Metadata-Based Access Control for Digital Libraries

Student: Chung-Wei Huang Advisor: Dr. Hao-Ren Ke, Dr. Wei-Pang Yang

Institute of Computer and Information Science

National Chiao Tung University

ABSTRACT

Digital libraries encompasses a large amount of information and provide access so that everyone can retrieve information on the web. However, some objects in a digital library are not free for use nor accessible to everyone. There must be an access control mechanism which provide authorized users the access rights to certain objects. Traditional access control models are developed for specific purpose. These models are not suitable for digital libraries due to the application of new media. This thesis proposes a metadata-based access control mechanism for digital libraries. Metadata is beneficial for resource discovery. We use the rich descriptive data about digital objects to construct the access control model.

Keyword: Access Control, Metadata, Digital Libraries

數位圖書館中以詮釋資料為基礎之存取控制

研究生：黃崇瑋　　指導教授：柯皓仁博士，楊維邦博士

國立交通大學資訊科學研究所

摘要

數位圖書館收藏為數眾多的資訊供大眾使用，但是有些資訊卻不是無條件可以提供給任何人使用的。有鑑於此，數位圖書館必須要有一個機制使特定的資訊僅提供給經過授權的使用者使用，此機制稱為存取控制。傳統的存取控制面臨新媒體的應用，顯得不適用在數位圖書館的環境中。本論文提出一套以詮釋資料為基礎之存取控制架構，詮釋資料的主要用途是提供資源探索，我們利用其對數位物件豐富的描述性資料來建立存取控制，此外尚提出一套策略以解決詮釋資料中多欄位階層的衝突。本架構具有簡單及易於使用的特性，和其他模型相比，不僅減少管理上的負擔，並兼顧實作上的可行性，有利於快速為數位圖書館建立存取控制機制。

關鍵字：存取控制、詮釋資料、數位圖書館

誌謝

本篇論文之能夠完成，要感謝的人實在太多、太多……兩年多來指導教授柯皓仁老師及楊維邦老師的耐心指導與照顧，實驗室黃夙賢學長及其他多位學長、姊、弟不時地伸出援手，還有計畫室黃明居老師及同事千棻、文亨、玉菱、佳欣、怡君、青華、雪卿、陳大哥、雅青、媛媛、慧貞、蔡姐等幾位陪我一起度過研究生涯。

此外，還有陪伴我無數年的家人、朋友、外甥女-鈺萱，以及淑鈐，沒有你們，日子只是一個饅頭的符號。最後，心中最想感謝的，就是帶我來到這個世界的父、母親，沒有你們，我將只是一團空氣，飄盪於無垠宇宙之中。

目錄

圖目錄

表目錄

第一章　緒論

第一節　研究動機

伴隨著網際網路的蓬勃發展，數位圖書館在近幾年逐漸受到大眾的重視。正因為網路所帶來的便利性，人們逐漸改變尋求資訊的途徑，數位圖書館不再侷限於將資料數位化，更進一步扮演了推廣與教育的角色，透過有系統的規劃與處理，將資訊呈現給大眾。但在資料數位化或資訊傳播的過程中，除了需要考量技術的問題外，還有一項重要的課題，那就是智慧財產權的保護。數位圖書館所收藏的資料，除非因為年代久遠而轉變為公共財，否則即牽涉到智慧財產權的議題。所呈現的資料，是否已經徵得著作權人的同意？是否充分保護著作權人的權益？在數位圖書館的開發過程中，即必須審慎思考數位版權(Digital Rights)的保護機制。

數位版權管理(Digital Rights Management, 簡稱 DRM)[9]包含很多層面，最基本的如存取控制(Access Control)，用來確保資料不被未授權所使用；其他尚包括資料的完整性(Integrity)、付費管理機制等，相關的機制如加密(Encryption)、數位簽章(Digital Signature)或電子商務(E-commerce)等。本篇論文將只著重於存取控制的討論。

傳統的存取控制方法如存取控制矩陣(Access Control Matrix，簡稱 ACM)[12]，建立一人與物之矩陣，描述每個人對每一物件的存取權限。此種做法的優點是簡單明瞭，每個人對每個物件的存取權限都有明確的定義，但是在數位圖書館的環境中，大量的數位物件使得建立矩陣的困難度增加。近來則有以角色(Role)或內容(Content)概念為基礎的存取控制管理，雖然兩種方法不盡相同，卻都有類似的管理策略，即是對使用者及物件分群，將存取控制應用於群的層次上，兩者雖然能大幅改善存取控制矩陣的缺點，但在數位圖書館的應用領域中，仍有一些實際應用的問題尚待克服。例如若增加一項限制為"高品質的影音檔需付費方可

使用”，在以角色為基礎的存取控制中，必須為高品質的影音檔建立一個角色，並一一將符合此條件的物件加入這個角色中，換句話說，在以物件為基礎的存取控制中，必須依據不同的限制要求，一再地建立角色，並做角色分派的動作，這在擁有大量物件的數位圖書館環境下是不切實際的。而在以內容概念為基礎的存取控制中，甚至無法建立前述的存取控制，此外，如何從內容或詮釋資料中萃取出該物件所包含的概念亦是另一項研究課題。

