

第二章 文獻探討

本章為相關文獻的探討與整理。第一節介紹數位學習標準 SCORM 2004 規範與架構。第二節介紹與內容儲存庫標準 ebXML Registry&Repository 有關的註冊模型、註冊服務。第三節介紹美國卡內基·梅隆大學，學習系統架構實驗室(LSAL)的學習服務架構(Learning Services Architecture)。

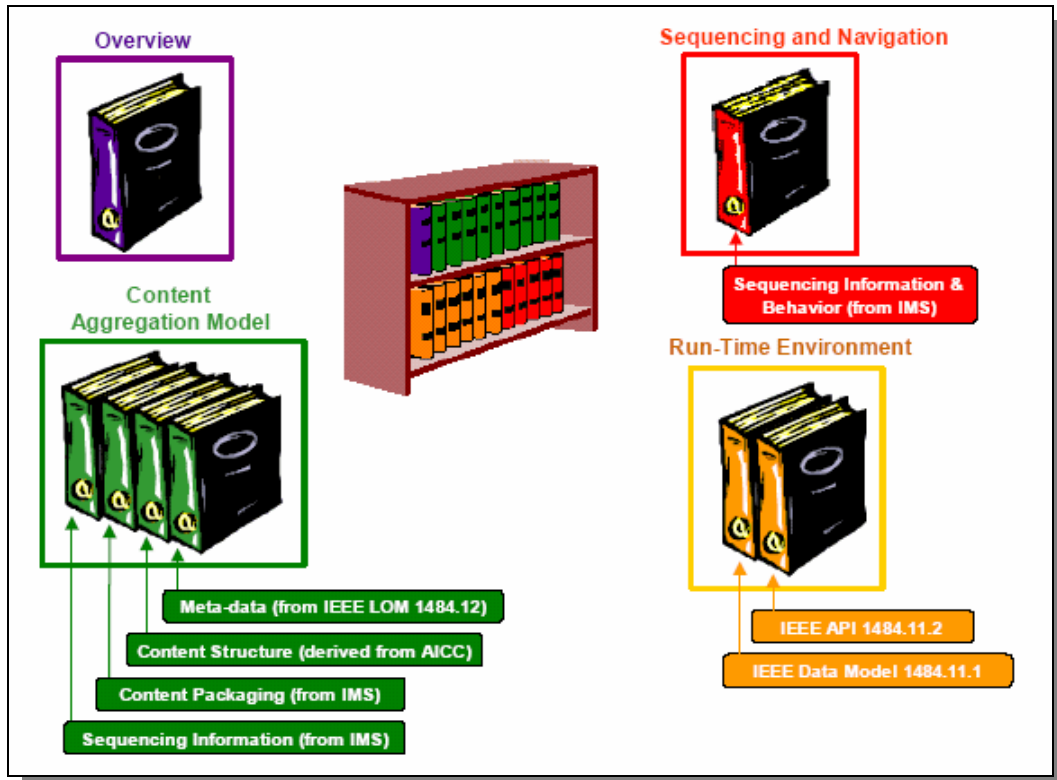
第一節 數位學習標準 SCORM 2004

為了減少教材開發的時程與成本，並使各教材在不同的學習管理系統(Learning Management Systems，簡稱 LMSs)中能夠重複使用、流通自如。在 1997 年底，美國國防部(DoD) 主導「先進分散學習計畫」(Advanced Distributed Learning，ADL)。此計畫集合教材開發廠商、使用者與 IMS【20】、AICC【13】、IEEE【19】等標準化推動單位，共同彙整美國各界過去在教材標準上的努力成果，以延伸式標記語言(eXtensible Markup Language，簡稱 XML)為發展基礎，進而研訂出整合性的規範，稱為 SCORM (Shareable Content Object Reference Model，共享教材元件參考模式)，為國際上最廣為採用的 e-Learning 標準。其內容涵蓋教材開發、包裝、交流等主題。【1】

