

數位學習 SCORM 教材註冊及儲存庫管理之研究

葉耀明 董才業

國立臺灣師範大學資訊教育學系

ymyeh@ice.ntnu.edu.tw tsaiye@ice.ntnu.edu.tw

摘要

近年來，數位學習產業迅速地發展，為了讓數位教材在不同教學網站的學習管理系統間重複使用，製作符合數位學習 SCORM 標準的數位教材是未來的趨勢。在此趨勢下，各教學網站會產生可進一步整合的數位教材，因此數位教材之儲存庫需求因應而生。在此環境下，數位教材儲存庫必須具備符合國際標準、開放式整合其他儲存庫的特性，才能滿足符合 SCORM 標準之數位教材的需求。

本研究參考美國卡內基 梅隆大學，學習系統架構實驗室提出的學習服務三層式架構，並導入 ebXML 註冊與儲存庫標準。將之應用在數位學習 SCORM 2004 教材內容包裝標準上，採用 IEEE 學習科技標準委員會發展的學習物件詮釋資料標準的描述能力，建立了 Asset-based 素材儲存庫、SCO-based 學習物件儲存庫、Content Aggregation-based 教材儲存庫等三種不同顆粒度之教材儲存庫模式，並實作出使用 SCORM 2004 標準之教材儲存庫系統。本系統除了具備開放標準、聯合互作、物件導向資料庫架構等特色之外，並發展出三種物件類型的查詢、分類系統查詢，與註冊物件詮釋資料進階查詢等功能。整體研究成果將有助於架構出一有效的數位學習 SCORM 教材之內容管理機制。

關鍵詞：SCORM、ebXML Registry & Repository、

LOM、e-Learning

1. 前言

早在電腦誕生後不久就有許多相關領域的學者專家嘗試將電腦應用在輔助學習方面，但是由於軟硬體及其他方面的技術限制，現存的電腦教學系統往往規格不一，缺乏一個可以依循的共通標準，各系統皆採用自己獨有的資料格式，導致教材在共享(Share)和重用(Reuse)上非常困難，也帶來開發成本上不必要的浪費，故實有必要建立一套規格標準，來加以規範平台與教材的一致性溝通程序。有鑑於此，在 1997 年底，美國國防部(DoD)主導「先進分散學習計畫」(Advanced Distributed Learning, 簡稱 ADL) [1]，參照許多機關所制訂的規格並遵循 W3C (World Wide Web Consortium)所提出的 XML 語言規範，提出共享教材元件參考模式(Sharable Content Object Reference Model, 簡稱 SCORM)作為數位學習的共同標準。SCORM 標準的誕生宣告了 e-Learning 的時代正式來臨，在網際網路上建立符合規範的學習環境並且提供教材與相關的學習模式亦成為當前推廣 e-Learning 首要的工作。

e-Learning 具備即時、便捷、跨國界及不受時空限制等特性，其基本構成之一在於數位教材，然而在以往的網路教學中，各學習管理系統(Learning Management System, 簡稱 LMS)間架構不同，對於課程資訊之定義亦不盡相同，因此若要將原有教材移植至其它平台中使用，有許多執行上之困難。為解決以往各學習平台間因為資料規格不一所造成之教材無法重複使用之問題，SCORM 提出教材內容整合模式(Content Aggregation Model, 簡稱 CAM) [2]作為制定學習物件的標準。而為了讓教材具備互相交換、重複使用等特性，本研究即採用 SCORM 2004 作為數位學習教材包裝的標準，其內容涵蓋教材本身開發原則、教材包裝格式、教材呈現順序及教材執行環境的實作規格等範疇，並將教材細分為多種層級的學習物件，具備學習物件重新組合再利用、教材跨學習平台交換等特性。

隨著時間的演進，製作完成的數位教材數量將愈來愈多，如何建立一個永續的儲存庫系統將是刻不容緩的課題。如今資料庫技術成熟，自己設計資料表來儲存與管理數位教材雖然是可行的，但是系統是封閉的，很容易遭到時代的淘汰；一個永續發展的系統，最簡單的做法就是符合時代潮流；而欲符合時代潮流最直接的做法便是符合國際標準。倘若儲存庫系統結合了國際標準，則就大大的降低被淘汰的機會，故本研究導入內容儲存庫標準 ebXML Registry&Repository (ebXML 註冊&儲存庫，或稱 ebXML R&R) [3-5]做為數位教材儲存庫系統的核心。

因此本文主要討論 SCORM 儲存庫的相關議題，第二節先探討 SCORM 及 ebXML 註冊&儲存庫的背景知識，第三節探討 LOM 及 ebRIM 的對應關係，第四節探討 SCORM 儲存庫系統的相關實作問題，第五節為結論與未來工作。

2. SCORM 及 ebXML 註冊&儲存庫探討

2.1 SCORM

2004 年 1 月 30 日，ADL 推出了新版本 SCORM 2004 (亦稱為 SCORM 1.3)，2004 年 7 月 22 日修正部份小錯誤，訂為 SCORM 2004 2nd Edition (亦稱為 SCORM 1.3.1，本文以下統稱為 SCORM 2004)，而 SCORM 2004 的整體目標是讓教材內容達到下列諸項要求：

- (1) 容易取得(Accessibility)：可輕易在本地或是遠端讀取課程資訊。
- (2) 適性化(Adaptability)：可根據個人及組織需求

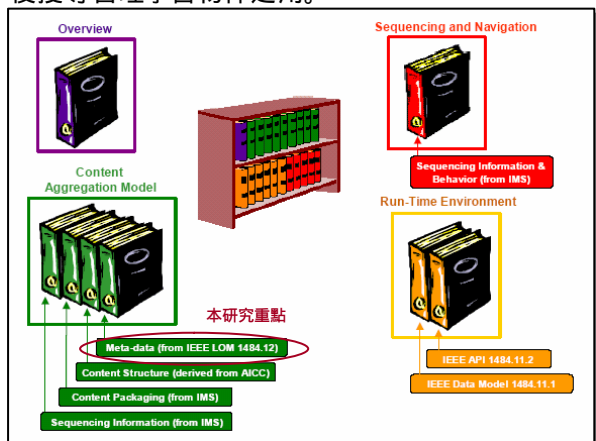
安排不同的課程。

- (3) 可負擔(Affordability)：減少遞送教材花的時間與費用，增進效率與生產力。
- (4) 耐用性(Durability)：教材不會因科技進步或標準異動而無法使用。
- (5) 互相通用(Interoperability)：教材可在不同平台呈現，透過不同工具重編輯。
- (6) 可重複使用(Reusability)：可輕易合併教材於其他系統，或其他教學內容。