第二節　研究目的

本論文主要目的是針對數位圖書館的物件提出以詮釋資料(Metadata)為基礎的存取控制架構，利用已存在且富含資料屬性之詮釋資料建立存取控制管理策略(Policy)，藉以大幅簡化使用傳統存取控制機制所需之人力物力。本論文所提出之存取控制管理策略兼具簡單及易於使用的特性，有利於為數位圖書館快速建立存取控制機制，從而奠定完整數位版權管理架構之基礎。

第三節　論文架構

本論文第二章介紹相關研究工作，第三章說明我們提出的以詮釋資料為基礎的存取控制架構，第四章說明系統實作，藉以驗證本論文的可行性，第五章是結論與未來的研究方向。

第二章　相關研究

本章說明與本論文相關的研究工作，主要分成三個部份，首先介紹數位圖書館相關背景，包括數位圖書館架構、詮釋資料；第二部份介紹存取控制相關之研究，例如[14]、[1]；因為本論文係根據[1]作修改，因此在第三節中將介紹[1]所提出的方法。圖 1 分別列出這些相關研究及發表年代(詳見本論文參考文件)。

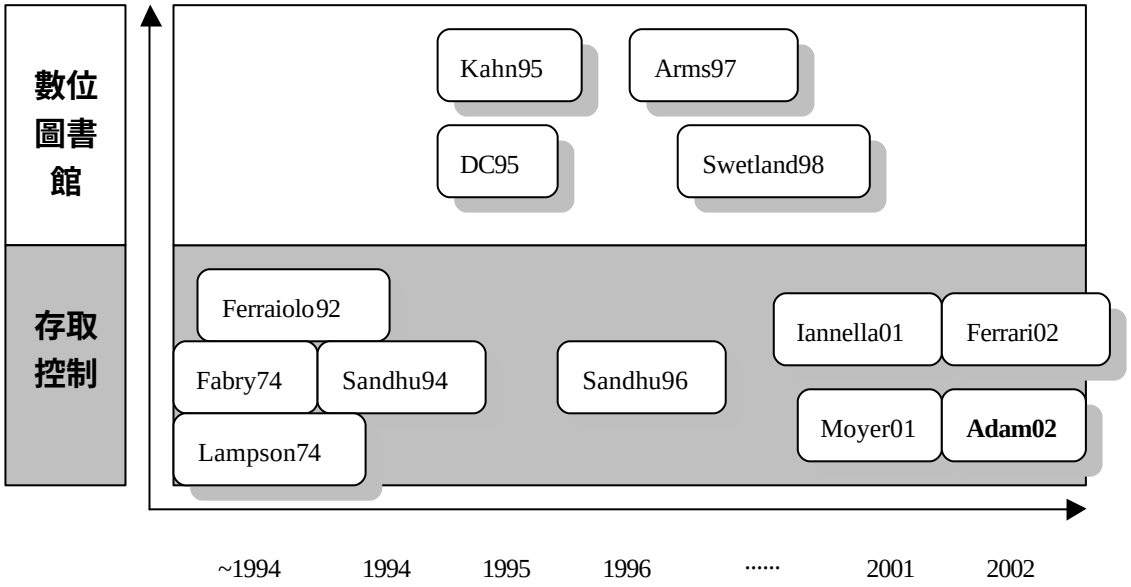


圖 1：相關研究工作

第一節　數位圖書館相關背景

2.1.1 數位圖書館架構

1975 年，Roger W. Christian 在"The Electronic Library"書中首先提出「電子圖書館(Electronic Library)」的概念。另外，美國圖書館學大師 F. W. Lancaster 在 1978 年透過其著作"Toward Paperless Information Systems"，預言在公元二千年後電子出版品將漸漸替代紙本式資料，未來的圖書館可能進入以電子媒體為館藏的電子圖書館時代。近年來，由於各種數位化技術相繼興起，於是「數位圖書館(Digital Library，簡稱 DL)」逐漸成為新興名詞。尤其是在 1992 年時，前美國參

議員高爾(Al Gore)提出「資訊基礎建設與科技法案」，自此之後，有關數位圖書館之各項研究便陸續展開。若要對數位圖書館下個定義，我們可以說所謂的數位圖書館乃是運用電腦技術將書籍、聲音影像、圖片等資料數位化，以電腦的儲存設備(如硬碟、磁帶等)來組織與儲存這些經數位化後的資料，配合功能強大的資訊檢索系統，並透過網際網路，提供讀者資料搜尋、擷取與處理[11]。

Kahn 和 Wilensky 兩人在[10]中首先提出要建立一個數位圖書館的三個基本要素：數位物件(Digital Object)、識別資料(Handle)[17][18]及貯藏庫(Repository)，數位物件即為經數位化後的資料，識別資料是一個用來識別數位物件的代號，貯藏庫則是用來儲存這些數位物件的儲存設備。Arms 等人在[2]中揭櫫數位圖書館中各構成要素之間的關係，要素之間透過網際網路連接，如圖 2 所示。

