

第四章 ebXML 內容儲存庫規劃

本章第一節將說明 ebXML R&R 內容儲存庫與關聯式資料庫的差異性，及採用 ebXML R&R 的好處。第二、三、四節則就 Asset-based 素材儲存庫、SCO-based 學習物件儲存庫、Content Aggregation-based 教材儲存庫三種模式闡述其與 ebXML R&R 註冊資訊模型的對應關係。

第一節 ebXML 內容儲存庫優勢

時代演進，符合標準(Standard)變的愈來愈重要，e-Learning 中最廣為人們採用的標準莫非是 SCORM，而 SCORM 的核心即是由 XML 組成 Metadata，用來描述學習物件以達到可共享與重複使用等理想，由此可見 Metadata 的重要性；儲存管理教材方面，我們尋找能支援 Metadata 註冊的儲存庫標準為首要目標，其中，ebXML R&R 具備完整架構，不僅能夠註冊與管理 Metadata 訊息，其他諸如使用者認證、整合式 SQL Query 或 Filter Query 查詢功能、儲存物件生命週期管理等，最重要的是這些功能呼叫完全是以 SOAP 訊息服務方式進行。另外，ebXML R&R 亦具備開放標準(Open Standard)、聯合互作(Federation)、物件導向資料庫架構(OO Database Architecture)等特性，使得以此註冊/儲存庫為基礎建構出來的系統更有價值。以下說明這些特性帶來的好處。

(1) Open Standard :

若是採用關聯式資料庫，自行設計資料表、資料欄位，再搭配動態網頁、SOAP 等技術實作出來的儲存庫系統，即使功能再多、再好，仍然屬於封閉式架構，推廣上還是會有一定的困難。而 ebXML R&R 是開放標準，規範由 OASIS 組織維護，並有 Open Source ebxmlrr Project 【18】實作規範中定義的 Data Model 及 API，而形成小有規模的雛型系統，經由修改後即可符合我們的需求，更重要的是符合開放標準方能夠與國際接軌。

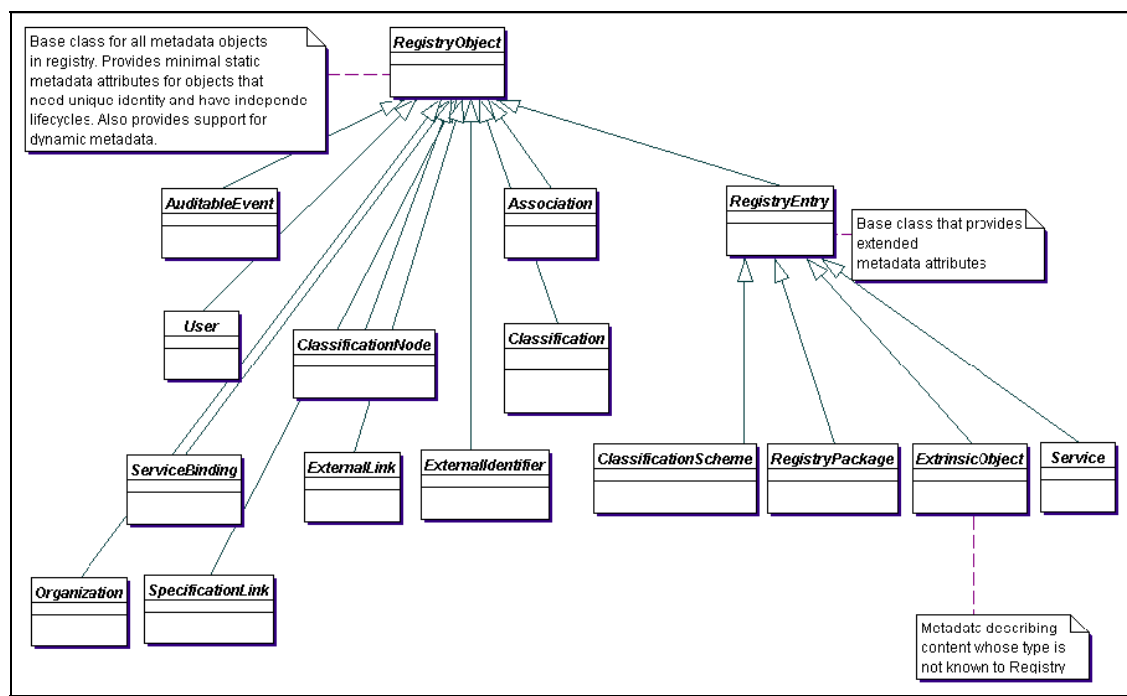
(2) Federation :