SCORM 2004 規格書分為四個部分：

- (1) 概要(Overview) 主要是介紹 SCORM 的來龍去脈以及相關資訊。[1]
- (2) 教材內容整合模式(Content Aggregation Model, 簡稱 CAM)主要是介紹教材元件的包裝標準。[2]
- (3) 執行環境(Run-Time Environment, 簡稱 RTE) 主要目的是讓學習平台與教材之間有統一的溝通介面。[6]
- (4) 順序與導覽(Sequencing and Navigation, 簡稱 SN) 主要敘述教材如何編排呈現順序以及學習系統如何處理學習者的學習行為。[7]

如圖一所示，本研究著重於 Content Aggregation Model 中的 Learning Object Metadata (LOM)，用來描述學習物件並註冊進儲存庫，供日後搜尋管理學習物件之用。



圖一 SCORM 2004 規格書架構圖

[資料來源：www.adlnet.org]

SCORM 的 Content Aggregation Model, 支援元件式教學模組，將教材的編寫過程視為學習物件的重新組合，可用來描述學習物件，並依此標準將各個學習物件包裝成教材，方便交換教材內容，以達到教材內容的知識分享。因應這樣的編輯模式，SCORM CAM 定義了內容模式(Content Model)、詮釋資料(Meta-data)與內容包裝(Content Package)三種元件模式，而 SCORM 2004 於 SCORM 1.2 原有架構下，增加了順序規則的資訊 (Sequencing Information)，茲就三種元件模式分述如下：

- (1) Content Model：定義出課程教材的資源類型。SCORM 2004 將教材內容定義為素材(Asset)

共享內容元件(SCO, Sharable Content Object)與內容組織(Content Organization)三種，說明如下：

- ◇ Asset：課程教材最基本的組成單位，例如文字、圖形、聲音、或是網頁等可以被傳送至使用者端的素材。
- ◇ SCO：學習管理系統(LMS)可以管理與追蹤的最小單位，可以由一個或一個以上的 Assets 所組成，例如一個網頁包括一張圖片與一段聲音。換句話說，SCO 與 Asset 最大的不同點在於只有 SCO 具備與 LMS 溝通的能力，而所謂的溝通指的是可以取得或設定 LMS 提供的變數，所有這些變數的集合就是資料模型(Data Model, 這部分將於後面介紹)。
- ◇ Content Organization：指的是課程架構，如書本的章節、單元或活動(Activities)等組成結構。結合上述三類內容模式，即成為完整的課程教材(Content Aggregation)。

- (2) Meta-data：描述課程教材資源內容的制式規格。

SCORM 2004 採用 IEEE 學習科技標準委員會(Learning Technology Standards Committee, 簡稱 LTSC)制定的學習物件詮釋資料 1.0 版(Learning Object Metadata v1.0, 簡稱 LOM v1.0), 作為描述課程教材資源之用。LOM 可以內嵌於學習清單檔 imsmainfest.xml 之中或是利用 imsmainfest.xml 內的 <adlcp:location>標籤語法獨立於單一 XML 檔中。

而 LOM v1.0 包含九種類別(Category)，每一類別裡面都包含許多 XML 元素，可以填資料值的元素共計有 58 個，對於九個類別的內容分述如下：

1. 一般(general)：描述學習資源一般性資訊，如標題、使用語言、關鍵字、涵蓋範圍、結構、整合程度等。
2. 生命週期(lifeCycle)：描述學習資源的版本、狀態和提供者的相關資料。
3. 描述資料(metaMetadata)：描述學習資源 Metadata 的提供者及其參考的 Schema 版本。
4. 技術(technical)：描述學習資源技術需求方面的訊息，如格式、大小、位置、執行環境需求、安裝教學、其他軟硬體限制、使用時間長度等。
5. 教育(educational)：描述學習資源在教育及教學方面的特性，如互動類型、學習資源類型、互動程度、預期使用者、使用情境、使用年齡層、難度、學習所需時間、使用方式等。
6. 版權(rights)：描述這份學習資源的收費及版權訊息。
7. 關聯(relation)：描述這份學習資源和其他學習資源之間的關係。
8. 註解(annotation)：包括學習資源的評論者、發表評論的時間和其評論內容，幫助教材引用者更瞭解此份學習資源的意義。
9. 分類(classification)：採用其他分類系統來做學習資源的分類，補足一般分類的不足。

(3) Content Package：描述整合與建立課程教材的需求與方式。

整合前述的內容模式(Asset、SCO 與 Content Organization)、詮釋資料、控制檔(包括與平台溝通用的 JavaScript 檔、驗證課程包裝用的 XSD 檔)，再加上學習清單檔 imsmanifest.xml，經壓縮後成為課程封裝交換檔(Package Interchange File, PIF)，包含以下兩種類型：

- ✧ Resource Package：不包含課程章節架構的 PIF，純粹將多個學習物件加入或不加入 Metadata 後，予以包裝在一起，方便用來與他人交換。
- ✧ Content Aggregation Package：包含課程章節架構的 PIF，可直接匯入學習平台中使用或是與他人交換。

Content Package 提供一種方法，讓數位學習資源可在不同的系統之間做交換，讓教材達到跨平台使用的目標。而 Content Package 分兩個主要部份：學習清單(Manifest)與實體檔案(Physical files)，分述如下：

- ✧ Manifest 必須有詮釋資料、課程結構(organizations)、學習資源實體檔案位置(resources)；其中詮釋資料是用於描述整個包裝(Package)中各種不同層次的內容；課程結構則是用於描述包裝裡的內容結構與組織，並定義教材的順序；學習資源實體檔案位置則用於描述包裝所參考到的內外部資源，不過沒有考慮順序或階層的問題。
- ✧ Physical files 則是表示在 PIF 當中所參考到的實際檔案，可能是在內容包裝內的內部檔案，也可能是被 Universal Resource Indicator(URI)所參考的外部檔案，最後把所有的實體檔案壓縮成一個課程封裝交換檔(PIF)。

