebXML R&R 的註冊與儲存庫機制，將之應用在數位學習 SCORM 2004 CAM 教材包裝標準上，採用 IEEE LTSC LOM 學習物件詮釋資料的描述能力，建立了 SCORM 儲存庫 Prototype 三層式架構，並已實作出一套可用的「SCORM 2004 儲存庫系統 Prototype」，其整體的架構如【圖 5-1】所示。



【圖 5-1】 SCORM 儲存庫 Prototype 三層式架構圖

此架構以 ebXML R&R 內容儲存庫標準為核心，而 Open Source ebxmlrr Project【18】參考 ebXML R&R 中的 ebRIM、ebRS 規範，實作出一套 API 函式庫，於 Infrastructure 層提供 ebXML 搜尋及 SOAP 訊息包裝能力；於 Learning Services 層提供 Metadata (ebRIM) 的查詢與更新能力，及角色管理、RepositoryItem 儲存管理，使用者安全認證等基本服務；Tool Layer 的 API 函式庫則將以上功能包裝成 SOAP 及 SOAP with attachments 形式。而本研究利用 ebxmlrr 提供的 API 函式庫，於 Common Application Layer 實作 SCORM 2004 CAM 規範中的 IEEE LTSC LOM，因而建立起「SCORM 2004 儲存庫系統 Prototype」。

5.1.2 SCORM 儲存庫環境架設需求

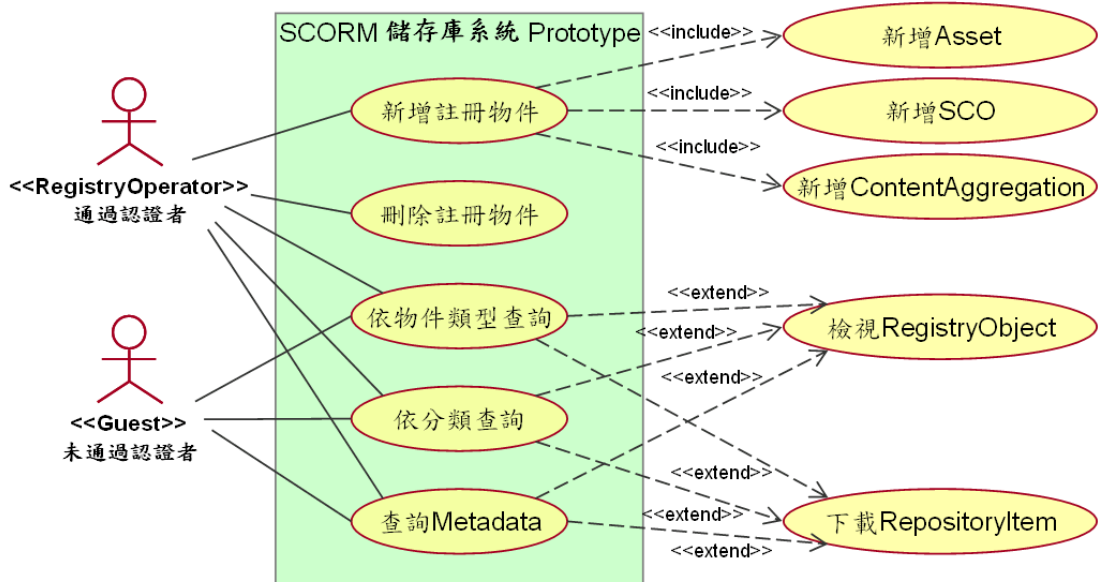
「SCORM 2004 儲存庫系統 Prototype」建立在 ebxmlrr 的基礎之上，環境架設需求如【表 5-1】所示。本系統採用 Java 環境，故需要安裝 J2SDK 1.4.2 以上版本，亦建議使用 1.4.2 版本，因根據本人測試結果，J2SDK 5.0 與 ebXML R&R 實作 Project：ebxmlrr (omar-3.0-alpha2 版) 之間存在不相容問題。Tomcat 5.0.28 是由 Apache Software Foundation 支持的 Jakarta 計畫所開發的，支援 Servlet 2.4 與 JSP 2.0，用來解析 JSP (Java Server Page) 程式，傳送與接收 SOAP 訊息。關聯式資料庫 PostgreSQL 8.0.3 用以儲存 ebRIM 資訊模型，換句話說即是儲存學習物件的 Metadata 資訊。最後，ebxmlrr 將以上程式整合為完整的 ebXML R&R 內容儲存庫，並提供一組 API 函式庫，讓我們得以實作 JSP 程式，將 LOM 及學習物件註冊/儲存進 ebXML R&R。(註：除了本研究開發的原始程式未放在網路上之外，其餘軟體皆可於網路上免費下載使用。)

