一個 LOT 存取控制模式應用於數位學習系統

薛夙珍　　呂瑞麟+　　白琤瑜
朝陽科技大學 資訊管理系　國立中興大學 資訊管理系　朝陽科技大學 資訊管理系
schsueh@cyut.edu.tw　jllu@ nchu.edu.tw　s9414622@cyut.edu.tw

摘要

隨著網路與資訊科技的日新月異，數位學習已是一個充實個人知識，掌握知識脈動的重要應用，因此產學界無不重視數位學習系統的建置。愈來愈多老師利用數位化教材資源來提升學生的學習效率與提供彈性的學習方式。有鑑於流通在數位學習系統中的教材資源數量愈來愈龐大，系統中使用者人數愈來愈多，為了有效的管理使用者使用教材資源的存取權限，數位學習系統中的存取控制機制即成為一項非常重要的研究議題。可是到目前為止，應用於數位學習系統中的存取控制機制仍顯不足，尤其是針對教材編輯者對於教材資源的存取控制。因此本研究提出一個 LOT 存取控制模式，主要以 RBAC 架構為基礎，並結合 ABAC 機制之屬性特性輔以教材資源的設定，主要之目的在於改善傳統 RBAC 模式應用於數位學習系統中教材編輯者角色數量過多之問題，使得有效控管數位學習系統中不同教材編輯者對教材資源的存取控制。本研究將以數位學習系統開發為實例，驗證本研究架構之可行性。

關鍵詞：數位學習系統、ABAC、屬性、存取控制

1. 前言

在網際網路普及與資訊科技愈趨成熟的環境下，各教學單位對於數位學習系統(Learning Management System，簡稱 LMS)的建置日益重視，許多老師結合傳統上課方式和數位化的教學來發展更多元的教學策略。然而隨著參與數位學習系統的使用者(學習者與教材編輯者)人數愈來愈多，數位化教材資源(課程、教材)在系統內相互傳播、交流的情況下，為避免使用者對教材資源的破壞或惡意刪除，即有必要對教材資源的存取做控管。

過去，許多專家學者曾為使用者之存取權限管理提出不同的存取控制機制[3][9][12]。其中 Sandhu 等學者所提出以角色為基礎的存取控制機制（Role-based Access Control，簡稱 RBAC）[6]較為主流。其不同於傳統的存取控制機制是由系統直接授予使用者對物件的存取權限[14]，或將物件分類為不同安全存取等級之方式[11]；而是以角色設定方式，針對系統中使用者、角色與物件存取權限的指派關係來減少授權規則(Authorization Rules)，以達存取控制之目的。日後，相繼發展之存取控制機制，多數均係以 RBAC 為基礎而加以延伸。

為了有效管理數位學習系統中使用者對教材資源的存取權限，Lu 與 Chen 將 RBAC 模式應用於 Web-based 數位學習系統中，該存取控制機制簡稱為 DSS[8]。該機制結合 RBAC 模式與 XML 檔案之特性，發展出符合數位學習系統之存取控制模式，使得系統中使用者之存取權限能有效控管，並減少系統安全管理者之負擔。

因 DSS 機制係以 RBAC 之角色設定方式來管理使用者權限。故以該存取控制機制用於管理學習者的存取權限時，縱令學習者數量眾多，我們亦僅需要依照學習者的學習課程或學習主題即可定義出適合之角色數量，而為有效控管。可惜的是，若將此存取控制機制用於管理教材編輯者之權限時，需為每位教材編輯者定義出一個角色，然而隨著系統中參與的教材編輯者人數愈來愈多時，即其角色數量則會隨著增加。如此一來，在管理教材編輯者中，所需之角色成本勢必大幅提升。因此，本研究進一步修改 DSS 系統模式中 RBAC 架構，結合以屬性為基礎之存取控制機制(Attribute Based Access Control，簡稱 ABAC)[16]中屬性之特性輔以教材資源的設定，發展出一套更適用於數位學習系統的存取控制模式，簡稱為 LOT model (Learning Objects with aTtribute Model)，使得當應用這個模式時，不論從學習者的角度，或從教材編輯者的角度都能夠有效的改善角色存取權限的管理，並減少系統中角色數量的產生，以簡化系統管理角色上的複雜度。