2004 年 1 月 30 日，ADL 推出了新版本 SCORM 2004 (亦稱為 SCORM 1.3)，最大的變革在於加入了序列及導覽(Sequencing and Navigation，簡稱 SN)的規格書，透過串列語法來串連各個教材單元，達到適性化學習的目標。2004 年 7 月 22 日修正部份小錯誤，訂為 SCORM 2004 2nd Edition (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：The SCORM Overview【7】、The SCORM Content Aggregation Model (CAM)【8】、The SCORM Run-Time Environment (RTE)【9】和 The SCORM Sequencing and Navigation (SN)【10】，其架構如【圖 2-1】所示。



【圖 2-1】 SCORM 2004 規格書架構圖 [資料來源：www.adlnet.org]

概要(Overview)，主要是介紹 SCORM 的來龍去脈以及相關資訊。教材內容整合模式(CAM)，主要是介紹教材元件的包裝標準。執行環境(RTE)，主要目的是讓學習平台與教材之間有統一的溝通介面。順序與導覽(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。後面針對教材內容整合模式、執行環境及順序與導覽作詳細說明。

2.1.1 SCORM 教材內容整合模式

SCORM 的教材內容整合模式(Content Aggregation Model, 簡稱 CAM), 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(Packaging)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。

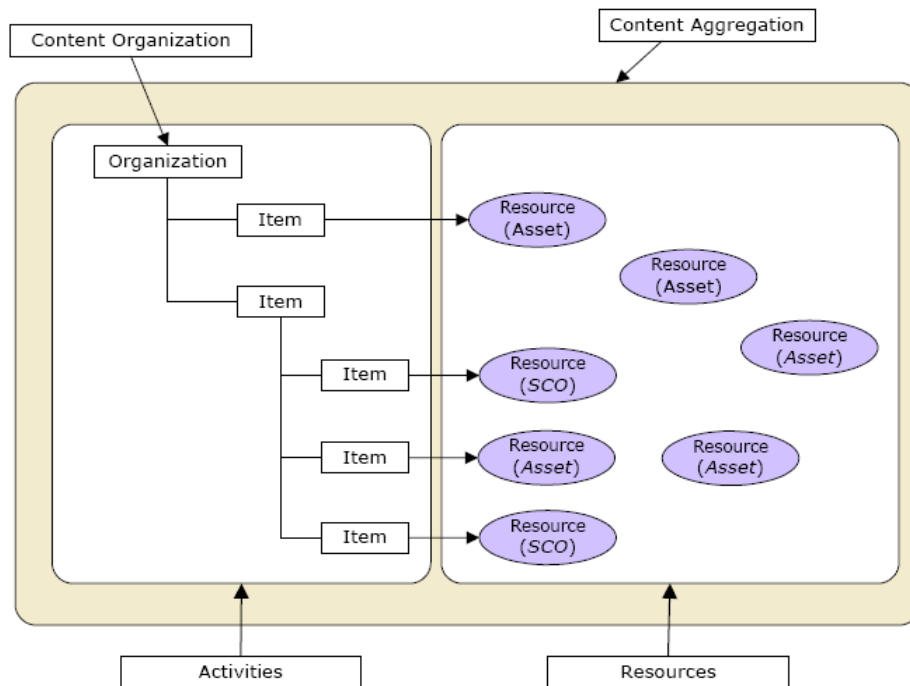
SCORM 2004 將教材內容定義為素材(Asset)、共享內容元件(SCO, Sharable Content Object)與內容組織(Content Organization)三種, 說明如下:

- ◇ 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)，其架構如【圖 2-2】所示。



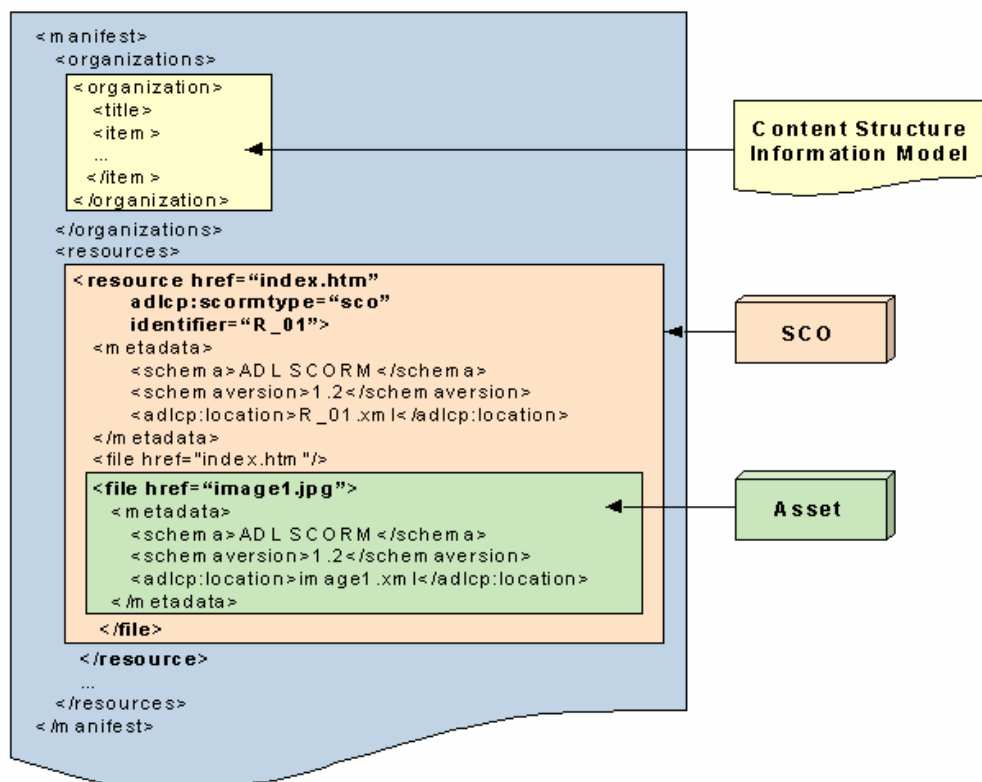
【圖 2-2】 Content Aggregation 示意圖 [資料來源：www.adlnet.org]

(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 imsmanifest.xml 之中或是獨立成單一 XML 檔，例如由【圖 2-3】可看出 SCO 及 Asset 分別對應的 LOM 是利用<adlcp:location>標籤語法獨立儲存於另一個 XML 檔案中。學習清單檔共計五個位置能夠以 LOM 加以描述，分別是：

- ✧ Content Aggregation Meta-data：位於 manifest 元素的下一層。
- ✧ Content Organization Meta-data：位於 organization 元素的下一層。
- ✧ Activity Meta-data：位於 item 元素的下一層。
- ✧ SCO Meta-data：位於 resource 元素的下一層，如【圖 2-3】所示。
- ✧ Asset Meta-data：位於 file 或 resource 元素的下一層，【圖 2-3】為 file 的例子。



【圖 2-3】imsmanifest.xml 示意圖 [資料來源：www.adlnet.org]

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：

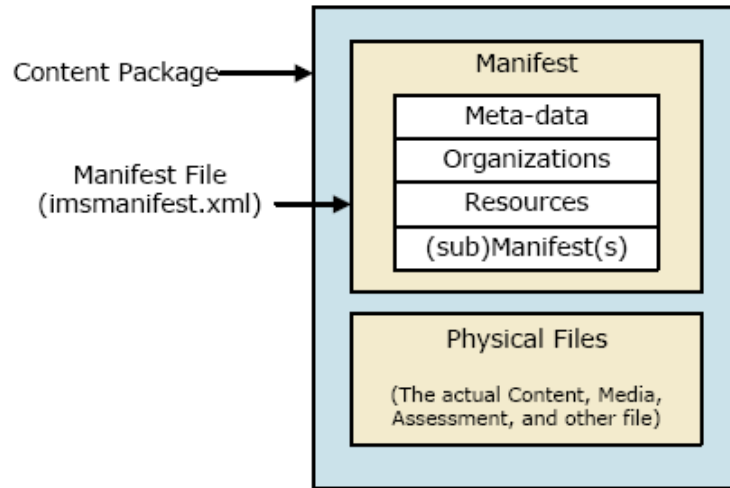
1. 一般(general)：描述學習資源一般性資訊，如標題、使用語言、關鍵字、涵蓋範圍、結構、整合程度等。
2. 生命週期(lifeCycle)：描述學習資源的版本、狀態和提供者的相關資料。
3. 描述資料(metaMetadata)：描述學習資源 Metadata 的提供者及其參考的 Schema 版本。
4. 技術(technical)：描述學習資源技術需求方面的訊息，如格式、大小、位置、執行環境需求、安裝教學、其他軟硬體限制、使用時間長度等。
5. 教育(educational)：描述學習資源在教育及教學方面的特性，如互動類型、學習資源類型、互動程度、預期使用者、使用情境、使用年齡層、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(例如：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)，需符合 PKZip v2.04g (.zip)格式，其包含以下兩種類型：