【表 5-1】SCORM 2004 儲存庫系統環境架設需求表

類型	本研究採用版本
Java 執行環境	Java™ 2 SDK, Standard Edition Version 1.4.2_06
Web Container	Tomcat 5.0.28
關聯式資料庫	PostgreSQL 8.0.3
ebxmlrr Project	omar-3.0-alpha2
SCORM 2004 儲存庫系統	本研究開發版本(JSP)

5.1.3 SCORM 儲存庫功能規劃

「SCORM 2004 儲存庫系統 Prototype」直接採用 ebxmlrr 提供的 X.509 證書的安全認證機制，做為使用者權限的區分，未通過認證者只有 Guest 權限，只能夠執行(1)依物件類型查詢、(2)依分類查詢、(3)查詢 Metadata 等功能，而對於這三種查詢結果，使用者能夠進一步執行檢視 RegistryObject 或是下載 RepositoryItem 的功能。通過認證者則具備 RegistryOperator 權限，除了上述三種查詢功能之外，亦能執行新增註冊物件與刪除註冊物件兩種功能，而新增註冊物件又可細分為新增 Asset、新增 SCO、新增 Content Aggregation 等三項子功能。【圖 5-2】說明上述情境。

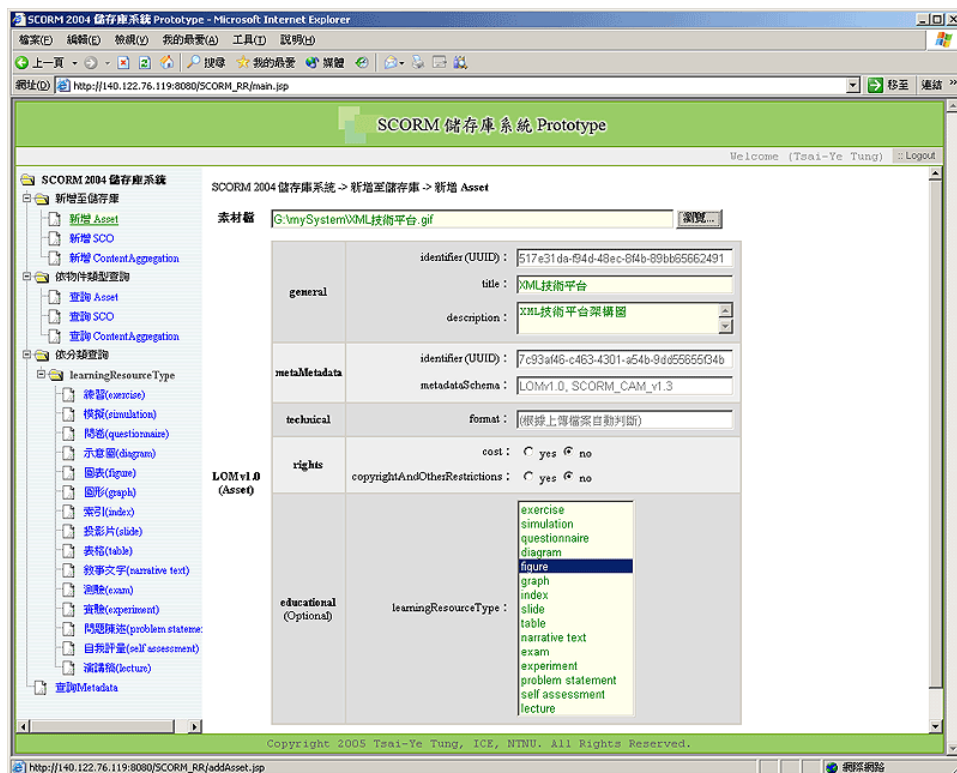


【圖 5-2】 SCORM 儲存庫系統 Use Case Diagram

第二節 Asset-based 素材儲存庫

此種儲存庫以 Asset 為儲存單元，將依分類結構儲存素材(如*.jpg、*.gif、*.wav)