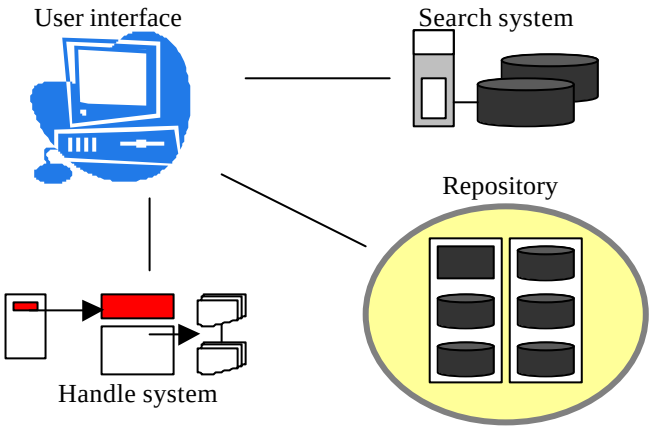


圖 2: 數位圖書館的主要系統構成要素[2]

[2]中重新定義數位物件應包含兩個部份：關鍵性詮釋資料(Key Metadata)及數位資料(Digital Material)，其中關鍵性詮釋資料描述如何在網路環境中操作數位物件的資訊，如儲存(Store)、傳輸(Transmit)，另外也包括識別資料；數位資料則為一連串的位元組，用來儲存數位化後的資料。數位物件如圖 3 所示：

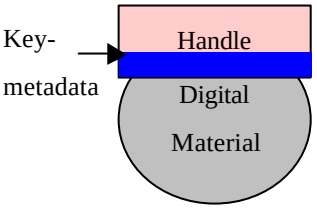


圖 3: 數位物件[2]

除了數位物件之外，[2]也對超物件(Meta Object)作了定義，超物件為數位物件的集合，並利用結構性詮釋資料(Structural Metadata)來描述各物件之間的關聯性資料，我們將在 2.1.2 中作較詳細的介紹。

2.1.2 詮釋資料

詮釋資料是有關資料的結構性資料(Structured Data About Data)，用來描述藏品的資料屬性，目的在促進資訊系統中對資料的檢索、管理與分析[19]，如同傳統圖書館中的編目分類資料。隨著都柏林核心集(Dublin Core，簡稱 DC)[16]的訂定，目前數位圖書館大都以此來處理數位化資料的著錄。都柏林核心集是在 1995 年時由 OCLC(Online Computer Library Center)與 NCSA(National Center for Supercomputing Applications)所共同訂定，目的是訂定網路資源的詮釋資料標準，輔助跨領域網路資源的搜尋。為了提供跨領域的資訊交換，格式必須簡單易用，因此都柏林核心集只定義了基本的十五個欄位(Element)，包括：題名(Title)、創作者(Creator)、主題(Subject)、簡述(Description)、出版者(Publisher)、貢獻者(Contributor)、日期(Date)、類型(Type)、資料格式(Format)、辨識資料(Identifier)、來源(Source)、語文(Language)、關聯(Relation)、時空涵蓋範圍(Coverage)、權限範圍(Rights)。為了符合使用者的不同需求，DC 具有延展性(Extensibility)及可變性(Modifiability)，利用 DC 修飾語(Qualifier)以定義注錄架構、控制詞彙或欄位值，增加詮釋資料的明確性和精確度，不過也因此可能降低了資訊的互通性，所以在使用的時候，應盡量選用正式的修飾語，以達資訊交流的目的。DC 修飾語分為兩大類別：

1. 欄位細分(Element Refinement)：使欄位的語意更加精確(也就是使欄位的定義更加狹義)，例如創作者(欄位，廣義)可能包含作詞者及作曲者(欄位細分，狹義)...等。
2. 編碼結構(Encoding Scheme)：包含控制詞彙(Controlled Vocabularies)及正式標記(Formal Notations)(或解析規則(Parsing Rules))，欄位值必須是用控

制詞彙或由正式標記所定義的字串(如"2000-01-01"是日期的標準表示式)來著錄，編碼結構的定義描述(包含正式標記和控制詞彙)必須引述出處且必須是公開使用的標準。

在本論文中，DC 修飾語只著重予欄位細分這一類別，若無特別註明，修飾語指 DC 修飾語中的欄位細分。

除了前述的描述性詮釋資料，在[2]中另外提出結構性詮釋資料，用來描述超物件中各物件之間的類型(Type)、版本(Version)、關聯(Relationship)和特徵(Characteristics)資料。例如一張經掃描數位化後的照片可能會有三種版本的圖檔，一張為低解析度的縮圖(Thumbnail)，一張為供網路瀏覽的圖檔，另一張則為高解析度的參考(Reference)影像，每一張都是一個數位物件，除此之外，並且建立一個超物件來描述照片及三個數位化版本間的關係，表 1 描述超物件及內含物件包含的資訊：

	數位物件	超物件
關鍵性詮釋資料	描述網路環境中如何操作數位物件的資訊	描述網路環境中如何操作超物件的資訊
結構性詮釋資料	描述特定版本的數位物件之資料類型、版本號、敘述(Description)、用途...等	描述原件及全部版本共通的描述、版本數、用途...等
資料	存放各版本的數位化影像資料	存放所含各版本的識別資料及之間的關聯