在第二節的文章中，將首先簡單地介紹過去與存取控制相關的研究以及說明；接著在第三節中，將會敘述 LOT 模式應用於數位學習系統的原則與架構；於第四節中，實作出系統雛型，證明其可行性；最後，在第五節中針對本研究做總結。

2. 文獻探討

2.1 傳統的存取控制模式

在傳統的系統存取控制領域中分為二類，一類為 Sandhu 與 Munawer 所提出之任意式存取控制機制（Discretionary Access Control，簡稱 DAC）[14]、另一類為 Nyauchama 與 Osborn 所提出之命令式存

本研究承蒙行政院國家科學委員會之經費補助(NSC 95-2221-E-005-050-MY2)，謹此致謝。

取控制機制(Mandatory Access Control，簡稱MAC)[11]。在 DAC 的存取控制機制中其使用者之存取權限是由系統所直接授予的，雖然 DAC 模式可提供一個具有彈性的存取控制機制，但是其本身並無法有效確保資料授權後的使用情形，所以此種模式並不適用於 Web 的環境。而 MAC 模式雖然可改善上述的問題，並且被廣泛的使用於軍事上的應用，但是由於必先針對所有系統內所涉及之不同主體及物件，設定不同保密程度與類別的標籤。是以，當使用者存取該物件時，此機制會比對此兩者標籤之等級而予以存取權限，其相較於 DAC 模式更為複雜。由於傳統的存取控制模式是由使用者為主，當使用者的存取權限有所改變時，則必需重新設定其權限。因此，任一的 DAC 或是 MAC 模式可能無法滿足組織安全上的需求，所以另有學者提出依角色為基礎之存取控機制，應用於組織權限之管理上。

2.2 RBAC

Sandhu 等學者所提出以角色為基礎的存取控制機制（Role-based Access Control，簡稱 RBAC）[13]，其不同於以往將權責直接指派至某一特定使用者之存取控制方式，而是針對企業組織中的角色存取物件之權限做管理。其利用組織上權責的架構，定訂所需之角色(Role)，再將一群使用者(Users)指定至角色，依據角色之不同給予不同的權責(Permissions)。而當使用者要改變其權責時，只要把使用者指定到一個新的角色，並取消原角色連結即可。由於企業組織架構與權責的變化不大，故 RBAC 能有效簡化權限之管理。

2.3 DSS

Lu 與 Chen 曾將 RBAC 應用於數位學習系統的研究上提出一個以 Web-based 的教材資源存取控制機制簡稱 DSS[8]。該機制是利用 XML 結構配合 RBAC 機制中以角色為主的觀念所建置而成，其角色設定是依學習課程或學習主題所定義，並利用 XML 存取控制之特性，設定角色權限的範圍與相關之存取限制，依據不同使用者所允許存取內容之不同，產生具有彈性的個人化導覽控制介面，讓系統中的使用者可以從 Web 介面看到屬於個人化之存取權限，而無法看到不屬於個人之非法權限。如此一來就不必擔心有越權之行為發生，使得系統中使用者對教材資源的存取權限能有效控管。此機制採用 XML 記錄使用者資訊、角色存取範圍與相關存取權限，不僅能達到文件的細緻化控制(fine-grained control)，並能減少系統安全管理者的負擔。