與對應 Metadata 之 ZIP 壓縮檔。新增 Asset 的執行畫面如【圖 5-3】所示。



【圖 5-3】SCORM 2004 儲存庫系統：新增 Asset 畫面

為了加快使用者編輯速度，以提高使用者對系統的使用意願，系統為許多欄位加上預設值，茲說明如下：

general 及 metaMetadata 類別底下的 identifier 元素在本研究的 Prototype 系統中採用 UUID (Universally Unique Identifier，是一組 128bit 的數字，其演算法讓程

式能在分散式環境下產生不會重複的 ID)，而此 ID 將由系統程式自動產生，不需要使用者輸入。

metaMetadata 類別底下的 metadataSchema 元素依照 SCORM CAM 規定值為 LOMv1.0, SCORM_CAM_v1.3，亦不需要使用者輸入。

technical 類別底下的 format 元素將依據使用者上傳的素材檔，由程式自動判斷其 MIME 類型，亦不需要使用者輸入，程式同時會判斷素材檔的位元組大小，產生 size 元素。

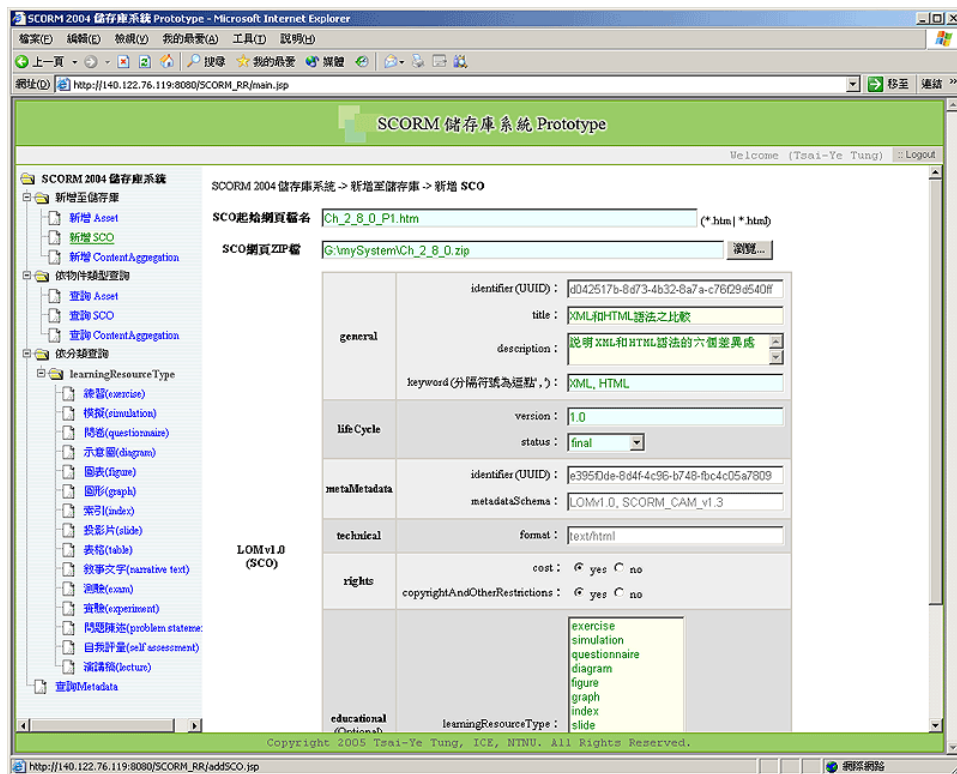
rights 類別底下的 cost(是否需付費)、copyrightAndOtherRestrictions(是否有版權或其他限制) 只能使用 IEEE 規定的 yes 或 no 兩項辭彙，在本系統中預設為 no，需要時亦可改為 yes。