表 1: 超物件及內含物件差異比較

圖 4 以一個內含兩個數位物件的超物件為例，假設該超物件的識別資料為"loc.ndlp.amrlp/3a16116"，並含有一個參考影像"loc.ndlp.amrlp/3a16116.1"及縮圖"loc.ndlp.amrlp/3a16116.2"。

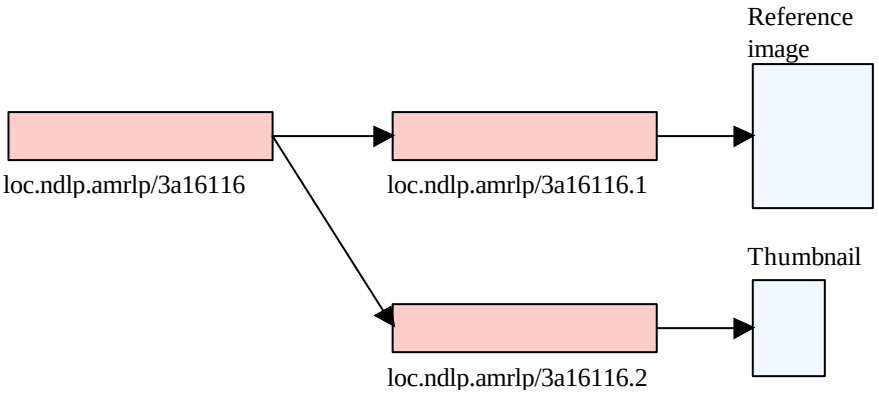


圖 4: **一個內含兩個數位物件的超物件**[2]

數位物件、超物件和結構性詮釋資料之間的關係如圖 5 **所示，*為超物件的結構性詮釋資料，**則為數位物件的結構性詮釋資料。**

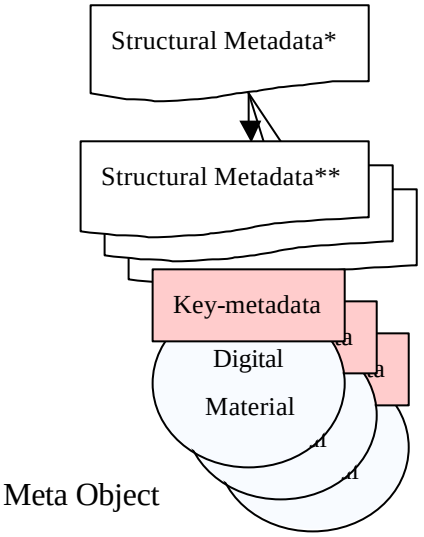


圖 5: **超物件、物件及結構性詮釋資料**

為了開發網路數位資訊系統，詮釋資料有了更廣義的定義，根據功能性(Functionality)**的不同，**Swetland **將詮釋資料分為：管理的**(Administrative)**、描述性的**(Descriptive)**、保存**(Preservation)**、用途**(Use)**和技術性的**(Technical)**等五大類詮釋資料**[7]**。表** 2 **列出五類詮釋資料的定義及功能，依據定義，**DC **屬於描述性的詮釋資料，結構性詮釋資料則屬於用途類詮釋資料。**

類型	定義	例子
管理的 (Administrative)	資源的管理 (Metadata used in managing and administering information resources)	物件權限、位置資訊、版本控制
描述性的 (Descriptive)	資源的描述及識別(Metadata used to describe or identify information resources)	編目資料、超連結、使用者註解
保存 (Preservation)	資源的保存管理(Metadata related to the preservation management of information resources)	資源的實際狀態文件、原件、數位物件的保存文件
用途(Use)	資源的使用層次及類型(Metadata related to the level and type of use of information resources)	展示記錄、使用記錄、內容重複使用及多版本資訊
技術性的 (Technical)	描述系統及詮釋資料如何運作 (Metadata related to how a system functions or metadata behave)	軟硬體文件、數位化資訊(格式、壓縮率)

表 2: 詮釋資料類型定義及功能[7]

第二節　存取控制模型

數位圖書館中收藏大量數位物件，必須要有一套管理系統，使得這些數位物件僅被經由授權的使用者使用，一般稱此管理系統為存取控制系統。至於如何建立一個存取控制機制，最簡單的方式為建立一個(*s,o,p*)模型，定義一個使用者 *s* 可以對物件 *o* 作 *p* 的操作。Lampson提出建立一個使用者和物件之間的存取控制矩陣[12]，矩陣中的每個元素標示某一個使用者是否可以存取某一個物件，存取與否可視為1或0。圖6則是一個較複雜的存取控制矩陣，它分別記錄了R(讀取)、W(寫入)、OWN(擁有)三種權限。一般情況下矩陣會變成一個稀疏矩陣(Sparse Matrix)，而浪費大量記憶空間，雖然可將0去掉，只存放1代表可以存取的權限，甚至利用群組(Group)的方式將具有相同存取權限的使用者設定成群組，但如此將耗費大量時間去作計算。此外大量的物件會使得此一矩陣過於龐大，維護此一矩陣將是一件浩大的工程，例如要新增一筆物件時，必須分別設定每個使用者對該物件的存取權限。以存取控制矩陣為基礎，又衍生出存取控制清單(Access