2.4 Attribute Based Access Control Models

以屬性為基礎(Attribute-based)之研究是最近幾年來存取控制模式的最新發展趨勢之一，這個存取控制機制稱為 Attribute-based Access Control，簡稱 ABAC[15]。ABAC 的應用方式有許多，例如：在 Unix 檔案系統中，管理者可依不同群組之使用者分別為其所存取之每份檔案設定屬性<讀 r、寫 w、執行 x>以達到存取控制之目地。Yuan 與 Tong 提出一個應用於 Web Services 上的 ABAC 模式[16]，此存取控制模式將使用者(subject)、系統資源(resource)與系統環境(environment)定義屬性，並利用存取規則(Rule)的描述以達到存取控制之目地。例如：此存取控制機制應用於網際網路中線上娛樂系統，不同年齡之使用者對不同等級之電影檔案的存取控制。其利用使用者年齡屬性 Age(u)與電影分級屬性 Rating(m)結合存取規則的描述 can_access(u,m)←(Age(u)<13∩Rating(m) ∈{G})，有效限制未滿 13 歲之兒童只允許觀看普通等級的電影，以達存取控制。

3. LOT Model

本研究提出一個 LOT(Learning Objects with aTtribute Model) 存取控制模式，此模式將改善 RBAC 架構應用於數位學習系統中教材編輯者角色數量過多之問題。為了解決此問題，我們輔以 ABAC 之屬性設定，採以對教材資源設定擁有者(owner)屬性，並以 XML 檔案來記錄使用者與學習資源之資訊、系統所需之角色與角色對學習資源之存取控制。

當教材編輯者登入系統時，系統會驗證其是否為合法之使用者，其驗證資訊儲存於 User Information 內；而使用者登入成功後，系統即會要求其選取所要擔任之角色，其角色資訊儲存於 Role 內；當使用者於系統中新增教材資源時，會先將該教材資源記錄於 Course Hierarchy 內，並同時新增元件屬性 owner 於教材資源的最小單位中，進而記錄教材資源的原始擁有者，以便於授權時的存取控管；並將該使用者所新增教材資源之授權規則記錄於 Authorization Rules 內。

在 LOT model 架構中所定義的系統模型包含了四個資訊內容，如圖 1 所示，有 User Information、Role、Authorization Rules 與 Course Hierarchy。所有內容皆記錄於 XML 文件裡。

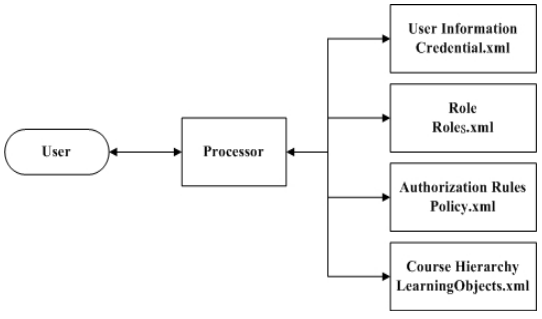


圖 1 系統架構圖

3.1 Role

角色即為此數位學習系統中所需之角色，角色在此系統中有一項要求：

(1) 角色在系統中是唯一且不可重覆的。在此以 rid 表示單一角色；∃! 表示存在且唯一之意，其關係式如下：

$$\exists!\ \text{rid} : \text{rid} \in \text{ROLES}$$

其中，ROLES表示系統內所有角色的集合，此 $\text{ROLES} \neq \varnothing$ 。

為了符合上述之要求，其 XML 的寫作方式，如圖 2 所示。以此數位學習系統為例，所列舉之角色即為教材編輯者與學習者。其根節點為<Roles>標籤，而<Roles>標籤內包含二個子標籤<Role>，表示系統擁有教材編輯者與學習者此二個角色；<Role>標籤內的屬性 id 表示系統中的角色，而 label 即為系統呈現該角色之資訊。