- ◇ Resource Package：不包含課程章節架構(Content Organization)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(Content Organization)的 PIF，可直接匯入學習平台中使用或是與他人交換。

內容包裝提供一種方法，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。而內容包裝分兩個主要部份：學習清單(Manifest)與實體檔案(Physical files)，其架構如【圖 2-4】所示。



【圖 2-4】 Content Package 概念圖 [資料來源：www.adlnet.org]

學習清單必須有詮釋資料、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(sequencing)；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。

實體檔案則是表示在 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)【3】。

2.1.2 SCORM 執行環境

SCORM 執行環境(Run-Time Environment，簡稱 RTE)，規定 SCORM 教材如何與 LMS 溝通的環境，提供了八個標準 API 作為教材與平台之間資料溝通的橋樑、並追蹤紀錄學習者的學習過程。傳統的網路教學中，不同的教學系統通常採用各自定義的資料格式和資料處理方式，造成不同平台間教材交換時的許多問題，SCORM 針對此點，特別定義了執行環境的規範，內容包含三個部分：啟動(Launch)、應用程式介面(Application Program Interface，API)和資料模型(Data Model)，茲分述如下：

(1) Launch

定義 LMS 啟動學習資源的方式，而這個方式其實就是將學習者瀏覽器網址列的網址指向學習元件的位置。SCORM 對於 LMS 啟動 Asset 與 SCO 時，存在不同的要求與限制，對於 Asset 類型的學習元件而言，規定 LMS 必須使用 HTTP 的通訊協定啟動 Asset。但是對於 SCO 類型的學習元件而言，除了規定 LMS 必須使用 HTTP 的通訊協定外，還存在其他的限制，包括：

- ◇ 當學習者登入 LMS，並且進入某課程時，學習者同一時間只能進行一個 SCO 的學習活動，不能同時進行數個 SCO 的學習活動。
- ◇ 只有 LMS 可以啟動 SCO，SCO 不能啟動其它的 SCO。

(2) API

定義 SCO 與 LMS 溝通的方式，也就是定義 SCO 取得與設定 LMS 變數的方式。API 就是事先定義好的功能函數，SCORM 2004 定義八個 API，達成 SCO 與 LMS 溝通時所需的各項功能，這些 API 分為工作階段函數(Session Methods)、資料傳輸函數(Data-transfer Methods)與支援函數(Support Methods)三種類型，以下說明各個 API 的功能，【圖 2-5】說明 SCO 與 LMS 開始與結束溝通階段時 API 呼叫情形。

◇ 工作階段函數：

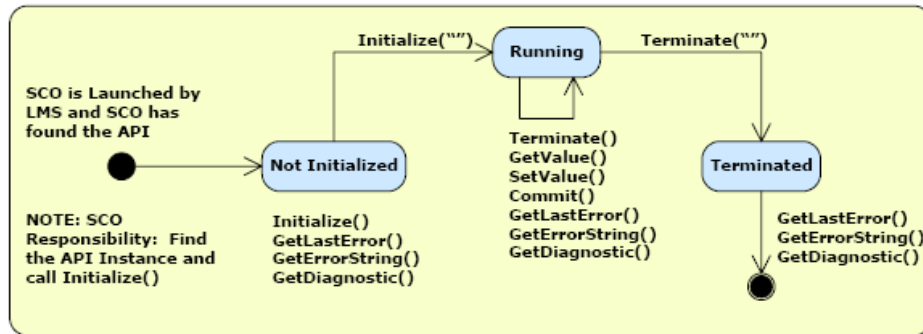
1. Initialize：建立 SCO 與 LMS 之間的資料傳輸通道。
2. Terminate：結束 SCO 與 LMS 之間的資料傳輸通道。