Control List，簡稱ACL)(如圖7)和存取能力表(Capability)[3](如圖8)，前者將矩陣分成數個行向量，一個行向量代表該件物件所賦予每個使用者的存取權限，後者則將矩陣分成數個列向量，一個列向量代表該位使用者對每件物件的存取權限，這兩個方法都可以減少因為稀疏矩陣所浪費的記憶空間以及利用群組所耗費的計算時間，但是仍然必須針對每個使用者設定其對每個物件的存取權限。

	File 1	File 2	File 3	File 4
John	Own R W		Own R W	
Alice	R	Own R W	W	R
Bob	R W	R		Own R W

圖 6: 存取控制矩陣(Access Control Matrix)[14]

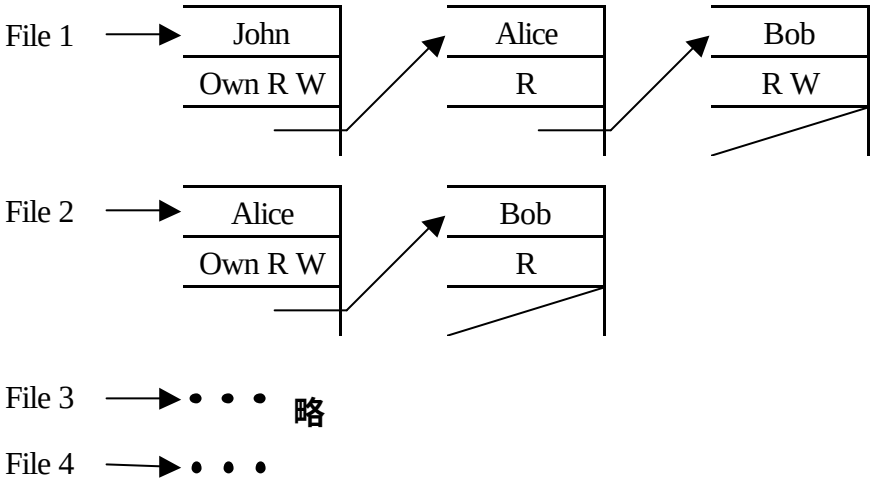


圖 7: 存取控制清單(Access Control Lists)[14]

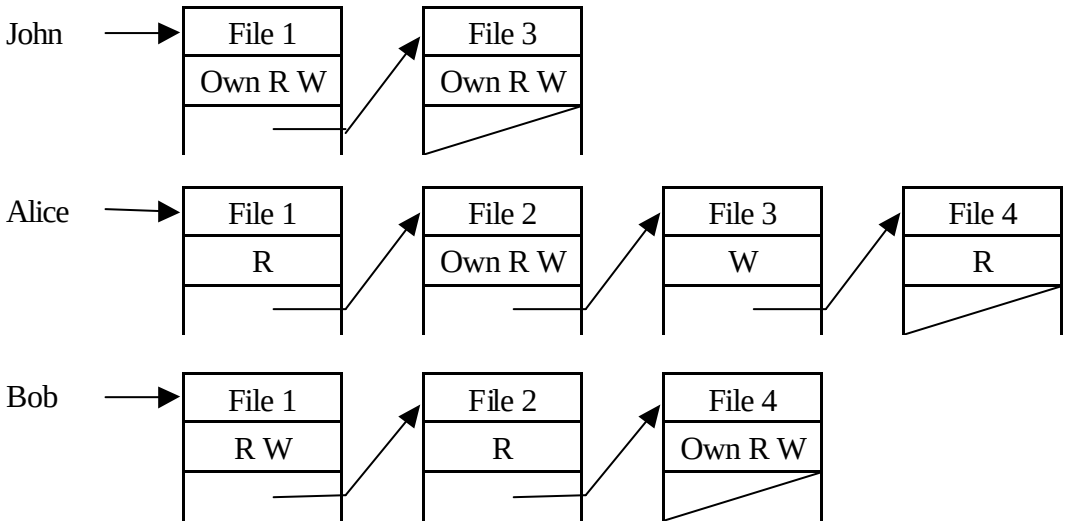


圖 8: **存取能力表**(Capability Lists)[14]

因為針對每個個體(包含使用者及物件)作設定並不夠有效率，Ferraiolo **及** Sandhu **等人提出以角色為基礎的存取控制方法**(Role-Based Access Control**，簡稱** RBAC)[5][14]**，在** RBAC **中，不再針對個體設定權限，取而代之是以角色作為設定權限的單位，使用者依其所扮演的角色而有不同的權限設定，每位使用者並不限定只能扮演一個角色，這個方法使矩陣大幅縮小，在存取控制的設定上也容易許多。在**[14]**中更將** RBAC **分為四個等級，**$RBAC_0$ **定義一個基本模型，**$RBAC_1$ **則在** $RBAC_0$ **的基礎上為角色建立階層**(Hierarchy)**，**$RBAC_2$ **則為** $RBAC_0$ **加上條件限制**(Constraint)**，最後** $RBAC_3$ **整合** $RBAC_1$ **及** $RBAC_2$ **的特色，既有角色階層也有條件限制。**Moyer **等人更進而將此概念延伸至物件及環境**(Environment)**，提出以廣義角色為基礎的存取控制**(Generalized Role-Based Access Control**，簡稱** GRBAC)[13]**，圖** 9 **描述** RBAC **四個等級之間的關係、使用者分派**(User Assignment)**及許可分派**(Permission Assignment)**，此兩種分派皆為多對多的關係，也就是說，使用者可以扮演很多角色，一個角色也包含了很多人；同樣地，一個角色可以有很多許可，一個許可可以分派給很多角色，**RBAC **的關鍵就建立在這兩種關係上。但是在數位圖書館環境中，大量的使用者及物件的多變性，要改變或新增存取設定，即須對角色做修改或新增的動作，並對使用者及物件作角色上的分派，**