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Roles>
    <Role id="T_001_00" label="Teacher" />
    <Role id="S_001_00" label="Student" />
  </Roles>
```

圖 2 Roles.xml

3.2 User Information

使用者資訊即在記錄系統中所有的使用者登入系統時所需要之使用者帳號及密碼與系統中所使用的使用者名稱。使用者資訊在此系統中有下列四項要求：

(1) 使用者帳號必需是唯一的。在此以 uid 表示使用者帳號，其關係式如下：

$$\exists!\ \text{uid} : \text{uid} \in \text{USERS}$$

其中，USERS表示系統內所有使用者的集合，此 $\text{USERS} \neq \varnothing$ 。

(2) UA 表示系統中使用者對應至角色的關係。其關係式如下：

$$\text{UA} \subseteq \text{USERS} \times \text{ROLES}$$

(3) Assigned_roles()為一函數，表示使用者所對應的角色，其所得之結果為使用者所擁有之角色集合；即表每位使用者可擁有一至多個角色。其關係式如下：

$$\text{Assigned\_roles}(\text{uid}) = \{r \in \text{ROLES} \mid (\text{uid}, r) \in \text{UA}\}$$

(4) currentRole()為一函數，表示使用者登入系統時，於所擁有之角色的集合中選擇所需擔任之角色；即目前所處之角色。其關係式如下：

$$\text{currentRole}(\text{uid}) = \{\text{rid} \in \text{Assigned\_roles}(\text{uid})\}$$

為了符合以上之要求，其 XML 的寫作方式，如圖 3 所示。其根節點為<Users>標籤，而<Users>標籤內包含多個<User>標籤，表示系統擁有多位使用者；每位使用者之資訊即分別記錄於<User>標籤內。<User>標籤內的屬性 id 表示使用者登入系統時所需之使用者帳號，passwd 表示使用者登入系統時所需驗證之密碼，username 為使用者在系統中的使用者名稱。而每個<User>標籤下能擁有一至多個<Role>子標籤，而此<Role>標籤之內容，必需為系統中所定義之角色。例如使用者 John Williams 登入系統時，需以帳號 John 與密碼 123 做為系統之驗證，登入系統後其擁有二個角色分別為 T_001_00 與 S_001_00，John 必需在此二個角色中，選擇其所需擔任之角色，而依角色之不同給予不同之存取權限。

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Users>
  - <User id="John" passwd="123" username="John Williams">
      <Role>T_001_00</Role>
      <Role>S_001_00</Role>
    </User>
  - <User id="May" passwd="456" username="May Lin">
      <Role>T_001_00</Role>
    </User>
  </Users>
```

圖 3 Credential.xml

3.3 Course Hierarchy

每個教材資源新增至系統時，系統會將教材資源的相關資訊記錄於 Course Hierarchy 內。每個教材資源下可有章(Chapter)、段落(Section)或是子段落(Subsection)；而在此系統所使用之教材資源範例中最小的教材資源單位為課。