2.2 ebXML 註冊&儲存庫

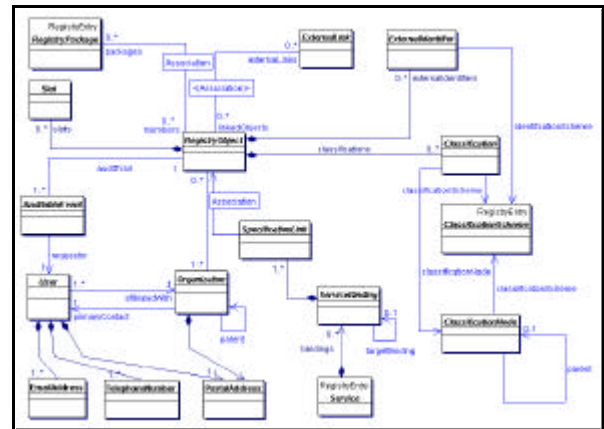
ebXML 是由聯合國貿易促進與電子商務中心(UN/CEFACT)與美國結構化資訊標準推動組織(OASIS)共同推動的一項電子商務架構標準。

而 OASIS 的發展重點則放在 ebXML 標準架構的底層部份，提出 ebXML 註冊資訊模型規範(OASIS/ebXML Registry Information Model Specification, 簡稱 ebRIM) [4]以及 ebXML 註冊服務規範(OASIS/ebXML Registry Services Specification, 簡稱 ebRS) [5]等內容儲存庫相關標準，其版本演進如下：

- ✧ ebRIM 與 ebRS：
 - V1.0 – 2001 年 5 月。
 - V2.0 – 2001 年 12 月。
 - V2.1 – 2002 年 6 月。
 - V2.5 – 2003 年 6 月。
 - V3.0 Draft – 2005 年 2 月。

如圖二所示, ebXML 註冊服務的基本資料單元

為註冊物件(RegistryObject)，這些註冊物件的格式是依照 OASIS 組織訂定的 ebRIM 標準來實作的。每一個註冊物件均附有 Metadata，以供註冊服務處作為分類、及其生命週期內的各種控管之用。



圖二 ebXML 註冊資訊模型圖

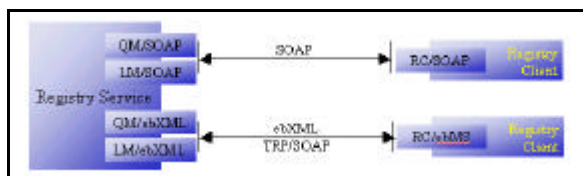
[資料來源：www.oasis-open.org]

一個註冊物件(RegistryObject)具有下列屬性：

- ✧ 屬性欄位(Slot)：用來增加註冊物件額外的屬性欄位。
- ✧ 關聯性(Association)：描述一個物件和其他物件間的關聯性。
- ✧ 外部識別碼(ExternalIdentifier)：提供外部識別碼。
- ✧ 外部連結(ExternalLink)：提供額外描述資訊的連結。
- ✧ 分類架構(ClassificationScheme)：分類樹的根節點。
- ✧ 分類節點(ClassificationNode)：分類樹根節點之外的子節點或結束節點。
- ✧ 分類(Classification)：說明註冊物件屬於分類樹的哪一類。
- ✧ 註冊套裝(RegistryPackage)：將相關註冊物件邏輯地綁在一起。
- ✧ 稽核事件(AuditEvent)：稽核與追蹤註冊物件用。
- ✧ 使用者(User)：和註冊物件有關的使用者。
- ✧ 郵寄地址(PostalAddress)：和註冊物件有關的郵寄地址。
- ✧ 電話號碼(TelephoneNumber)：和註冊物件有關的電話號碼。
- ✧ 電子郵件地址(EmailAddress)：和註冊物件有關的電子郵件地址。
- ✧ 組織(Organization)：和註冊物件有關的組織，例如註冊物件的提供組織。
- ✧ 服務資訊(Service)：例如 Web Services。
- ✧ 服務繫結方式(ServiceBinding)：提供如何繫結服務的技術資訊。
- ✧ 服務技術規範連結(SpecificationLink)：提供存取服務的連結資訊，如 WSDL。

ebXML 註冊&儲存庫架構是由服務端和客戶

端組成。服務端提供儲存庫的管理介面，包括交付的物件管理介面(LifeCycleManager, 簡稱 LM)、和物件查詢管理介面(QueryManager, 簡稱 QM), 在客戶端則是提供使用服務端的介面(RegistryClient, 簡稱 RC)。ebXML 註冊服務規範 2.1 版說明兩種技術：SOAP 和 ebXML Messaging Service 供參考。註冊服務可用任一種技術、或兩者，製作其概念層介面。客戶端可選用適當的技術使用註冊服務，如圖三所示。



圖三 ebXML 註冊服務架構圖

[資料來源：www.oasis-open.org]

3. LOM 及 ebRIM 的對應

為了讓學習物件達到可重複使用的目的，加入 Metadata 來描述學習物件是必須的步驟，而 SCORM 2004 的教材內容整合模式(CAM)採用 IEEE LTSC LOM v1.0 來描述學習物件。LOM 包含 general、lifeCycle、metaMetadata、technical、educational rights、relation、annotation、classification 等九種類別 XML 元素，每一類別裡面都包含許多子元素，可以填資料值的元素共計有 58 個。LOM 並針對 Content Aggregation、SCO、Asset 等不同層級的學習資源，做出不同的元素要求，限制哪些是必要性元素，哪些是選擇性元素。

由上述可知 LOM 可以填資料值的元素達 58 個之多，相當不利於人們直接手動編輯，即使有程式幫忙提示使用者輸入資料值，編輯好完整的九類 LOM 也將耗去相當多的時間與心力，是不經濟實惠的作法。事實上，LOM 大部分選擇性元素是以不同觀點做學習資源分類或描述兩個學習資源之間的關係，而對於需求不大的元素是可以不使用的，「簡化使用門檻並維持系統價值」才是本研究的精神所在。