換句話說，我們大部分情況下只須選擇欲上傳的 Asset 素材檔，例如 XML 技術平台.gif，編寫淺黃底色的文字框項目：title(標題)、description(文字描述)，並於分類選單中選擇 learningResourceType(學習資源類型)，即完成一項 Asset LOM 的編輯工作。而使用者按 Submit 送出後，程式將依序執行下列步驟：

1. 接收上傳的 Asset 素材檔。
2. 依據使用者填寫資訊，產生 Asset Metadata 檔。
3. 將 Asset 素材檔及 Asset Metadata 檔壓縮成單一 ZIP 檔。
4. 將 Asset Metadata 對應成 Asset-RO 後註冊進儲存庫，並做好分類。
5. 將 ZIP 檔存入儲存庫的 RepositoryItem。

第三節 SCO-based 學習物件儲存庫

此種儲存庫以 SCO 為儲存單元，將依分類結構儲存單一 SCO 之 Resource Package (*.zip)，裡面除了網頁及相關資源檔案之外，亦包含 SCO Metadata 及 Resource Manifest 兩份 XML 檔，以及與平台溝通用的 JavaScript 檔、驗證課程包裝用的 XSD 檔等。新增 SCO 的執行畫面如【圖 5-4】所示。



【圖 5-4】SCORM 2004 儲存庫系統：新增 SCO 畫面

相較於 Asset Metadata 的必要性元素，SCO 必要性元素多了 lifeCycle 類別底下的 version 及 status，以及 general 類別底下的 keyword 共三項元素，而 version

及 status 在本系統中都有預設值，故使用者基本上只需要多填 keyword 一項，即可符合 SCO Metadata 的必要元素需求，除此之外，使用者需先將欲製作成 SCO 的網頁及相關檔案壓縮成單一 ZIP 檔，並填寫 SCO 起始網頁(或稱為首頁)的檔名，其餘欄位則與新增 Asset 時相同，茲說明如下：

general 及 metaMetadata 類別底下的 identifier 元素在本研究的 Prototype 系統中採用 UUID (Universally Unique Identifier，是一組 128bit 的數字，其演算法讓程式能在分散式環境下產生不會重複的 ID)，而此 ID 將由系統程式自動產生，不需要使用者輸入。

lifeCycle 類別底下的 version 元素能讓使用者自由輸入文字，預設值為 1.0，而 status 元素限定為 draft、final、revised、unavailable 四者選一，預設值為 final。

metaMetadata 類別底下的 metadataSchema 元素依照 SCORM CAM 規定值為 LOMv1.0, SCORM_CAM_v1.3，不需要使用者輸入。

technical 類別底下的 format 元素設定為 text/html，亦不需要使用者輸入。

rights 類別底下的 cost(是否需付費)、copyrightAndOtherRestrictions(是否有版權或其他限制) 只能使用 IEEE 規定的 yes 或 no 兩項辭彙，在本系統中預設為 yes，需要時亦可改為 no。

換句話說，我們只須填入 SCO 起始網頁檔名，並選擇欲上傳的 SCO 網頁壓縮檔，例如 Ch_2_8_0.zip，編寫淺黃底色的文字框項目：title(標題)、description(文