為了有效達到角色存取權限之管理與有效的減少角色數量產生，本研究主要在於新增教材資源時於每個最小單位的教材資源中增加了課程元件之屬性 owner，此屬性記錄了課程元件的原始擁有者。是以，當不同使用者皆擁有同一教材編輯者角色時，此角色雖擁有某特定之教材資源的基本存取權限；然而，因教材資源具有屬性 owner，因此此模式將會依此教材資源屬性 owner 之值比對使用者帳號，若使用者帳號符合 owner 值，即給予此使用者對教材資源更高的存取權限。例如：教材資源 Course-1 與 Course-2 為使用者 John 所新增的，所以在此二份教材資源下的每個課程元件中屬性設定皆為 owner=John；而教材資源 Course-3 為使用者 May 所新增，所以其下每個課程元件中屬性 owner 設定皆為 owner=May。使用者 John 登入系統後，選擇教材編輯者 T_001_00 這個角色（角色存取權限將於下段 Authorization Rules 中說明），而此角色雖擁有對教材資源 Course-1、2 與 3 的基本存取權限 VIEW，而因此模式亦會比對教材資源下課程元件屬性 owner 值，使用者 John 為教材資源 Course-1

```
<?xml version="1.0" encoding="UTF-8" ?>
- <LOs>
  - <Lo id="Course-1" label="Artificial Intelligence" level="1">
      <Element id="Lesson1" label="Introduction" src="/CourseImports/Course-1/AI/Introduction.htm" level="1-1" owner="John" />
      <Element id="Lesson2" label="Stimulus-Response (S-R) Agents" src="/CourseImports/Course-1/AI/Lesson1.htm" level="1-2" owner="John" />
      <Element id="Lesson3" label="Agents as Tools of the Information Society" src="/CourseImports/Course-1/AI/Lesson2.htm" level="1-3" owner="John" />
      <Element id="Lesson4" label="Intelligent Agents" src="/CourseImports/Course-1/AI/Lesson3.htm" level="1-4" owner="John" />
      <Element id="Lesson5" label="Intelligent Software Agents" src="/CourseImports/Course-1/AI/Lesson4.htm" level="1-5" owner="John" />
      <Element id="Lesson6" label="Solving Problems by Searching" src="/CourseImports/Course-1/AI/Lesson5.htm" level="1-6" owner="John" />
      <Element id="Lesson7" label="Logical Agents" src="/CourseImports/Course-1/AI/Lesson6.htm" level="1-7" owner="John" />
    </Lo>
  - <Lo id="Course-2" label="Photoshop" level="2">
      <Element id="Lesson1" label="Introduction" src="/CourseImports/Course-2/Photoshop/intro.htm" level="2-1" owner="John" />
      <Element id="Lesson2" label="Interface" src="/CourseImports/Course-2/Photoshop/Lesson1.htm" level="2-2" owner="John" />
      <Element id="Lesson3" label="Toolbox" src="/CourseImports/Course-2/Photoshop/Lesson2.htm" level="2-3" owner="John" />
      <Element id="Lesson4" label="Palettes" src="/CourseImports/Course-2/Photoshop/Lesson3.htm" level="2-4" owner="John" />
      <Element id="Lesson5" label="Layers" src="/CourseImports/Course-2/Photoshop/Lesson4.htm" level="2-5" owner="John" />
      <Element id="Lesson6" label="Color Balance" src="/CourseImports/Course-2/Photoshop/Lesson5.htm" level="2-6" owner="John" />
      <Element id="Lesson7" label="Brightness and Contrast" src="/CourseImports/Course-2/Photoshop/Lesson6.htm" level="2-7" owner="John" />
      <Element id="Lesson8" label="Hue and Saturation" src="/CourseImports/Course-2/Photoshop/Lesson7.htm" level="2-8" owner="John" />
      <Element id="Lesson9" label="Selection Tools" src="/CourseImports/Course-2/Photoshop/Lesson8.htm" level="2-9" owner="John" />
      <Element id="Lesson10" label="Transform" src="/CourseImports/Course-2/Photoshop/Lesson9.htm" level="2-10" owner="John" />
    </Lo>
  - <Lo id="Course-3" label="Java" level="3">
      <Element id="Lesson1" label="Introduction" src="/CourseImports/Course-3/Java/intro.htm" level="3-1" owner="May" />
      <Element id="Lesson2" label="Defining Your Own Classes" src="/CourseImports/Course-3/Java/Lesson1.htm" level="3-2" owner="May" />
      <Element id="Lesson3" label="Selection statements" src="/CourseImports/Course-3/Java/Lesson2.htm" level="3-3" owner="May" />
      <Element id="Lesson4" label="File Input and Output" src="/CourseImports/Course-3/Java/Lesson3.htm" level="3-4" owner="May" />
      <Element id="Lesson5" label="GUI" src="/CourseImports/Course-3/Java/Lesson4.htm" level="3-5" owner="May" />
    </Lo>
</LOs>
```