◇ 資料傳輸函數：

3. GetValue：取得學習者之學習資訊。
4. SetValue：儲存學習者之學習資訊。
5. Commit：將所有暫存學習資訊全數儲存至 LMS。

◇ 支援函數：

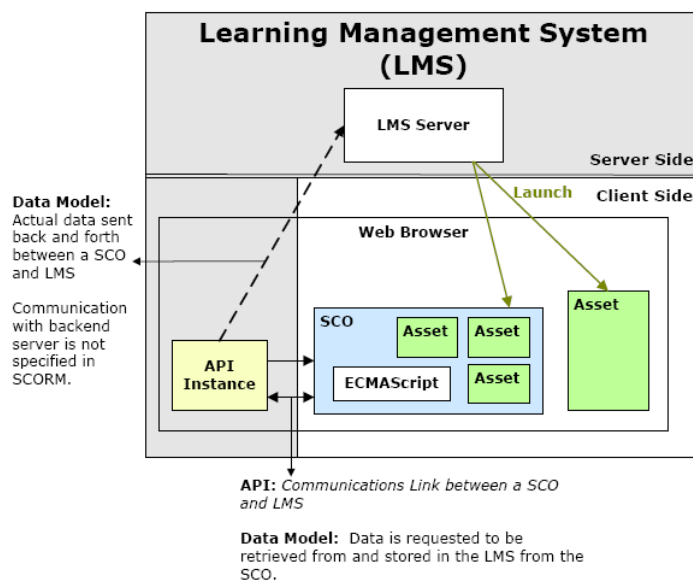
6. GetLastError：取得上一次執行的錯誤代碼。
7. GetErrorString：取得錯誤代碼的解釋。
8. GetDiagnostic：取得錯誤的處理方式。



【圖 2-5】 SCORM 執行環境 API 狀態圖 [資料來源：www.adlnet.org]

(3) Data Model

定義教材與平台溝通的辭彙和語言，API 相當於函數，Data Model 則相當於參數。SCORM 2004 採用 IEEE 1484.11.1 Draft 5 Draft Standard for Learning Technology – Data Model for Content Object Communication. 作為資料模型標準，SCOs 可以利用上述 API 呼叫 Data Model 來與 LMS 作溝通，整個執行環境 Server Side 與 Client Side 互動的關係如【圖 2-6】所示。



【圖 2-6】 SCORM 執行環境架構圖 [資料來源：www.adlnet.org]

2.1.3 SCORM 順序及導覽

順序及導覽(Sequencing and Navigation，簡稱 SN)，提供 Sequencing Rules 使得獨立的學習元件串連起來，形成不同的學習路徑。讓教學設計師將教學策略透過 Sequencing Rules 融入數位教材設計之中。而 SCORM 2004 教材中包含一個 imsmanifest.xml 檔案，此檔案除了承襲 SCORM 1.2 之課程結構(organizations)、實體檔案位置(resources)兩大類資訊外，增加了 sequencing 資訊，用來描述課程各章節之間的串連關係，共提供了八大類語法，分列如下：**【6】【26】**

1. Control Mode：設定教材章節之流程。
2. Sequencing Rules：設定各個教材章節之條件(Condition)及行為(Action)。
3. Rollup Rules：描述子章節與父章節之學習狀態之間的關係。
4. Auxiliary Resource：教材輔助教材。
5. Objectives：設定章節之學習目標。
6. Limit Conditions：設計章節之閱讀限制條件。
7. Randomization Controls：設定教材章節是否隨機出現。
8. Delivery Controls：設定是否記錄章節之學習狀態。