如此無形中增加了管理上的負擔。

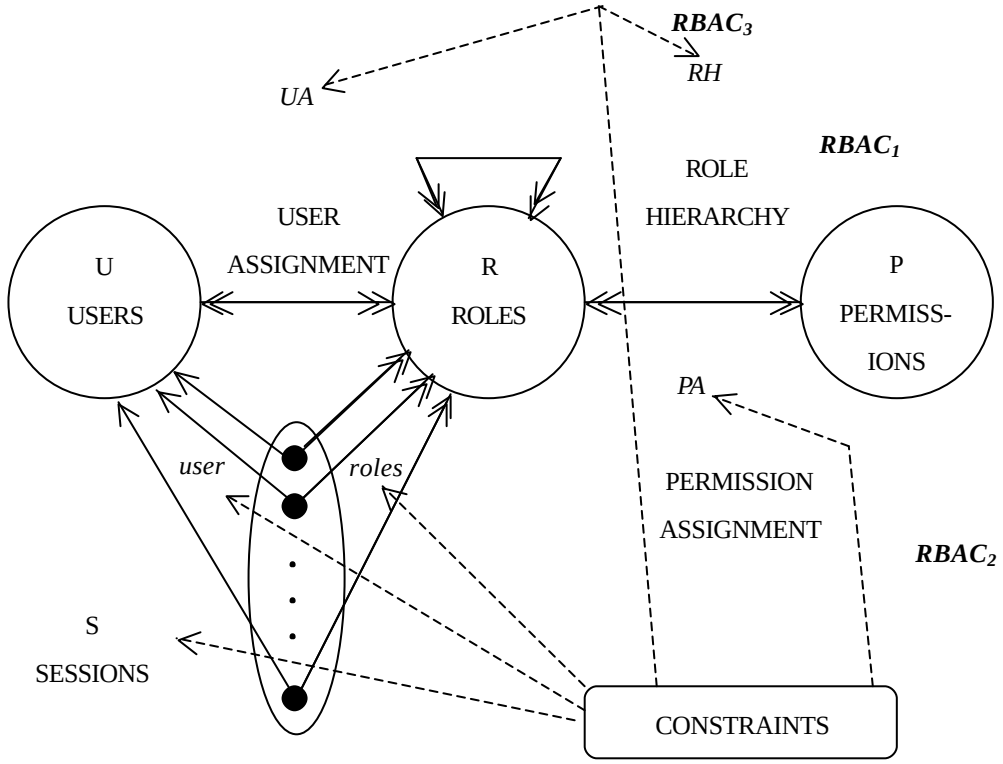


圖 9: RBAC 模型[14]

為了能更適合地將存取控制套用於數位圖書館的環境中，Adam 等人提出了以內容為基礎的授權模型(Content-Based Authorization Model，簡稱 CBAM)[1]。CBAM 並非是第一個提出以內容為基礎的存取控制模型，在[3]、[7]裡即有在物件導向資料庫(Object-Oriented Database)中應用此構想，但是兩者都只能處理結構性資料(Conventional, Structured Data)(也就是說，結構必須和資料庫綱要(Schema)完全吻合)，在這類模型中，存取控制可以利用為欄位值設立條件來建立。舉例來說，使用者 *u* 只能存取收入 *salary* 低於三萬的員工資訊 *employees* 可以用 (*u, employees, salary<30k*)來表示；在 CBAM 中，必須根據員工資訊內容作判斷是否符合該條件；而在 GRBAC 中，則必須建立一個角色 *R*，讓收入低於三萬的員工資訊可以扮演該角色，並訂定一條規則為所有使用者可以存取角色 *R* 的物件。由上述的例子可以看出，CBAM 可以減少建立角色和角色分派的管理工作，更加適合應用在數位圖書館的環境中。CBAM 在物件的管理上，因為是建立在

內容的基礎上，所以只適合用於文字資料型態的物件，其他非文字資料型態的物件如多媒體資料等，則必須利用詮釋資料所注錄的內容描述萃取(Extract)出該物件的概念(Concepts)。

表 3 為先前提及之各種存取控制模型及其所控制物件之內容間的關連性。傳統模型大都只專注於物件本身，利用物件的識別資料(Identifier)來作判斷，物件之能否存取和內容沒有關係，而 CBAM 則兩者兼具，提供以內容來作為能否存取的依據。