圖 4 LearningObjects.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
- <Policy_base>
    <Policy_spec cred_expr="//User[@role='T_001_00']" path="//Lo[@id='Course-1']" priv="VIEW" />
    <Policy_spec cred_expr="//User[@role='T_001_00']" path="//Lo[@id='Course-2']" priv="VIEW" />
    <Policy_spec cred_expr="//User[@role='T_001_00']" path="//Lo[@id='Course-3']" priv="VIEW" />
  </Policy_base>
```

圖 5 Policy.xml

與 Course-2 之擁有者，因此其對 Course-1 與 Course-2 的存取權限即會自動被提升為 UPDATE，Course-3 仍維持基本權限 VIEW；反之，使用者 May 登入系統時，選擇 T_001_00 這個角色，因其非教材資源 Course-1 與 2 下課程元件之 owner，因此只有此二份教材資源之基本存取權限 VIEW，而擁有教材資源 Course-3 之 UPDATE 存取權限。

其 XML 的寫作方式，如圖 4 所示。其根節點為<LOs>標籤，而<LOs>標籤內包含多個<Lo>標籤，表示此系統內存有多份教材資源，當系統中新增一份教材資源，則會新增一個<Lo>標籤，並在其節點下新增教材資源階層架構<Element>標籤；每份教材資源之資訊即分別記錄於<Lo>標籤內，<Lo>標籤內的屬性 id 表示此份教材資源於系統中之代碼，其會隨著系統內教材資源的增加而依序增加；label 表示教材資源呈現於系統時之資訊，如 id 為 Course-1 的教材資源，其內容是關於 Artificial Intelligence；level 則是用來描述教材資源的階層架構。Element 標籤內的屬性 id 表示此課程在此份教材資源內的代碼；label 則為課程之資訊；src 表示教材資源的 URL；level 表示此課程在教材資源內的階層架構；owner 則記錄此份課程的原始擁有者。

我們認為最適合教材資源屬性 owner 的設定方式是將其設定於教材資源的最小單位之中。其原因如下：

(1) 一份課程教材內的各章節可能由數個教材編輯者所分擔編輯而成，所以一份課程教材中各章節的 owner 可能會是不同人。

(2) 便於日後委任授權他人使用時可達到細微控制，並便於追蹤此份教材中的各章節之原始擁有者為誰。

4. Authorization Rules

授權規則為特定之角色對特定之教材資源所擁有之存取權限。當角色新增教材資源時，系統會將角色所擁有之教材資源的存取權限記錄於授權規則中。授權規則在此系統中有下列四項要求與二項規則：

(1) PRMS 表示系統內某角色所擁有權責的集合，即教材資源所擁有之存取權限。其關係式如下：

$$PRMS \subseteq ACTIONS \times OBJECTS$$

其中，ACTIONS 表示對教材資源的存取權限，而 OBJECTS 表示系統內所有教材資源之集合。

(2) PA 表示系統中權責對應至角色的關係。其關係式如下：

$$PA \subseteq PRMS \times ROLES$$

(3) getAction()為一函數，其目的在於取出角色對此份教材資源所擁有的存取權限。其關係式如下：

$$getAction(rid,o)=\{a \in ACTIONS \mid (a,o,rid) \in PA\}$$

在此系統預設之存取權限有二種： VIEW，表示角色對教材資源擁有可閱讀之權限；UPDATE，表示角色對教材資源擁有可修改之

權限。

(4) getOwner()為一函數，其目的即在找尋某權責之教材資源的擁有者，即新增此份教材資源的教材編輯者。其關係式如下：

getOwner(p)={uid∈USERS | (rid,p) ∈PA}

(5) 因教材資源擁有屬性 owner 值，因此在存取控制時即限制了唯有教材資源的擁有者才能允許對此教材資源做 UPDATE 的存取權限。Rule 1：can_Update()為一布林函數，其目的即在判斷使用者所對應之角色，其角色所擁有之權責下教材資源的擁有者，若此使用者為此份教材資源之擁有者，則其允許擁有此份教材資源 UPDATE 的存取權限，其關係式如下：

R1:can_Update(uid,o)←(uid∈ getOwner(getAction(currentRole(uid),o))) || (“UPDATE”∈getAction(currentRole(uid),o))

(6) 當 Rule1 條件不成立時，則需判斷 Rule2。Rule 2：can_View()為一布林函數，其目的即在判斷使用者所對應之角色，其角色若擁有此權責 VIEW 的權限，則允許閱讀此份教材資源。其關係式如下：

R2:can_View(uid) ←(“VIEW” ∈ getAction(currentRole(uid),o))