透過分散式網路將各地的儲存庫整合在一起，使其具備聯合註冊機制 (Registry Federation) 以及聯合查詢 (Federated Query) 能力統稱為聯合互作 (Federation)。假如 A、B、C 三個儲存庫都具備 Federation 功能，當使用者向儲存庫 A 執行 Federated Query 時，除了對儲存庫 A 本身進行內部搜尋外，並會透過網路向儲存庫 B 與儲存庫 C 進行外部搜尋，再將結果整合在一起，對使用者而言，由單一介面即可完成數個儲存庫查詢，其查詢廣度提升。這裡要特別說明的是，ebRS 3.0 規範草案已定義 Federation，但目前 Open Source ebxmlrr Project : OASIS ebxmlrr 3.0-alpha2 版本 【18】尚未實作此功能，而本研究以 ebxmlrr 為基礎來發展 SCORM 儲存庫系統，因此我們的系統尚未具備 Federation 能力。可以預期的是，等 ebRS 3.0 規範正式釋出後，不論是依據規範裡的 Data Model 及 API 定義

自行開發此功能或是採用 ebxmlrr 後續支援版本，必將享受到 Federation 所帶來的強大功能與便利性。

(3) Object-Oriented Database Architecture, OODB 架構：

ebXML R&R 的底層亦是使用關聯式資料庫，但是使用者面對的資訊模型採用註冊物件(RegistryObject, 簡稱 RO)為基礎類別，再以物件導向方式擴充，如【圖 4-1】所示，其中儲存實體檔案的外部物件(ExtrinsicObject)類別繼承自 RegistryEntry，而 RegistryEntry 又繼承自 RegistryObject。故概念上 ebXML R&R 具備 OODB 架構，程式發展者不需直接操作資料表，只需了解 RegistryObject 繼承關係與 API 使用方式，此種以物件為基礎的操作方式更符合人類思維。



【圖 4-1】 ebXML 資訊模型繼承圖 [資料來源：www.oasis-open.org]