重要的是 SCORM 2004 RTE 平台建議的基本需求為全部必要性元素的存在，為了符合 RTE 規範並達到簡化使用門檻的目的，本研究使用全部必要性元素，定義了 Asset-based 素材儲存庫、SCO-based 學習物件儲存庫、Content Aggregation-based 教材儲存庫等三種儲存庫運作模式；另外為了強化系統價值，本研究採用一項選擇性元素作為儲存庫分類用途，此元素為 educational 類別底下的學習資源類型(learningResourceType)元素，內容結構如圖四所示。



圖四 learningResourceType 分類法

以下分別就 Asset-based 素材儲存庫、SCO-based 學習物件儲存庫、Content Aggregation-based 教材儲存庫等三種儲存庫模式說明 Metadata 規劃情形及與註冊物件之間的對應關係：

(1) Asset-based 素材儲存庫

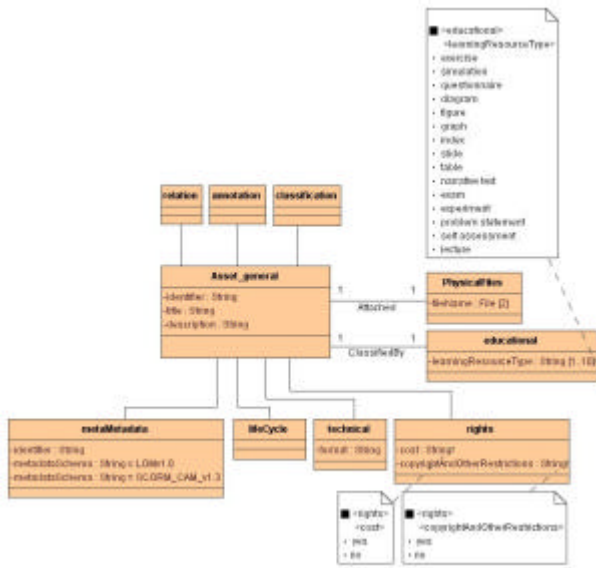
此種儲存庫以 Asset 為儲存單元，其 LOM 位於 imsmanifest.xml 檔 <resource> 或 <file> 元素的下一層，若位於 <resource> 下一層表示此 Asset 可能由多個檔案組成但具備共同特性，故合在同一個 <resource> 底下只以一份 LOM 資訊(Asset Meta-data) 加以描述；若位於 <file> 下一層表示此 Asset 只能為單一素材檔案(如 *.jpg *.gif *.wav)，並以一份 LOM 資訊加以描述。此外，LOM 可直接內嵌於 imsmanifest.xml 之中或是獨立成單一 XML 檔。為了增加使用上的彈性，本研究只針對單一素材檔案並採用外部獨立的 XML 檔來放置 LOM 資訊，如此一來，描述素材的 LOM 檔與描述課程架構的 imsmanifest.xml 檔將是分離的，有助於素材重新使用，亦有助於管理。而儲存庫將依分類結構儲存素材(如 *.jpg、*.gif、*.wav)與對應 Metadata 之 ZIP 壓縮檔。

Asset-based 素材儲存庫將採用所有 Asset 層級的必要性元素(Mandatory, 標示為 M)，並以選擇性元素(Optional, 標示為 O) <learningResourceType> 做為分類用途，用到的元素及範例如表一所示：

表一 Asset Meta-data 元素需求表

元素類別	子元素	條件	範例
1 general	1.1 identifier (1.1.2 entry)	M	c09ee0ac-8c3a-436e-8e79-e6df560f5006
1 general	1.2 title	M	鑰匙圖檔
1 general	1.4 description	M	大小為 32x32pixel 的鑰匙圖示檔
3 metaMetadata	3.1 identifier (3.1.2 entry)	M	d1f8e1d0-6475-4a64-b3b5-b0f313f4a46a
3 metaMetadata	3.3 metadataSchema	M	LOMv1.0, SCORM_CAM_v1.3(固定值)
4 technical	4.1 format	M	image/gif
6 rights	6.1 cost	M	no
6 rights	6.2 copyrightAndOtherRestrictions	M	no
5 educational	5.2 learningResourceType	O	graph

將 Asset Meta-data 元素需求轉換為 ebRIM 的 RegistryObject 之後，可產生 Asset-RO 註冊物件，如圖五所示，對應關係則如表二。



圖五 Asset-RO 註冊物件 Class Diagram

表二 Asset-RO 對照表

LOM / File	個數	ebRIM
general.identifier	1	RO.Slot
general.title	1	RO.Name
general.description	1	RO.Description
metaMetadata.identifier	1	RO.Slot
metaMetadata.metadataSchema	2...10	RO.Slot
technical.format	1	RO.Slot
rights.cost	1	RO.Slot
rights.copyrightAndOtherRestrictions	1	RO.Slot
educational.learningResourceType	1...10	Classification
PhysicalFiles	1	EO.RepositoryItem

general.title 及 general.description 直接採用 RegistryObject 預設屬性 Name 及 Description 來對應儲存。

general.identifier、metaMetadata.identifier、metaMetadata.metadataSchema、technical.format、rights.cost、rights.copyrightAndOtherRestrictions 則以 RegistryObject 擴充屬性 Slot 來對應儲存。

educational.learningResourceType 利用 ebRIM 的分類功能 Classification 來對應儲存分類架構 (ClassificationScheme) 底下的分類節點 (ClassificationNode)。

為了儲存實體檔案，我們系統實作上是採用 RegistryObject 的繼承類別 ExtrinsicObject，因為 ExtrinsicObject 多了 RepositoryItem，能夠指定實體檔案。因此 PhysicalFiles.fileName 是將 Asset 素材檔、Asset LOM 檔壓縮成單一 ZIP 檔 (PhysicalFiles)，再存入 RegistryObject 的繼承類別 ExtrinsicObject 所提供的 RepositoryItem 項目中。

(2) SCO-based 學習物件儲存庫

此種儲存庫以 SCO 為儲存單元，其 LOM 位於 imsmanifest.xml 檔 <resource> 元素的下一層。通常此種 SCO 為網頁形式，其中包括一個起始頁面、有連