其 XML 的寫作方式，如圖 5 所示。其根節點為<Policy_base>標籤，而<Policy_base>標籤內包含多個<Policy_spec>標籤，表示目前系統中有多項權責，當系統內的授權規則增加時，<Policy_spec>即會隨著增加；cred_expr 之值即代表一個角色，此角色能對 path 中的教材資源做存取；path 表示被授權的教材資源；priv 則表示對教材資源的存取的權限，權限內容可以是“VIEW”或“UPDATE”。

5. LOT 存取控制模式應用於數位學習系統之系統雛型

圖 6 系統雛型

本研究採用 XML 記錄系統內所有相關資訊及安全防護架構，同時運用 Apache’s Tomcat 網站伺服器版本 5.0.28、Sun’s Java Standard Edition 版本 1.5 及 JAVA 等技術結合 ADL 所提供的 LMS，結合本研究所提出之 LOT 存取控制機制，開發數位學習系統平台雛型，提供一個以 Web-based 的使用者介面，系統雛型如圖 6 所示。

本研究以 DSS 架構中 RBAC 模式與 LOT 存取控制模式相互比較，證明本研究所提之模式確實能有效的減少數位學習系統中教材編輯者角色數量的產生，亦達到存取權限的管理。依 RBAC 中角色存取權限控管之方式，為了讓每位教材編輯者只允許對自己所擁有之教材資源擁有存取權限，系統需依每位教材編輯者定義出一個角色，而其角色只擁有自己所增加之教材資源的存取權限。如圖 7 所示，使用者 John、May 與 Tom 各自新增了二份教材資源，而每位教材編輯者只擁有自己之教材資源的存取權限，因此需設定三位角色，每個角色分別擁有其各自教材資源的存取權限，而隨著系統教材編輯者人數愈來愈多(假設人數為 N)，即角色數量亦會跟著成長為 N；反之，在 LOT model 中，因每份教材資源皆擁有屬性 owner 之設定，因此即使是由不同教材編輯者同處一個角色，其存取權限仍會受到屬性設定而限制，因此只需要一個角色即可管理系統內所有之教材編輯者對其學習資源的存取權限。因此 LOT 存取控制模式，無論由學習者的角度，或從教材編輯者的角度都能夠有效的管理角色數量的產生，簡化系統管理角色之複雜度。

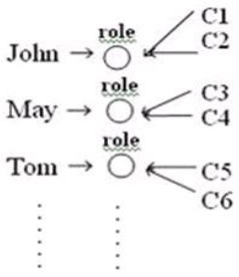
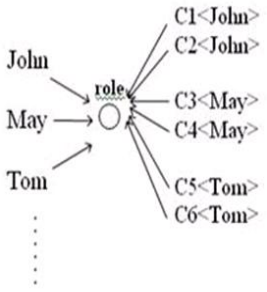
RBAC	LOT model
教材編輯者之角色數量	教材編輯者之角色數量
	

圖 7 RBAC 與 LOT model 之比較

5. 結論與未來發展

以往 RBAC 的存取方式為有不同的權責產生時，為了達到有效的存取控制則需產生新的角色，例如：John 屬於 T_001_00 這個教材編輯者的角色，而此角色擁有學習資源 Artificial Intelligence 與 Photoshop 的 VIEW 與 UPDATE 的權限，而另一位使用者 May 雖然亦是教材編輯者的身份，但為了達到存取控制(沒有權限去 UDATE 教材資源 Artificial Intelligence 和 Photoshop)，則需產生另一個角色 T_002_00，因此隨著系統中相同性質的使用

者人數愈來愈多時，角色數量的控管為一個困難的工作。

以教材資源設定屬性的方式不但可讓相同性質的使用者 John 與 May 同屬於教師 T_001_00 這個角色，因課程教材中多了屬性 owner 這個設定，所以在匯入或新增課程元件時即會記錄 owner 為何人，因此即使是同一個角色要存取相同課程元件仍會有不同權限之分別。

此模式應用於數位學習系統中，能有效減少傳統 RBAC 模式的系統中角色過多的問題。未來將於此研究模型上加委任授權模式以使此模式更符合數位學習系統之需求。

6. 參考文獻

[1] ADL Technical Team. Sharable Content Object Reference Model (SCORM) 2004 Documentation 3rd Edition. Advanced Distributed Learning (ADL), November 16 2006.

[2] M. A. Al-Kahtani and R. Sandhu, “A Model for Attribute-Based User-Role Assignment,” Computer Security Applications Conference, p. 353-362, 2002.

[3] M. A. Al-Kahtani and R. Sandhu, “Access Control Models and Mechanisms:Induced Role Hierarchies with Attribute-Based RBAC,” Proceedings of the. eighth ACM symposium on Access control models and technologies, p. 142-148, 2003.

[4] Shihyu Chou, Eric Jui-Lin Lu, and Yi-Hui Chen, “X-RDR: A Role-based Delegation Processor for Web-based Information Systems,” ACM Operating Systems Review, vol. 39, no. 1, p. 4-21, 2005.