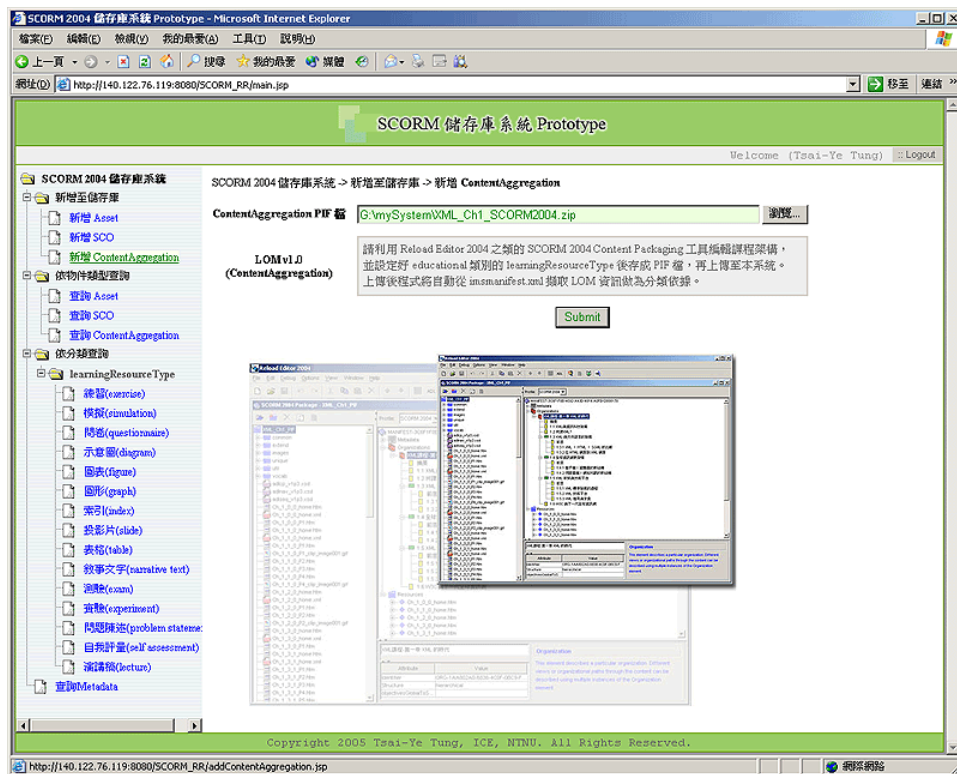
字描述)，以及淺藍底色的文字框項目：keyword(關鍵字)，而 keyword 若不止一個時須以半形逗號分隔，最後於分類選單中選擇 learningResourceType(學習資源類型)，即完成一項 SCO LOM 的編輯工作。而使用者按 Submit 送出後，程式將依序執行下列步驟：

1. 接收上傳的 SCO 網頁壓縮檔。
2. 加入 APIWrapper2004.js 及 SCOFuctions2004.js 檔，產生新的 SCO 起始網頁。
3. 依據使用者填寫資訊，產生 SCO Metadata 檔。
4. 分析 SCO 網頁壓縮檔內容，產生 SCO Resource Manifest 檔。
5. 將步驟 2, 3, 4 產生的檔案，加入上傳的 SCO 網頁壓縮檔。
6. 將 SCO Metadata 對應成 SCO-RO 後註冊進儲存庫，並做好分類。
7. 將新的 SCO 網頁壓縮檔存入儲存庫的 RepositoryItem。

經由本系統重新包裝過的 SCO 網頁壓縮檔，將保證能夠順利通過 ADL SCORM 2004 Conformance Test Suite Version 1.3.3 (Self Test) 【11】官方相容性測試套件中的 Metadata Conformance Utility Test 以及 Sharable Content Object (SCO) Run-Time Environment (RTE) Conformance Utility Test，說明了本系統正確無誤地產生 SCO Metadata，並成功地加入 APIWrapper 與 SCOFuctions 兩支 JavaScript，確定此 SCO 能夠被 SCORM RTE 正確啟動(Launch)並紀錄瀏覽時間。

第四節 Content Aggregation-based 教材儲存庫

此種儲存庫以 Content Aggregation (PIF)為儲存單元，將依分類結構儲存具課程架構之 Content Aggregation Package (*.zip)，此為可放入 SCORM RTE 平台內執行的完整教材。新增 Content Aggregation 的執行畫面如【圖 5-5】所示。



【圖 5-5】SCORM 2004 儲存庫系統：新增 Content Aggregation 畫面

本系統未實作線上編輯課程架構的功能，相反地，本系統提供自動分類功能，當使用者利用 Reload Editor 2004【17】之類的 SCORM 2004 Content Packaging 工具編輯完成課程架構，並設定好 educational 類別的 learningResourceType 後存

成 PIF 檔，再上傳至本系統。上傳後程式將自動從 imsmanifest.xml 擷取 LOM 資訊，將之註冊進儲存庫系統，並擷取 learningResourceType 做為分類依據。使用者按 Submit 送出後，程式將依序執行下列步驟：

1. 接收上傳的 PIF 檔。
2. 取得壓縮檔內 imsmanifest.xml 中的 LOM 資訊。
3. 將 LOM 資訊對應成 Content Aggregation-RO 後註冊進儲存庫，並做好分類。
4. 將 PIF (ZIP 檔) 存入儲存庫的 RepositoryItem。