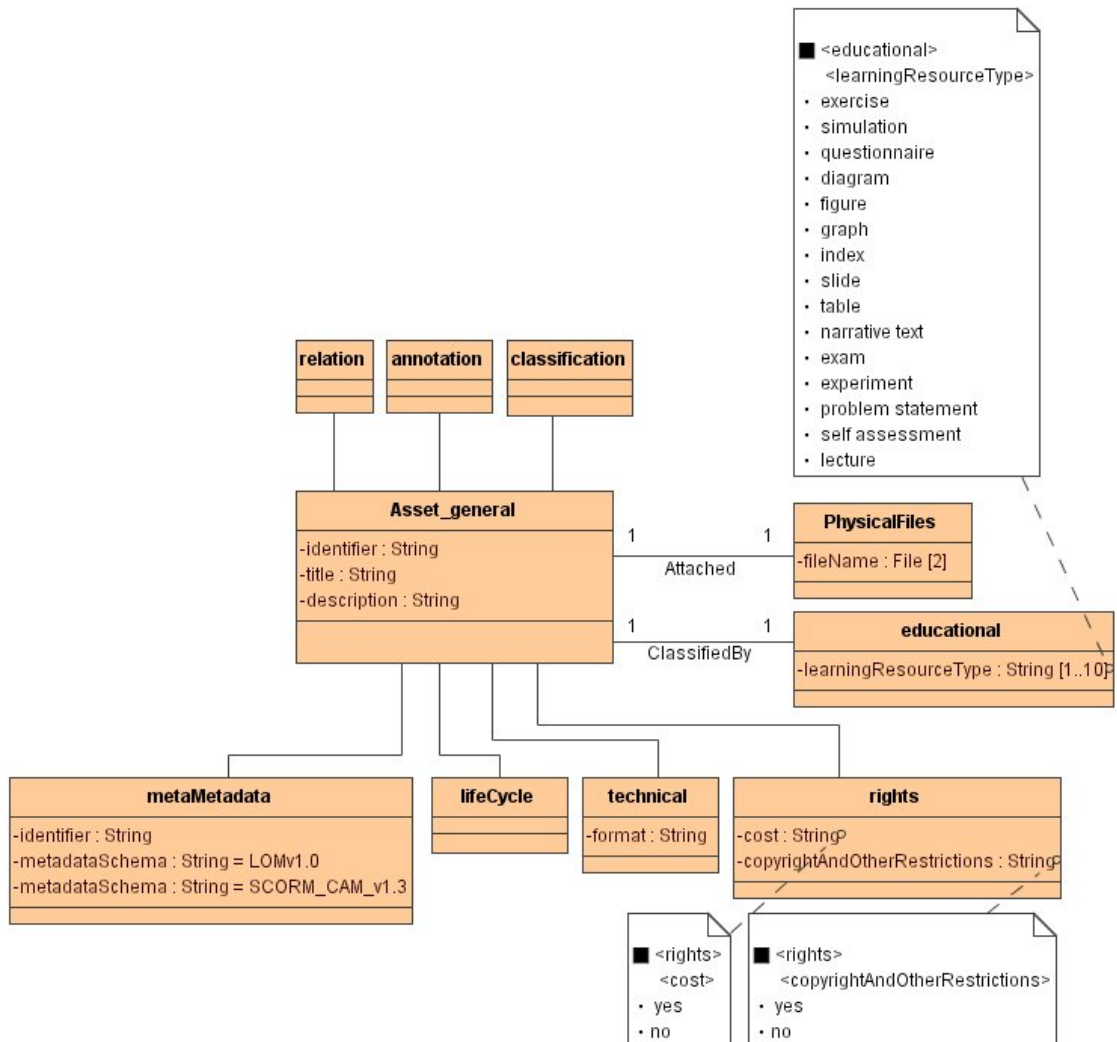
SCORM 學習物件中，不論 Asset、SCO、Content Aggregation，都是由實體檔案與描述用 Metadata 組成，換句話說，SCORM 本身即具備物件概念，故利用 ebXML R&R 的物件特性來儲存 SCORM 學習物件再適合不過了。ebXML R&R 透過【圖 4-1】的 ExtrinsicObject 來管理儲存項目(RepositoryItem)與註冊物件(RegistryObject)，與 SCORM 對應關係如下：

- ✧ RepositoryItem：即是 Asset、SCO、Content Aggregation 實體檔案。
- ✧ RegistryObject：即是用來描述 RepositoryItem 的 Metadata，以 SCORM 2004 來說，就是描述 Asset、SCO、Content Aggregation 用的 IEEE LTSC LOM v1.0。

以下小節分別就 Asset、SCO、Content Aggregation 設計出 Asset-RO 註冊物件、SCO-RO 註冊物件、Content Aggregation-RO 註冊物件，用以說明其 LOM 與 ebRIM (ebXML 註冊資訊模型)之間 Metadata 的對應情形。

第二節 Asset-RO 註冊物件規劃

第二章我們提過 ebRIM (詳見【圖 2-7】), 加上第三章規劃的 Asset Meta-data 元素需求表(詳見【表 3-10】), 我們將前者的 RegistryObject 物件模型概念, 對應到後者的元素需求, 因而產生以下的 Asset-RO 註冊物件, 如【圖 4-2】所示。



【圖 4-2】 Asset-RO 註冊物件 Class Diagram

上圖 Asset Meta-data 與 ebRIM 的對應關係可以整理成下表：

【表 4-1】Asset-RO 對照表

LOM / File	Multiplicity	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。而 RegistryObject 繼承類別 ExtrinsicObject 的範例 XML