[5] Steve DeRose and Eve Maler, ” XML Pointer Language (XPointer),” World Wide Web Consortium Working Draft, 1998.

[6] David F. Ferraiolo et al., “ Proposed NIST Standard for Role-based Access Control,” ACM Transactions on Information and Systems Security, vol. 4, no. 3, p. 224-274, 2001.

[7] Sherry Lorraine and Morse Richard, “An Assessment of Training Needs in the Use of Distance Education for Instruction,” International Journal of Educational Telecommunications, vol. 1, no. 1, p. 5-22, 1995.

[8] Eric Jui-Lin Lu and Yi-Hui Chen, “The Design of A Delegable SCORM Conformant Learning Management System,” Journal of Computer Assisted Learning, vol. 22, no. 6, p. 423-436, 2005.

[9] Eric Jui-Lin Lu and Rai-Fu Chen, "Design and Implementation of A Fine-Grained Menu Control Processor for Web-based Information Systems ," Future Generation Computer Systems, vol. 19, no. 7, p. 1105-1119, 2003.

[10]Eric Jui-Lin Lu and Yi-Hui Chen, “A Study on Delegation Processors for Web-based Information Systems,” Department and Graduate Institute of Information Management Chaoyang University of Technology, p. 1-73, 2003.

[11]Matunda Nyauchama and Sylvia Osborn, “Modeling mandatory access control in role-based security systems,” Proceedings of the ninth annual IFIP TC11 WG11.3 working conference on Database security IX : status and prospects: status and prospects, p. 129-144, 1996.

[12]Sylvia Osborn, R. S. Sandhu, and Qamar Munawer, “Configuring role-based access control to enforce mandatory and discretionary access control policies,” ACM Transactions on Information and System Security (TISSEC), vol. 3, no. 2, p. 85-106, 2003.

[13]R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” IEEE Computer, vol. 29, no.2, pp. 38-47, 1996.

[14]R. S. Sandhu and Qamar Munawer, “How to do discretionary access control using roles,” Proceedings of the third ACM workshop on Role-based access control, 1998

[15]Lingyu Wang, duminda Wijesekera, and Sushil jajodia, “A Logic-based Framework for Attribute based Access Control,” Proceedings of the 2004 ACM workshop on Formal methods in security engineering, p. 45-55, 2004.

[16]Eric Yuan and Jin Tong, “Attribute Based Access Control (ABAC) for Web Services,” Proceeds of the IEEE International Conference on Web Services, p. 561-569, 2005.