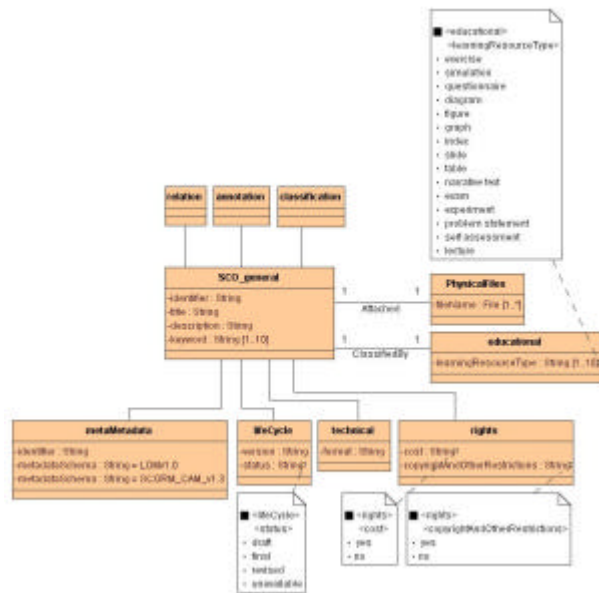
結關係的頁面以及網頁中用到的所有資源如圖片、聲音等。而儲存庫將依分類結構儲存單一 SCO 之 Resource Package (*.zip)，裡面除了網頁及相關資源檔案之外，亦包含 SCO Metadata 及 Resource Manifest 兩份 XML 檔，以及與平台溝通用的 JavaScript 檔、驗證課程包裝用的 XSD 檔等。

SCO-based 素材儲存庫將採用所有 SCO 層級的必要性元素，並以選擇性元素 learningResourceType 做為分類用途，用到的元素及範例如表三所示：

表三 SCO Meta-data 元素需求表

元素類別	子元素	條件	範例
1 general	1.1 identifier (1.1.2 entry)	M	c791b151-e1b9-4aec-82ef-f24a6ef401e1
1 general	1.2 title	M	XML 技術平台
1 general	1.4 description	M	說明 XML 技術核心, XML 視覺呈現, XML 資料處理等項目。
1 general	1.5 keyword	M	XML, XML 技術架構
2 lifeCycle	2.1 version	M	1.0
2 lifeCycle	2.2 status	M	final
3 metaMetadata	3.1 identifier (3.1.2 entry)	M	22efce52-fc65-4ac1-ba21-8ed2d3b0ace1
3 metaMetadata	3.3 metadataSchema	M	LOMv1.0, SCORM_CAM_v1.3(固定值)
4 technical	4.1 format	M	text/html
6 rights	6.1 cost	M	no
6 rights	6.2 copyrightAndOtherRestrictions	M	no
5 educational	5.2 learningResourceType	O	'figure', 'narrative text'

將 SCO Meta-data 元素需求轉換為 ebRIM 的 RegistryObject 之後，可產生 SCO-RO 註冊物件，如圖六所示，對應關係則如表四。



圖六 SCO-RO 註冊物件 Class Diagram

general.title 及 general.description 直接採用 RegistryObject 預設屬性 Name 及 Description 來對應儲存。

表四 SCO-RO 對照表

LOM / File	個數	ebRIM
general.identifier	1	RO.Slot
general.title	1	RO.Name
general.description	1	RO.Description
general.keyword	1...10	RO.Slot
lifeCycle.version	1	RO.Slot
lifeCycle.status	1	RO.Slot
metaMetadata.identifier	1	RO.Slot
metaMetadata.metadataSchema	2...10	RO.Slot
technical.format	1	RO.Slot
rights.cost	1	RO.Slot
rights.copyrightAndOtherRestrictions	1	RO.Slot
educational.learningResourceType	1...10	Classification
PhysicalFiles	1	EO.RepositoryItem

general.identifier、general.keyword、lifeCycle.version、lifeCycle.status、metaMetadata.identifier、metaMetadata.metadataSchema、technical.format、rights.cost、rights.copyrightAndOtherRestrictions 則以 RegistryObject 擴充屬性 Slot 來對應儲存。

educational.learningResourceType 利用 ebRIM 的分類功能 Classification 來對應儲存分類架構 (ClassificationScheme) 底下的分類節點 (ClassificationNode)。

為了儲存實體檔案，我們系統實作上是採用 RegistryObject 的繼承類別 ExtrinsicObject，因為 ExtrinsicObject 多了 RepositoryItem，能夠指定實體檔案。因此 PhysicalFiles.fileName 是將 SCO 網頁及相關資源檔案、SCO LOM 檔、SCO Resource Manifest 檔，以及與平台溝通用的 JavaScript 檔、驗證課程包裝用的 XSD 檔壓縮成單一 ZIP 檔 (PhysicalFiles)，再存入 RegistryObject 的繼承類別 ExtrinsicObject 所提供的 RepositoryItem 項目中。

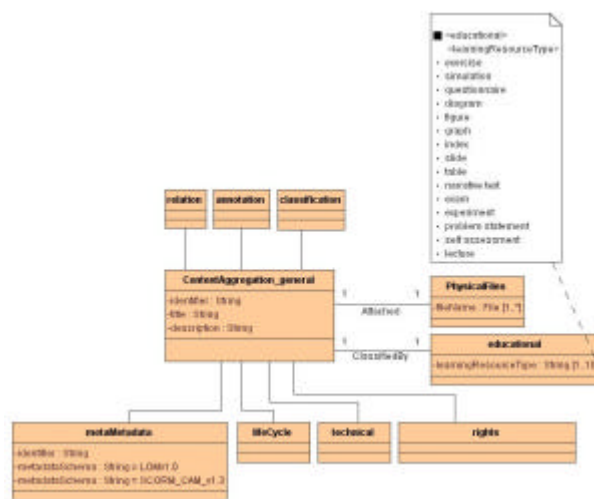
(3) Content Aggregation-based 教材儲存庫

此種儲存庫以 Content Aggregation (PIF) 為儲存單元，其 LOM 位於 imsmanifest.xml 檔 <manifest> 元素的下一層。整個 PIF 檔由學習清單檔 imsmanifest.xml 加以描述，Manifest 部分包括 LOM 詮釋資料 (Content Aggregation Meta-data) 課程架構 (Organizations)、課程架構對應的資源位置 (Resources)。Physical Files 則由 Asset、SCO 實體檔案組成，除此之外，PIF 檔還包括控制檔，如與平台溝通用的 JavaScript 檔及驗證課程包裝用的 XSD 檔。而儲存庫將依分類結構儲存具課程架構之 Content Aggregation Package (*.zip)。