檔如下所示：

【程式碼 4-1】Asset 的 ExtrinsicObject 範例 XML 檔

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ExtrinsicObject mimeType="application/octet-stream" isOpaque="false" majorVersion="1"
minorVersion="0" objectType="urn:uuid:0b02d61a-ad3a-4e82-a87d-e7fed50aa3a5"
status="urn:uuid:37d17f1b-3245-425b-988d-e0d98200a146"
id="urn:uuid:f74a8ea6-7b3a-46f2-a7f6-a085e0d63414"
xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
  <Name>
    <LocalizedString value="鑰匙圖檔" charset="UTF-8" xml:lang="zh-TW"/>
  </Name>
  <Description>
    <LocalizedString value="大小為32x32pixel的鑰匙圖示檔" charset="UTF-8"
xml:lang="zh-TW"/>
  </Description>
  <Slot slotType="" name="copyrightAndOtherRestrictions">
    <ValueList>
      <Value>no</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="cost">
    <ValueList>
      <Value>no</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="fileName">
    <ValueList>
      <Value>key.zip</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="format">
    <ValueList>
      <Value>image/gif</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="general_identifier">
    <ValueList>
      <Value>c09ee0ac-8c3a-436e-8e79-e6df560f5006</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="meta_identifier">
    <ValueList>
      <Value>d1f8e1d0-6475-4a64-b3b5-b0f313f4a46a</Value>
    </ValueList>
  </Slot>
</ExtrinsicObject>
```