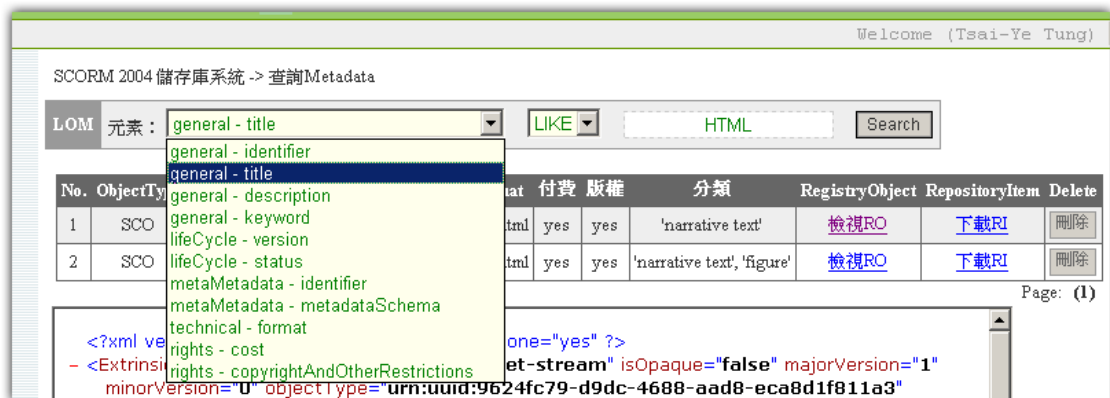
若使用者將儲存庫內的 SCOs 以 Reload Editor 2004【17】編輯課程架構，再匯出成 PIF 檔，將能夠順利通過 ADL SCORM 2004 Conformance Test Suite Version 1.3.3 (Self Test)【11】官方相容性測試套件中的 Content Package Conformance Test，說明了其 PIF 檔能夠執行在符合 SCORM 2004 的平台之上。

未來可望加入從儲存庫內的 SCOs 直接組成 Content Aggregation，並提供線上編輯課程架構，轉成 PIF 之功能。如此一來，整個儲存庫系統方達到垂直整合的目標。

第五節 儲存庫查詢模式

目前系統已實作依物件類型查詢、依分類查詢以及查詢 Metadata 三種搜尋方式，茲分述如下：

1. 依物件類型查詢共有 Asset、SCO、Content Aggregation 三種類型。
2. 依分類查詢則是從 LOM 中的 learningResourceType 元素做選擇。
3. 查詢 Metadata 則屬於自由度最高的進階查詢方式，如【圖 5-6】所示。



【圖 5-6】SCORM 2004 儲存庫系統：查詢 Metadata 畫面截圖

使用者選擇某一項 LOM 中的 Metadata 元素，再選擇比對方式，其中「=」代表字串精確比對，「LIKE」代表字串模糊比對，最後輸入任意字串即可對儲存庫中的所有學習物件進行學習物件 Metadata 的查詢。