	和物件內容無關 (Content-independent)	和物件內容有關 (Content-dependent)
(s, o, p)模型	v	
存取控制矩陣 (Access Control Matrix)	v	
存取控制清單 (Access Control List)	v	
存取能力表(Capability)	v	
以角色為基礎的存取控制 (RBAC, GRBAC)	v	
和物件內容相依的存取控制 [3][7]		v
以內容為基礎的存取控制 (CBAM)	v	v

表 3: 存取控制模型和物件內容相關性

表 4 則是比較表 3 中最後兩種存取控制模型所能控制的物件結構性，結構性物件定義如前文所述，非／半結構性物件則泛指文字、多媒體等資料檔案。

	結構性物件 (Structured data)	非結構性物件／半結構性物件 (Unstructured/Semistructured)
和物件內容相依的存取控制[3][7]	v	
以內容為基礎的存取控制 (CBAM)		v*

表 4: 存取控制模型和物件結構性比較
*在[1]中只討論文字性物件

第三節　以內容為基礎的存取控制

CBAM 和傳統的(*s,o,p*)模型不同之處在於，CBAM 利用(*credentials, concepts, privilege*)來描述一個授權。使用者身份(Credentials)相當於使用者的特徵(Characteristics)和資格(Qualifications)。概念(Concepts)指的是使用者期望從數位物件中找到的抽象概念(Abstractions)或想法(Notions)，換句話說，概念是對物件內容的簡潔敘述；然而，概念不僅僅是關鍵字(Keywords)而已，例如一篇令讀者心酸的文章可能在文中並沒有任何有關悲傷的字眼出現。權力(Privileges)則分為瀏覽的(Browsing)和著作的(Authoring)權力。

2.3.1 物件(DL Objects)

一個數位物件包含非結構性的媒體類型資訊(如：文字、圖形、影片)及唯一的物件識別(Object Identifier)。物件內部不同的部分通常有不同的保護需求，例如一篇論文中，摘要可以給任何人使用，而其他部分則僅供訂閱的人使用，在 CBAM 中，物件分成數個區段(Slots)，區段和物件相似，可以包含不同類型的資訊，並且可以給定識別資料。物件亦可利用鏈結(Links)來描述其他相關物件。所以 CBAM 中利用(*i, slots, links, concepts*)來描述一個物件，其中 *i* 為物件的識別，而 *concepts* 為與物件直接相關的概念。CBAM 利用 *C(dlo)*來表示物件中所有相關概念的集合，這個集合包含了在 *concepts* 所描述的概念及所有比 *concepts* 更普遍(General)的概念，正式的定義為 $C(dlo) = concepts \cup \{cp \mid cp \in CP \text{ such that } \exists\, cp' \in concepts, \text{ with } cp' \pi_{CP} cp\}$。通常來說，一個特定領域內的概念(以 *CP* 表示)互有關聯性，因此可以建立一個概念階層，階層內的概念具有局部次序(Partial Order)的關係，以 π_{CP} 表示，假設兩個屬於 *CP* 的概念 cp_1 和 cp_2，以 $cp_1 \pi_{CP} cp_2$ 表示 cp_1 具有比 cp_2 更具體(Specific)的概念。圖 10 為一個物件概念階層的例子，其中具有 Imports Tax π_{CP} Import－Export 、 Imports Tax π_{CP} GLIN Legal Document ...等關係。

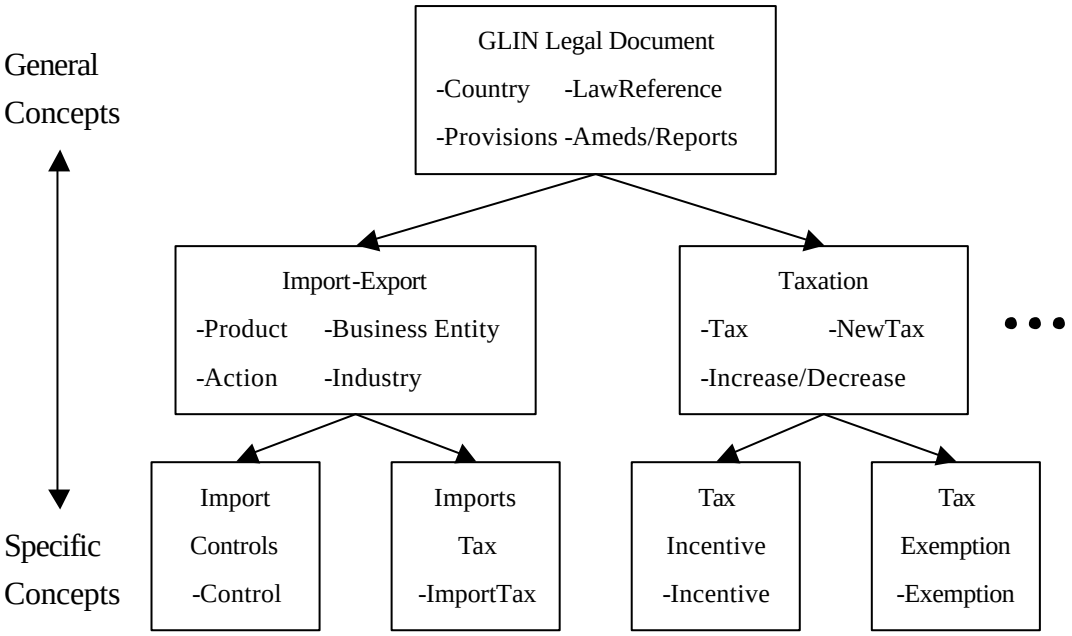


圖 10: **概念階層**

CBAM **利用**(*credentials, concepts, privilege*)**來描述一個授權，在** *concepts* **部分是以實體**(Entity)**為授權單位，實體可以是特定的物件**(**藉由列舉**(List)**物件的識別)、包含特定概念的物件**(**藉由概念表示式**(Conceptual Expression)**來表達**)**或物件中的區段或鍊結。實體的正式定義如下：**