表五 Content Aggregation Meta-data 元素需求表

元素類別	子元素	條件	範例
1 general	1.1 identifier (1.1.2 entry)	O	2a64d4d0-3cbd-40c3-af19-f127ef5edac f
1 general	1.2 title	O	XML 課程-第一章 XML 的時代
1 general	1.4 description	O	Introduction to XML
3 metaMetadata	3.1 identifier (3.1.2 entry)	O	e71a9bf-9337-4902-bef8-960d9b382ff a
3 metaMetadata	3.3 metadataSchema	M	LOMv1.0, SCORM_CAM_v1.3(固定值)
5 educational	5.2 learningResourceType	O	'figure', 'narrative text'

將 Content Aggregation Meta-data 元素需求轉換為 ebRIM 的 RegistryObject 之後，可產生 Content Aggregation-RO 註冊物件，如圖七所示，對應關係則如表六。



圖七 Content Aggregation-RO 註冊物件 Class Diagram

表六 Content Aggregation-RO 對照表

LOM / File	個數	ebRIM
general.identifier	1	RO.Slot
general.title	1	RO.Name
general.description	1	RO.Description
metaMetadata.identifier	1	RO.Slot
metaMetadata.metadataSchema	2...10	RO.Slot
educational.learningResourceType	1...10	Classification
PhysicalFiles	1	EO.RepositoryItem

general.title 及 general.description 直接採用 RegistryObject 預設屬性 Name 及 Description 來對應儲存。

general.identifier、metaMetadata.identifier、metaMetadata.metadataSchema 則以 RegistryObject 擴充屬性 Slot 來對應儲存。

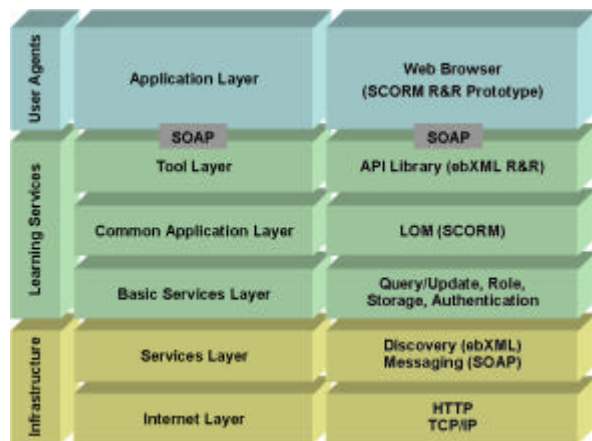
educational.learningResourceType 利用 ebRIM 的分類功能 Classification 來對應儲存分類架構 (ClassificationScheme) 底下的分類節點 (ClassificationNode)。

為了儲存實體檔案，我們系統實作上是採用 RegistryObject 的繼承類別 ExtrinsicObject，因為 ExtrinsicObject 多了 RepositoryItem，能夠指定實體檔案。因此 PhysicalFiles.fileName 是將整個 PIF 相關資源檔案、與平台溝通用的 JavaScript 檔及課程包裝驗證用的 XSD 檔、Content Aggregation LOM 檔、Content Aggregation Manifest 檔壓縮成單一 ZIP 檔 (PhysicalFiles)，再存入 RegistryObject 的繼承類別 ExtrinsicObject 所提供的 RepositoryItem 項目中。

4. SCORM 儲存庫系統

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

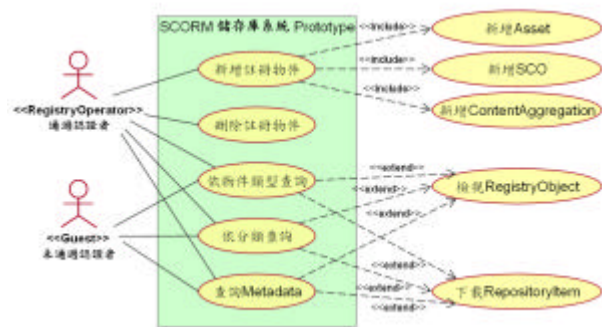
(Learning Services Architecture) [8]，對應 User Agents、Learning Services、Infrastructure 三層 Layers 特性，引入 ebXML R&R 的註冊與儲存庫機制，將之應用在數位學習 SCORM 2004 CAM 教材包裝標準上，採用 IEEE LTSC LOM 學習物件詮釋資料的描述能力，建立了 SCORM 儲存庫 Prototype 三層式架構，並已實作出一套「SCORM 2004 儲存庫系統 Prototype」，其整體的架構如圖八所示。



圖八 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 [3] 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，因而建立出「SCORM 2004 儲存庫系統 Prototype」。

本系統直接採用 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 SCQ、新增 Content Aggregation 等三項子功能。圖九說明上述情境。

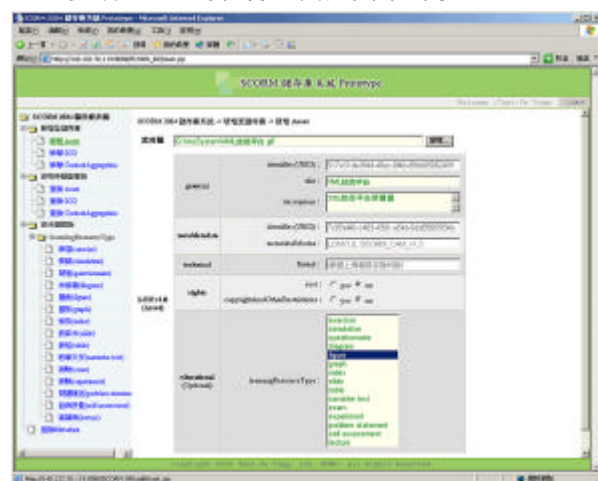


圖九 SCORM 儲存庫系統 Use Case Diagram

以下開始介紹本儲存庫系統的各項功能與執行畫面：

◇ 新增 Asset 功能：

新增 Asset 的執行畫面如圖十所示：



圖十 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。

rights 類別底下的 cost、copyrightAndOtherRestrictions 只能使用 IEEE 規定的 yes 或 no 兩項辭彙，亦有預設值。