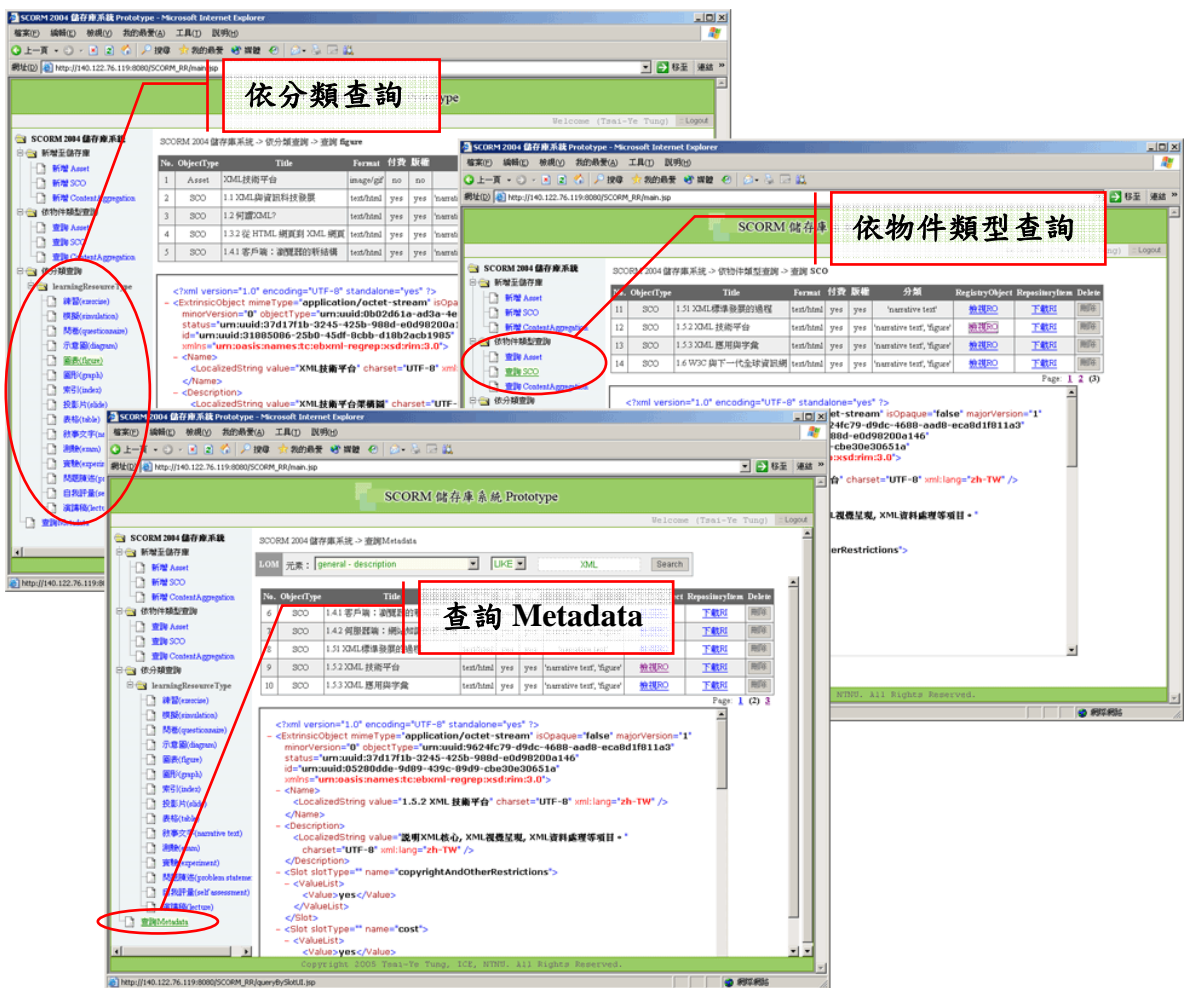
由於 LOM 資訊都已經對應成 ebRIM 格式，對應關係如【表 5-2】所示。系統的「依分類查詢」功能實作 ebXML R&R 中的 Classification Searching，可查詢 **educational.learningResourceType** 元素；「查詢 Metadata」功能則結合 ebXML R&R 中的 RO-Slot Searching、RO-Name Searching 與 RO-Description Searching，透過 RO-Name Searching 可查詢 **general.title** 元素，透過 RO-Description Searching 可查詢 **general.description** 元素，透過 RO-Slot Searching 則能查詢其餘的 LOM 元素。這代表的是所有的 LOM 元素在本系統中皆能查詢，換句話說，本研究支援完整的內容儲存庫 Metadata 查詢能力。

【表 5-2】LOM 與 ebRIM 對照表

LOM	Multiplicity	ebRIM	查詢功能
general.identifier	1	RO.Slot	查詢 Metadata
general.title	1	RO.Name	查詢 Metadata
general.description	1	RO.Description	查詢 Metadata
general.keyword	1...10	RO.Slot	查詢 Metadata
lifeCycle.version	1	RO.Slot	查詢 Metadata
lifeCycle.status	1	RO.Slot	查詢 Metadata
metaMetadata.identifier	1	RO.Slot	查詢 Metadata
metaMetadata.metadataSchema	2...10	RO.Slot	查詢 Metadata
technical.format	1	RO.Slot	查詢 Metadata
rights.cost	1	RO.Slot	查詢 Metadata
rights.copyrightAndOtherRestrictions	1	RO.Slot	查詢 Metadata
educational.learningResourceType	1...10	Classification	依分類查詢