```
<Slot slotType="" name="metadataSchema">
  <ValueList>
    <Value>LOMv1.0</Value>
    <Value>SCORM_CAM_v1.3</Value>
  </ValueList>
</Slot>
<Slot slotType="" name="size">
  <ValueList>
    <Value>1091</Value>
  </ValueList>
</Slot>
<Classification classifiedObject="urn:uuid:f74a8ea6-7b3a-46f2-a7f6-a085e0d63414"
classificationNode="urn:uuid:43f4d465-1212-4d97-86f3-eb0c524b1909"
objectType="urn:uuid:65e731a8-3325-4ac5-bd95-d71a277e3216" status="null"
id="urn:uuid:095b9031-4236-439b-8136-924ac417c9f9">
  <Name/>
  <Description/>
</Classification>
</ExtrinsicObject>
```

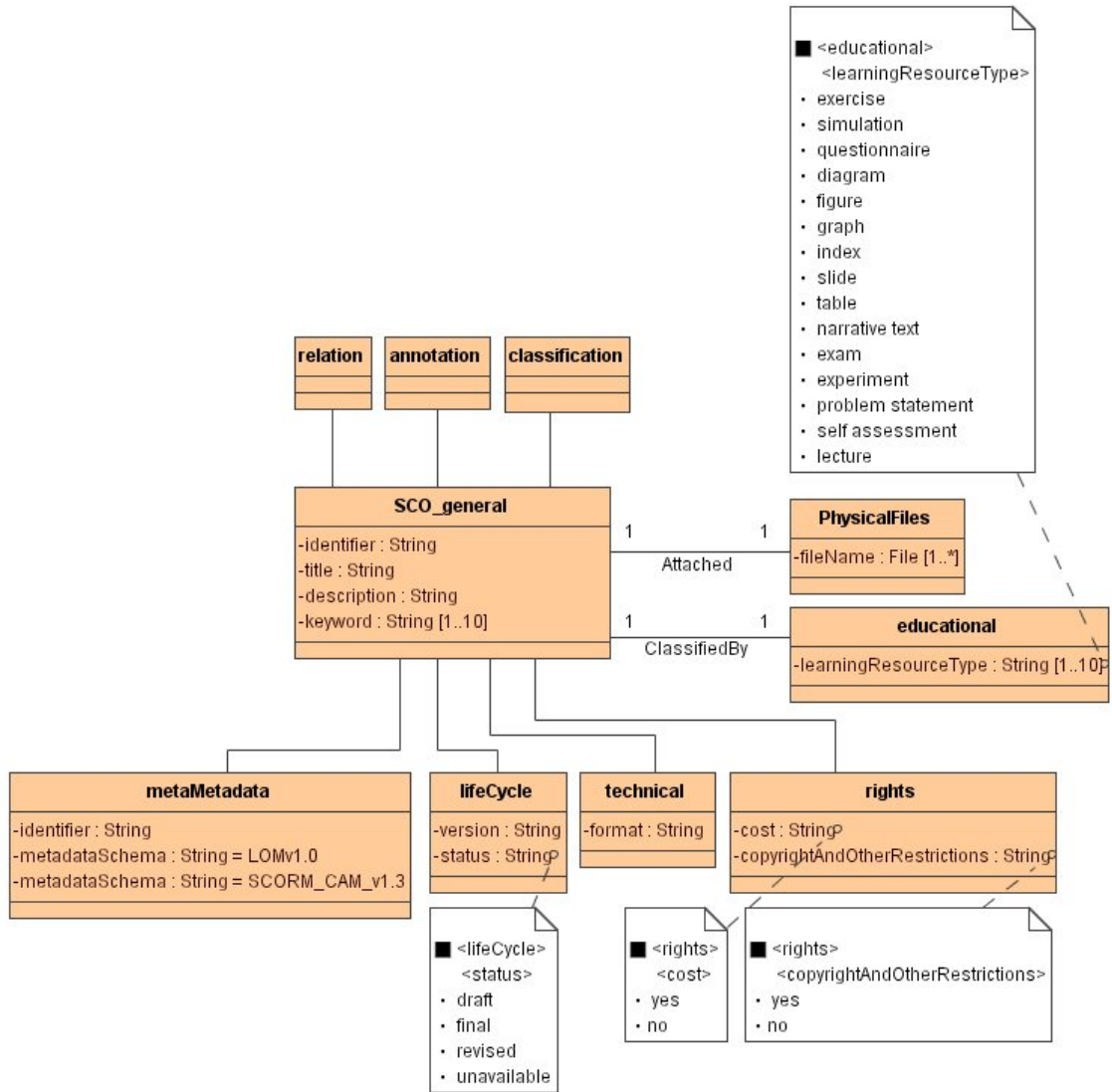

第三節 SCO-RO 註冊物件規劃

相較於第二節的 Asset-RO 註冊物件，本節的 SCO-RO 註冊物件中的

PhysicalFiles.fileName 為一到多個實體檔案壓縮至一個 ZIP 檔內，除此之外，增

加了 **general.keyword**、**lifeCycle.version**、**lifeCycle.status** 三項元素，架構如【圖

4-3】所示。



【圖 4-3】SCO-RO 註冊物件 Class Diagram

上圖 SCO Meta-data 與 ebRIM 的對應關係可以整理成下表：

【表 4-2】SCO-RO 對照表

LOM / File	Multiplicity	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.title 及 **general.description** 直接採用 RegistryObject 預設屬性 Name 及 Description 來對應儲存。

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。而 RegistryObject 繼承類別 ExtrinsicObject 的範例 XML 檔如下所示：

【程式碼 4-2】SCO 的 ExtrinsicObject 範例 XML 檔

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ExtrinsicObject mimeType="application/octet-stream" isOpaque="false" majorVersion="1"
minorVersion="0" objectType="urn:uuid:9624fc79-d9dc-4688-aad8-eca8d1f811a3"
status="urn:uuid:37d17f1b-3245-425b-988d-e0d98200a146"
id="urn:uuid:22a253e1-9f0e-4fb6-b627-fc20a175ffc6"
xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
  <Name>
    <LocalizedString value="XML技術平台" charset="UTF-8" xml:lang="zh-TW"/>
  </Name>
  <Description>
    <LocalizedString value="說明XML技術核心, XML視覺呈現, XML資料處理等項目。"
    charset="UTF-8" xml:lang="zh-TW"/>
  </Description>
  <Slot slotType="" name="copyrightAndOtherRestrictions">
    <ValueList>
      <Value>no</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="cost">
    <ValueList>
      <Value>no</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="fileName">
    <ValueList>
      <Value>Ch_1_5_2.zip</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="format">
    <ValueList>
      <Value>text/html</Value>
    </ValueList>
  </Slot>
</ExtrinsicObject>
```