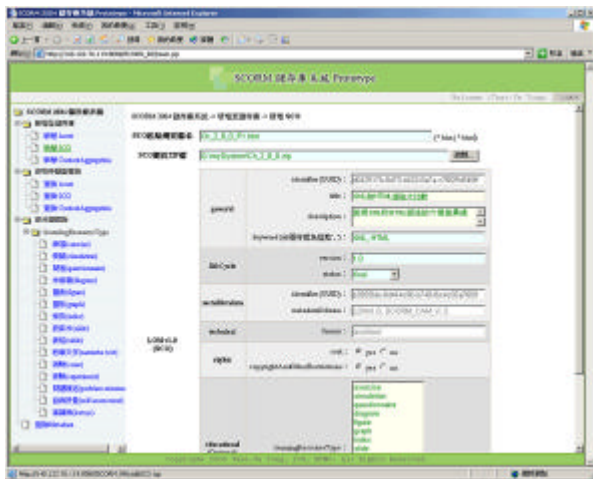
換句話說，我們只須選擇欲上傳的 Asset 素材檔，例如 XML 技術平台.gif，編寫淺黃底色的文字框項目：title、description，並於分類選單中選擇 learningResourceType，即完成一項 Asset LOM 的編

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。

◇ 新增 SCO 功能：

新增 SCO 的執行畫面如圖十一所示：



圖十一 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。

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、description，以及淺藍底色的文字框項目：keyword，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，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：

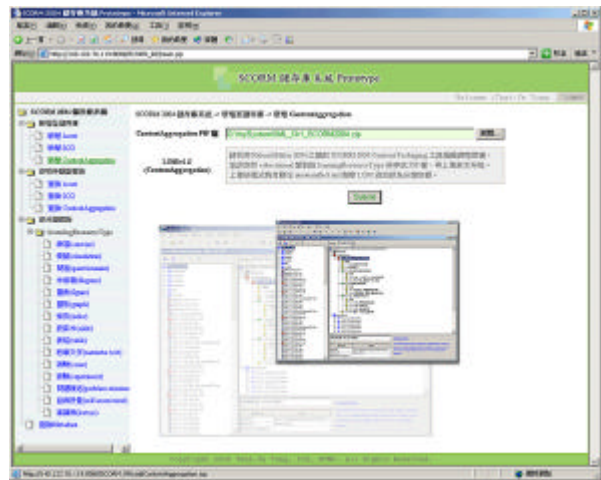
1. 接收上傳的 SCO 網頁壓縮檔。
2. 加入 APIWrapper.js 及 SCOFUNCTIONS.js 檔，產生新的 SCO 起始網頁。
3. 依據使用者填寫資訊，產生 SCO Metadata 檔。
4. 分析 SCO 網頁壓縮檔內容，產生 SCO Resource Manifest 檔。
5. 將步驟 2, 3, 4 產生的檔案，加入上傳的 SCO 網頁壓縮檔。
6. 將 SCO Metadata 對應成 SCO-RO 後註冊進儲存庫，並做好分類。

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) [9]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 SCOFUNCTIONS 兩支 JavaScript，確定此 SCO 能夠被 SCORM RTE 正確啟動(Launch)並紀錄瀏覽時間。

◇ 新增 Content Aggregation 功能：

新增 Content Aggregation 的執行畫面如圖十二所示：



圖十二 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，使用者利用 Reload Editor 2004 [10]之類的 SCORM 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。

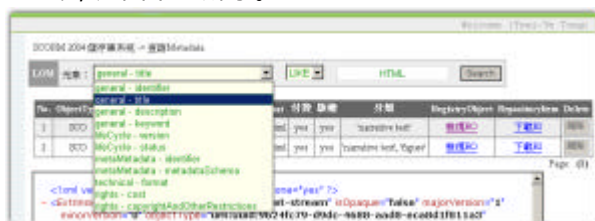
若使用者將儲存庫內的 SCOs 以 Reload Editor 2004 [10]編輯課程架構，再匯出成 PIF 檔，將能夠順利通過 ADL SCORM 2004 Conformance Test Suite Version 1.3.3 (Self Test) [9]官方相容性測試套件中的 Content Package Conformance Test，說明了其 PIF 檔能夠執行在符合 SCORM 2004 的平台上。

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。

前面介紹了三種新增功能，分別針對 Asset、SCQ Content Aggregation 三種顆粒度的學習物件，有不同的使用情境，而最終目的都是為了將學習物件本身及學習物件的 Metadata 一同註冊/儲存進 SCORM 儲存庫系統。下文則開始介紹本系統提供的三種查詢模式：依物件類型查詢、依分類查詢以及查詢 Metadata，茲分述如下。

- ◇ 依物件類型查詢共有 Asset、SCO、Content Aggregation 三種類型。
- ◇ 依分類查詢則是從 LOM 中的 learningResourceType 元素做選擇。
- ◇ 查詢 Metadata 則屬於自由度最高的進階查詢方式，如圖十三所示。



圖十三 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。

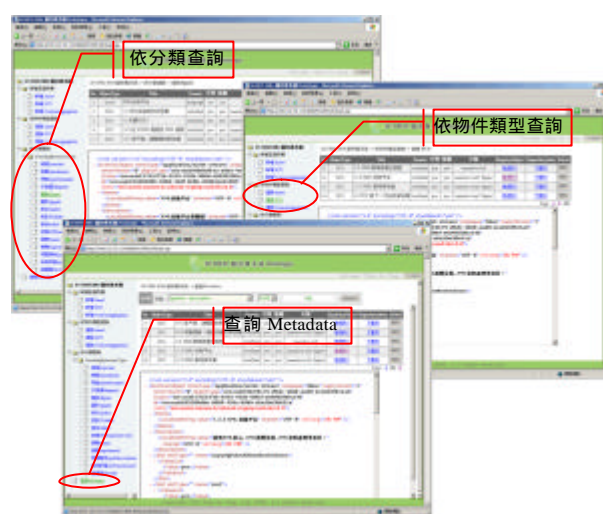
由於 LOM 資訊都已經對應成 ebRIM 格式，對應關係如表七所示。系統的「依分類查詢」功能實作 ebXML R&R 中的 Classification Searching，可查詢 educational.learningResourceType 元素；「查詢 Metadata」功能則結合 ebXML R&R 中的 RO-Slot Searching、RO-Name Searching 與 RO-Description Searching，透過 RO-Name Searching 可查詢 general.title 元素，透過 RO-Description Searching 可查詢 general.description 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。