以上三種不同搜尋方式的查詢結果，將以相同的分頁列表方式呈現，畫面如

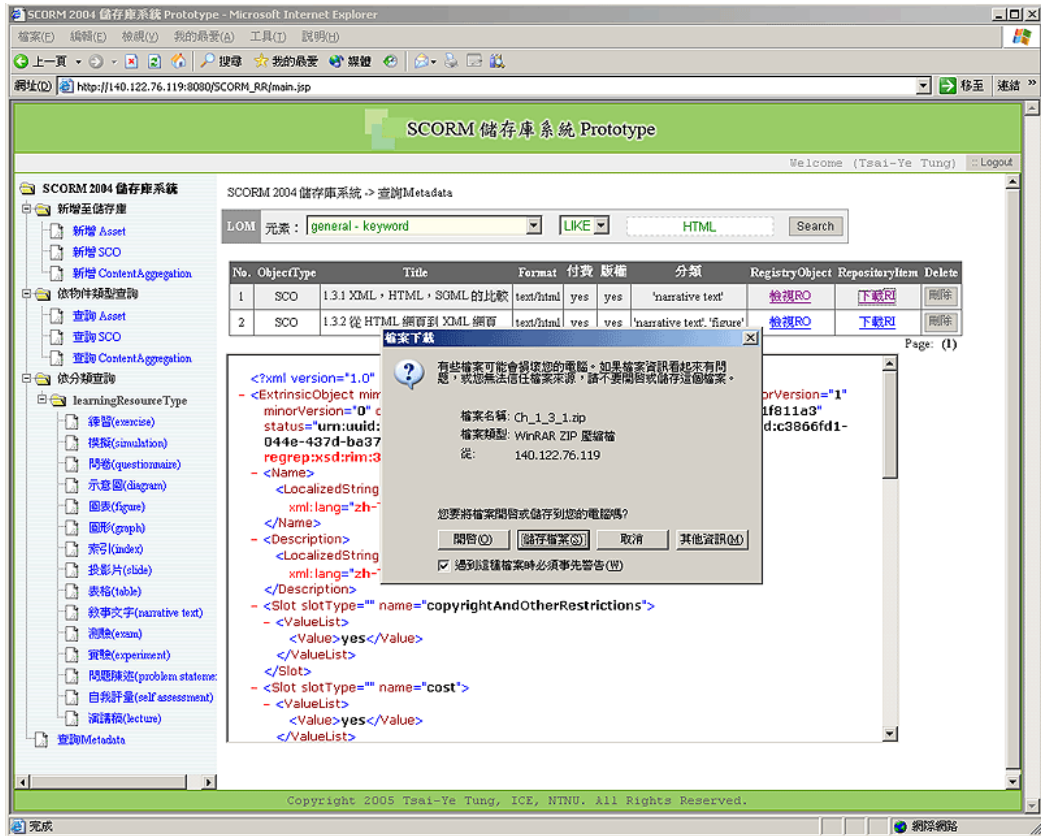
【圖 5-7】所示，若為通過認證的使用者，則結果列表最右邊會多出刪除按鈕，然而，每個學習物件只會隸屬於當初上傳此物件之通過認證的使用者，故此刪除功能只能夠刪除自己上傳的學習物件，無法刪除其他使用者上傳的學習物件。另外，使用者可以點選頁面編號切換至其他頁面，觀看其他筆資料，或是點選檢視 RegistryObject 超連結，觀看 ebXML R&R 內容儲存庫如何註冊/儲存此項學習物件的 Metadata 資訊。



【圖 5-7】 SCORM 2004 儲存庫系統：檢視 RegistryObject 畫面

當使用者需要進一步觀看 Asset、SCO、Content Aggregation 實際內容時，可

以點選下載 RepositoryItem 超連結，以取得實際檔案，畫面如【圖 5-8】所示。



【圖 5-8】SCORM 2004 儲存庫系統：下載 RepositoryItem 畫面