第二節 ebXML 註冊/儲存庫標準

ebXML (Electronic Business using eXtensible Markup Language)，是一個開放性的電子商務建構標準。主要是結合 XML 的技術優勢與 Internet 的普及和低廉成本等特性，應用至商業領域，使全球任何產業的企業，皆可透過 ebXML 架構達到彼此交流、相互溝通的目的。因此 ebXML 計畫使命，開宗明義即為提供一個開放性的 XML-based 基礎環境，讓全球市場上所有的參與者能以互通、安全及一致性的方式，使用電子商務資訊。【25】從另一方面來說，這個標準使得企業資訊內容，能夠遵循一定的設計架構與使用方法，註冊、儲存與交換企業內的組織知識，達到管理的層次，同樣地，將這套註冊與儲存機制應用到 SCORM 教材管理上亦有絕對的成效。

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。目前 UN/CEFACT 組織是訂定國際電子商務標準 ebXML 相關標準的最高標準組織。UN/CEFACT 組織在西元 2001 年和 OASIS 組織合作訂定完成了 ebXML 標準架構。在西元 2002 年以後這兩個標準組織個別發展 ebXML 標準架構後續的標準。UN/CEFACT 組織的發展重點放在 ebXML 標準架構的上層有關商業作業觀點(Business Operation View，簡稱 BOV)的部份，因而發展出 UMM 方法論。而 OASIS 的發展重點則放

在 ebXML 標準架構的底層有關商業服務觀點(Business Service View，簡稱 BSV)的部份，並與 UDDI 論壇合作，發展服務基礎架構(Service Oriented Architecture，簡稱 SOA)相關的底層標準，並提出 ebXML 註冊資訊模型規範(OASIS/ebXML Registry Information Model Specification，簡稱 ebRIM)【21】【23】以及 ebXML 註冊服務規範(OASIS/ebXML Registry Services Specification，簡稱 ebRS)【22】【24】等電子商務註冊中心的相關標準，其版本演進如下：

◇ ebRIM 註冊資訊模型規範：

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。

◇ ebRS 註冊服務規範：

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。

2.2.1 ebXML 註冊機制與儲存庫

OASIS 組織在發展 ebXML 底層的標準方面，訂定了 ebRIM 和 ebRS 等標準，依據這些標準所發展的內容儲存庫通稱為 ebXML 註冊&儲存庫(ebXML Registry&Repository，簡稱 ebXML R&R)。目前 Sun Microsystems 工程師主持的開放原始碼(Open Source)專案(OASIS ebXML Registry Reference Implementation Project，簡稱 ebxmlrr)【18】，主要依據 OASIS 組織發展的 ebRIM 2.1 和 ebRS 2.1 正式標準，並參考 ebRIM 3.0 和 ebRS 3.0 委員會草案的部分內容，而發展出 ebXML 註冊&儲存庫系統(目前為 OASIS ebxmlrr 3.0-alpha2 版本)，也是目前市面上可免費取得的 ebXML 註冊&儲存庫系統之一。本研究即是採用 ebxmlrr 作為 SCORM 數位教材儲存庫的核心，以下小節就 ebRIM 2.1 和 ebRS 2.1 做說明。