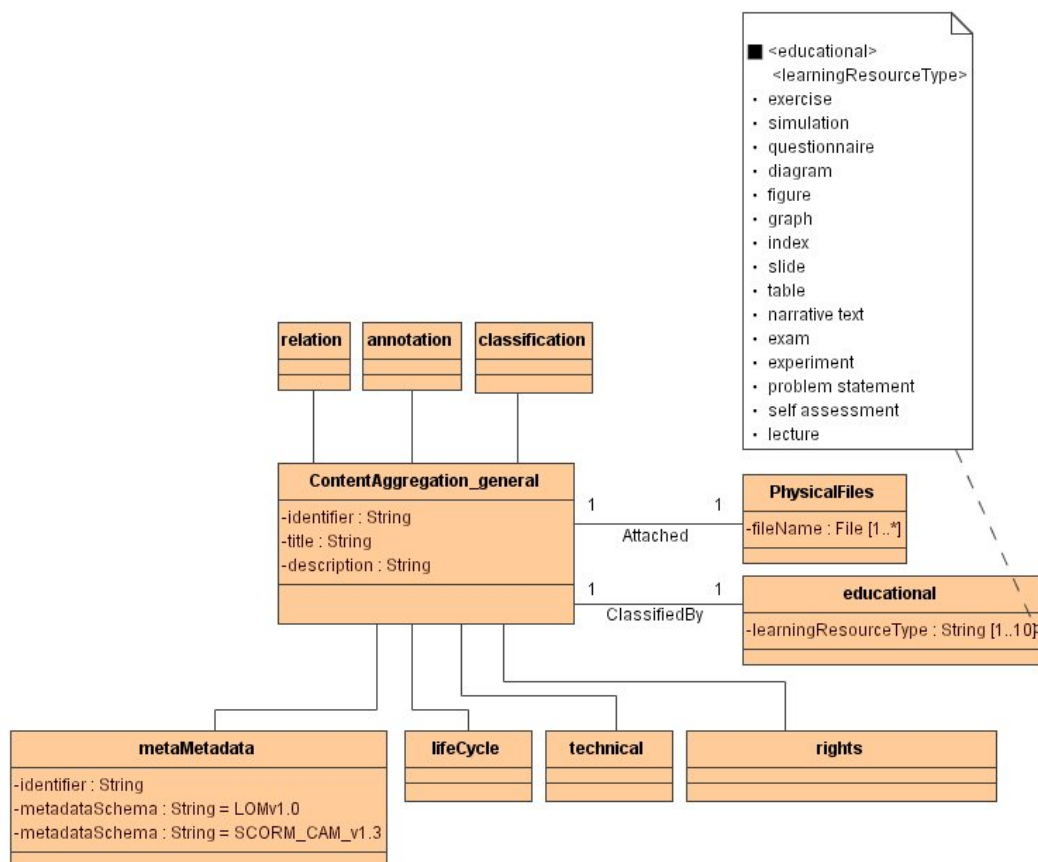
```

    </ValueList>
  </Slot>
  <Slot slotType="" name="general_identifier">
    <ValueList>
      <Value>c791b151-e1b9-4aec-82ef-f24a6ef401e1</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="keyword">
    <ValueList>
      <Value>XML</Value>
      <Value>XML技術架構</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="meta_identifier">
    <ValueList>
      <Value>22efce52-fc65-4ac1-ba21-8ed2d3b0ace1</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="metadataSchema">
    <ValueList>
      <Value>LOMv1.0</Value>
      <Value>SCORM_CAM_v1.3</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="status">
    <ValueList>
      <Value>final</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="version">
    <ValueList>
      <Value>1.0</Value>
    </ValueList>
  </Slot>
  <Classification classifiedObject="urn:uuid:22a253e1-9f0e-4fb6-b627-fc20a175ffc6"
  classificationNode="urn:uuid:3eaac060-4d30-4172-a556-d0c91b9d73a9"
  objectType="urn:uuid:65e731a8-3325-4ac5-bd95-d71a277e3216" status="null"
  id="urn:uuid:cbc74c57-cd63-4f1a-8345-ea7901d5b506">
    <Name/>
    <Description/>
  </Classification>
  <Classification classifiedObject="urn:uuid:22a253e1-9f0e-4fb6-b627-fc20a175ffc6"
  classificationNode="urn:uuid:9aa6518d-dfda-44ee-b804-8f74b1d5cd02"
  objectType="urn:uuid:65e731a8-3325-4ac5-bd95-d71a277e3216" status="null"
  id="urn:uuid:2d33edb4-0471-460d-b8c0-f54c394d9176">
    <Name/>
    <Description/>
  </Classification>
</ExtrinsicObject>