表七 LOM 與 ebRIM 對照表

LOM	個數	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

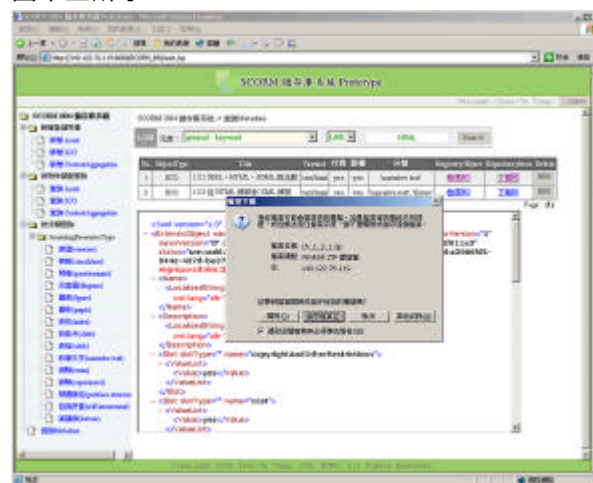
educational.learningResourceType	1...10	Classification	依分類查詢
----------------------------------	--------	----------------	-------

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，畫面如圖十四所示，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。



圖十四 檢視 RegistryObject 畫面

當使用者需要進一步觀看 Asset SCQ Content Aggregation 實際內容時，可以點選下載 RepositoryItem 超連結，以取得實際檔案，畫面如圖十五所示：



圖十五 下載 RepositoryItem 畫面

以上說明為本研究 SCORM 儲存庫系統所規劃的各式功能。

5. 結論與未來發展

本研究針對 Asset、SCO、Content Aggregation 不同層級的學習物件，提出了 Asset-based 素材儲存庫、SCO-based 學習物件儲存庫、Content Aggregation-based 教材儲存庫等三種不同顆粒度之教材儲存庫模式，並在學習物件詮釋資料(LOM)與 ebXML 註冊資訊模型(ebRIM)之間實作 Metadata 的對應轉換，使得我們能夠直覺地查詢學習物件的 Metadata，而精確地找到符合需求的學習物件。除此之外，內容儲存庫標準 ebXML R&R 具有以 XML 為基礎的物件導向資料庫架構，其資料模式的運作、設計概念與傳統關聯式資料庫有所差異，本研究探討學習物件確實可以有效地使用物件導向資料庫，並提供強大的查詢功能。

換句話說，本研究成功地結合內容儲存庫 ebXML R&R 及數位學習 SCORM 2004 兩項標準，並實作出使用 SCORM 2004 標準之教材儲存庫系統，其系統除了具備 ebXML 註冊與儲存庫特有的 (1)開放標準、(2)聯合互動、(3)物件導向資料庫架構等特色之外，並發展出三種物件類型的查詢 (Three kind of RO-ObjectType Searching)、分類系統查詢(Classification Searching)、與註冊物件詮釋資料進階查詢 (包括 RO-Slot Searching、RO-Name Searching 與 RO-Description Searching)等功能。整體研究成果將有助於架構出一有效的數位學習 SCORM 教材之內容管理機制。

然而，為了適應分散式網路環境，(1)儲存庫改用 Global ID；(2)儲存庫增加 Federation 功能，包括聯合註冊機制(Registry Federation)以及聯合查詢 (Federated Query)能力；(3)儲存庫參考 CORDRA [11] 架構等項目，是本研究 SCORM 儲存庫系統未來值得且應該發展的方向。

6. 參考文獻

- [1] ADL (July 2004), “Sharable Content Object Reference Model (SCORM) 2004 2nd Edition Overview”, <http://www.adlnet.org/>
- [2] ADL (July 2004), “Sharable Content Object Reference Model (SCORM) Content Aggregation Model (CAM) Version 1.3.1”, <http://www.adlnet.org/>
- [3] ebxmlrr (June 2004), “ebxmlrr 3.0-alpha2”, OASIS ebXML Registry Reference Implementation Project, <http://ebxmlrr.sourceforge.net/>
- [4] OASIS (June 2002), “OASIS/ebXML Registry Information Model (ebRIM) Specification v2.1 Committee Approved”, OASIS ebXML Registry Technical Committee, <http://www.oasis-open.org/committees/regrep>
- [5] OASIS (June 2002), “OASIS/ebXML Registry Services (ebRS) Specification v2.1 Committee Approved”, OASIS ebXML Registry Technical Committee, <http://www.oasis-open.org/committees/regrep>
- [6] ADL (July 2004), “Sharable Content Object Reference Model (SCORM) Run-Time Environment (RTE) Version 1.3.1”, <http://www.adlnet.org/>
- [7] ADL (July 2004), “Sharable Content Object Reference Model (SCORM) Sequencing and Navigation (SN) Version 1.3.1”, <http://www.adlnet.org/>
- [8] CMU (November 2001), “Learning Services Architecture”, Learning Systems Architecture Lab, Carnegie Mellon University, <http://www.lsal.cmu.edu/lisal/expertise/technologies/learningservices/lisa.pdf>
- [9] ADL (May 2005), “SCORM 2004 Conformance Test Suite Version 1.3.3 (Self Test)”, <http://www.adlnet.org/scorm/history/2004/sets.cfm>
- [10] CMU (February 2005), “Reload Editor 2004 v1.3.2 Beta2”, Learning Systems Architecture Lab, Carnegie Mellon University, <http://www.lsal.cmu.edu/adl/scorm/tools/reload/index.html>
- [11] CMU (February 2004), “CORDRA”, Learning Systems Architecture Lab, Carnegie Mellon University, <http://www.lsal.cmu.edu/lisal/expertise/projects/cordra/index.html>