2.2.2 ebXML 註冊資訊模型

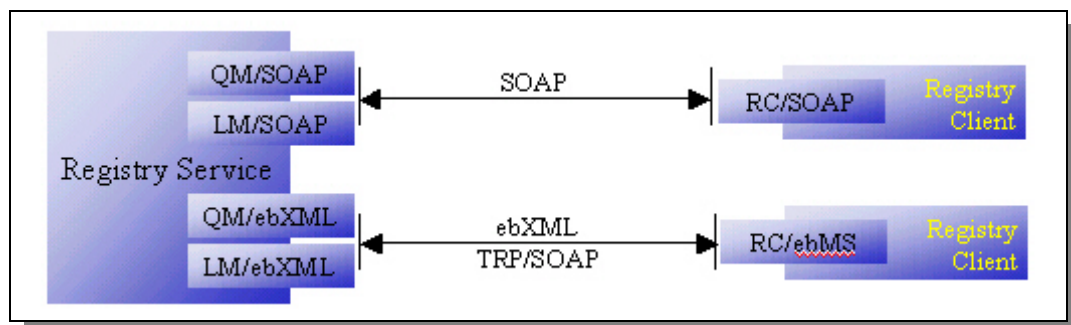
ebXML 註冊服務處的基本資料單元為註冊物件(RegistryObject)，這些註冊物件的格式是依照 OASIS 組織訂定的註冊資訊模型(Registry Information Model)標準來實作的。每一個註冊物件均附有詮釋資料(Metadata)，以供註冊服務處作為分類、及其生命週期內的各種控管之用。

- ◇ 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(AuditableEvent)：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。

2.2.3 ebXML 註冊服務

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)。註冊服務端透過兩個介面：LM、QM，提供儲存庫內物件的列管及使用。服務端介面的定義可分成概念層(abstract level)及實質層(concrete level)，前者定義註冊服務處應該提供的介面、介面的使用方法(method)、及每一方法所需的參數及回應。概念層並不涉及任何的製作技術細節。有關製作技術的細節是在實質層定義，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 2-8】所示。



【圖 2-8】 ebXML 註冊服務架構圖 [資料來源：www.oasis-open.org]