```

第四節 Content Aggregation-RO 註冊物件規劃

相較於前兩節的 Asset-RO、SCO-RO 註冊物件，本節 Content Aggregation Meta-data 的必要性元素只有 **metaMetadata.metadataSchema** 這一項，內容限定至少須有 LOMv1.0 及 SCORM_CAM_v1.3 兩項屬性值，而 Content Aggregation-RO 註冊物件中的 **PhysicalFiles.fileName** 為一到多個實體檔案壓縮至一個 ZIP 檔內，為了將 Content Aggregation Meta-data 確實對應至 ebRIM 模型，我們增加 **general.identifier**、**general.title**、**general.description** 及 **metaMetadata.identifier**，其架構如【圖 4-4】所示。



【圖 4-4】 Content Aggregation-RO 註冊物件 Class Diagram

上圖 Content Aggregation Meta-data 與 ebRIM 的對應關係可以整理成下表：

【表 4-3】 Content Aggregation-RO 對照表

LOM / File	Multiplicity	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。而 RegistryObject 繼

承類別 ExtrinsicObject 的範例 XML 檔如下所示：

【程式碼 4-3】 Content Aggregation 的 ExtrinsicObject 範例 XML 檔

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ExtrinsicObject mimeType="application/octet-stream" isOpaque="false" majorVersion="1"
minorVersion="0" objectType="urn:uuid:4be6de79-a5e1-4aa0-89af-9a5e7e8be103 "
status="urn:uuid:37d17f1b-3245-425b-988d-e0d98200a146"
id="urn:uuid:1339b4f2-62dd-4920-b053-dc6f0d072f58"
xmlns="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
  <Name>
    <LocalizedString value="XML課程-第一章 XML 的時代" charset="UTF-8"
xml:lang="zh-TW"/>
  </Name>
  <Description>
    <LocalizedString value="Introduction to XML" charset="UTF-8" xml:lang="zh-TW"/>
  </Description>
  <Slot slotType="" name="fileName">
    <ValueList>
      <Value>Ch_1_5_2.zip</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="general_identifier">
    <ValueList>
      <Value>2a64d4d0-3cbd-40c3-af19-f127ef5edacf</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="meta_identifier">
    <ValueList>
      <Value>e71af9bf-9337-4902-bef8-960d9b382ffa</Value>
    </ValueList>
  </Slot>
  <Slot slotType="" name="metadataSchema">
    <ValueList>
      <Value>LOMv1.0</Value>
      <Value>SCORM_CAM_v1.3</Value>
    </ValueList>
  </Slot>
  <Classification classifiedObject="urn:uuid:1339b4f2-62dd-4920-b053-dc6f0d072f58"
classificationNode="urn:uuid:9aa6518d-dfda-44ee-b804-8f74b1d5cd02"
objectType="urn:uuid:65e731a8-3325-4ac5-bd95-d71a277e3216" status="null"
id="urn:uuid:128f868b-9d53-48ac-b1e0-9eecbb6edcdf">
    <Name/>
    <Description/>
  </Classification>
  <Classification classifiedObject="urn:uuid:1339b4f2-62dd-4920-b053-dc6f0d072f58"
classificationNode="urn:uuid:3eaac060-4d30-4172-a556-d0c91b9d73a9"
objectType="urn:uuid:65e731a8-3325-4ac5-bd95-d71a277e3216" status="null"
id="urn:uuid:74637971-6e90-4113-8b64-d38fee28d1b1">
    <Name/>
    <Description/>
  </Classification>
</ExtrinsicObject>
```