客戶端介面(RC)是用來和服務端建立連結關係，介面內有一些方法讓服務端傳送非同步的回應給客戶。【5】

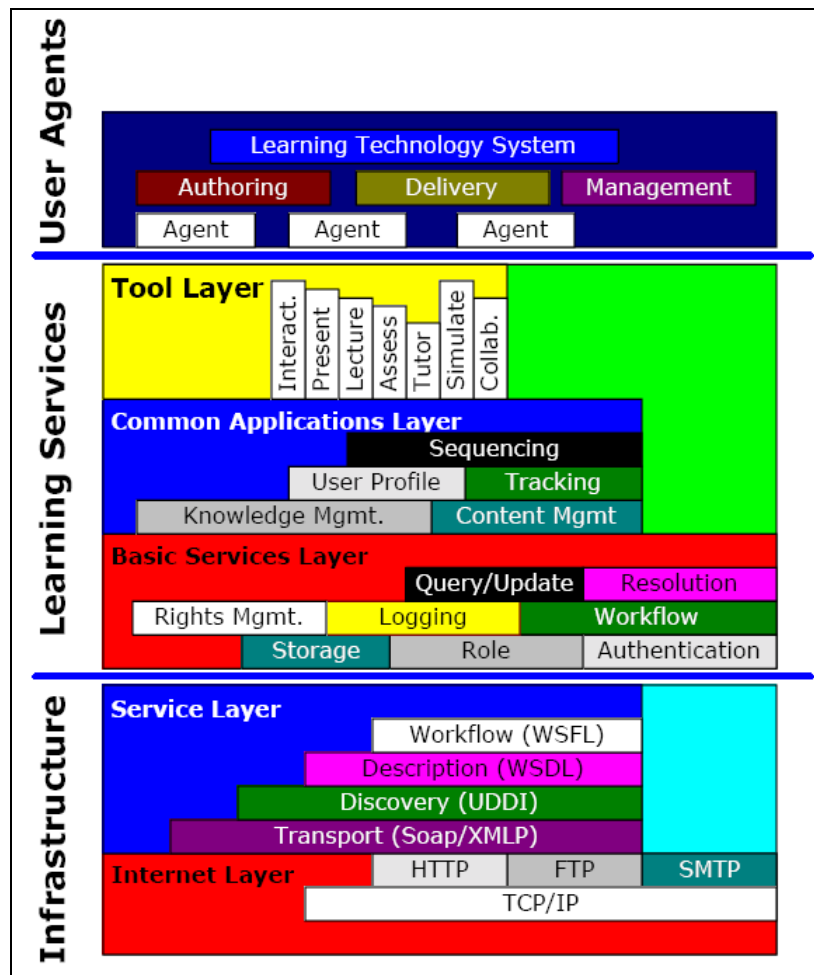
第三節 Learning Services Architecture

美國卡內基·梅隆大學(Carnegie Mellon University)的學習系統架構實驗室(Learning Systems Architecture Lab, 簡稱 LSAL)在學習科技領域中有舉足輕重的地位,也貢獻了很多研究成果給 ADL,如數位學習標準 SCORM (Sharable Courseware Object Reference Model)的一些最佳實作經驗,以及規劃中的儲存庫標準 CORDRA (Content Object Repository Discovery and Registration/Resolution Architecture)。

LSAL 表示:「過去存在的單一獨立性的學習科技系統時代已經結束,與其建立龐大、封閉的系統,不如聚焦在提供一個彈性的架構,使得系統元件及學習內容都存在可交換性(即模組化),並採用開放標準(Open Standards)來達成資料交換與元件整合。因此 LSAL 發展了學習服務架構(Learning Services Architecture)及學習服務堆疊(Learning Services Stack)的概念,作為發展下一代學習科技系統的框架(Framework)。而過程中使用了 Web Services 技術來發展學習服務。【15】

Learning Services Architecture 的理念主要是以 Learning Services Stack 示意圖來呈現, Learning Services Stack 以目前發展 Web Services 及服務基礎系統(service-based systems)的解決方法為基礎,將傳統網路溝通堆疊加以改變、擴充

後精緻化。Learning Services Architecture 依據功能區分為三個主要群組，或稱為服務層(Service Layers)，分別為使用者代理人層(User Agents Layer)、學習服務層(Learning Services Layer)及基礎建設層(Infrastructure Layer)。每個階層分別提供定義良好的介面與其他階層溝通，而實際功能則被清楚地隔離在不同階層，因此每個階層中的服務只能與同一層級中的服務溝通，或利用底層提供的服務。架構如【圖 2-9】所示，功能分述如后。



【圖 2-9】 Learning Services Stack [資料來源：www.lsal.cmu.edu]

2.3.1 使用代理人層

User Agents 即是學習科技系統裡，能夠存取中間層 Learning Services 的使用者介面。User Agents 包括內容的建立、編輯、遞送、管理等功能。具有下列特性：

- ◇ 使用者與系統溝通的媒介。
- ◇ 能被不同身份使用者存取，如學習者、發展人員、管理者、代理人等。
- ◇ 提供觀感(look and feel)。
- ◇ 在 Learning Services 之上建立應用程式。

2.3.2 學習服務層

Learning Services Layer 包括具有特定學習功能的所有服務及元件。並細分為三個子層級：工具層(Tools Layer)、一般應用層(Common Applications Layer)、基本服務層(Basic Services Layer)，分述如下：

- ◇ Tools Layer：提供高階整合服務去建立出一個 User Agents 可使用的公開介面，例如課程遞送、線上家庭教師、模擬訓練、測驗與評量、內容呈現、合作學習、成績與紀錄本、註冊、課程監督、課程管理等功能。

- ◇ Common Applications Layer: 提供 Tools Layer 及 User Agents 需要的標準或時常使用的服務，包括內容選擇與呈現順序、學習者基本資料、使用者追蹤紀錄、學習物件管理、內容管理、報表產生、知識管理等與學習密切相關的功能。
- ◇ Basic Services Layer: 包括核心學習服務及特定學習科技版本下時常使用的系統服務，例如儲存管理、工作流程、版權管理、使用者認證、驗證、系統記錄檔等與學習無直接關聯的功能。

2.3.3 基礎建設層

Infrastructure Layer 是學習服務架構與學習服務堆疊的基礎，組成獨立於學習領域外的服務，Services Layer 提供傳輸能力(SOAP)、發現能力(UDDI 或 ebXML)、描述能力(WSDL)、工作流程(WSFL)等。Internet Layer 提供 HTTP、SMTP、FTP、TCP/IP 等核心網路協定，以建立網際網路上的 Web-based 學習科技系統。