定義 2.1 (Entity Specification). *An entity specification has one of the following two forms:*

1. co-spec.slot-spec, *where* co-spec *is either a conceptual expression in COEX, or a set, possibly empty, of object identifiers in* 2^{OI}, *and* slot-spec$\in 2^{SN}$ *is a set, possibly empty, of slot names, or*
2. link-spec, *where* link-spec$\in 2^{LI}$ *is a set, possibly empty, of link identifiers.*

2.3.2 **使用者身份**(Credentials)

使用者身份為使用者的屬性集合，為了方便使用者身分的描述，CBAM **將具有相似結構**(Structure)**的身分群組化，建立使用者身份類型**(Credential-types)**，用**(*ct-id, attr*)**來表示，** *ct-id* **為使用者身份類型的識別；** *attr* **為**(*a_name, a_dom, a_type*)**的集合，其中** *a_name* **為屬性的名稱，** *a_dom* **為屬性的領域**(Domain)**，如**

整數、字串...等，*a_type* 表示該屬性是(*a_type*="opt")否(*a_type*="mand")允許空值(Null)，並用 *CT* 來表示 *ct-id* ***的集合***。例如一個職員的身分類型如下：(employee, {(age, string, opt), (address, string, mand), (salary, integer, opt), (nationality, string, mand), (national origin, string, mand)})。如同概念階層，使用者身份類型亦可建立階層，階層內的類型具有局部次序的關係，以 π_{CT} 表示，假設兩個屬於 *CT* 的類型 ct_1 和 ct_2，以 $ct_1\ \pi_{CT}\ ct_2$ 表示 ct_1 為 ct_2 的一個子類型，ct_1 繼承(Inherit)ct_2 的所有屬性，並可包含額外的屬性。圖 11 為一個使用者身份類型階層的例子，其中具有 Legal Research Analyst π_{CT} LLoc employee、LLoc employee π_{CT} employee...等關係。

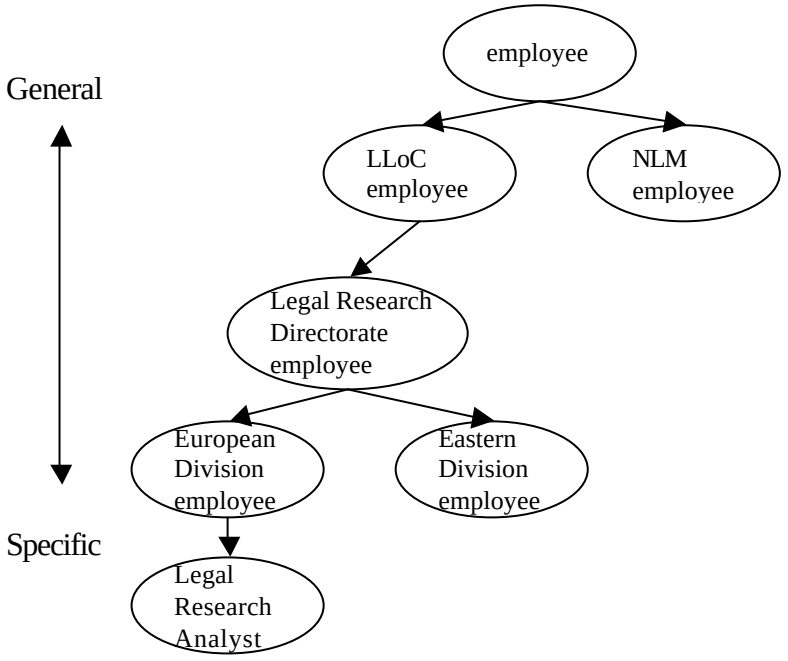


圖 11: 使用者身份類型階層

一個使用者身分為使用者身份類型的個體(Instance)，用(*c_id, user_id, state, ct_id*)來表示，*c_id* 為使用者身份的識別；*user_id* 為使用者的識別；*state* 為(a_1: v_1, ... a_n: v_n)的集合，a_1, ... a_n 為屬性名稱，v_1, ... v_n 為對應的值；*ct_id* 為使用者身份類型的識別。在下例中，legal research analyst 這個身分類型比 employee 多了 project 這個屬性資料：

(c_1, Bob, (age: null, address: Queen Street, salary:2,000, nationality: US, national

origin: Italy), employee)

(c_2, Ann, (age: 29, address: Broad Street, salary:null, project:P125, nationality: US, national origin: US), legal research analyst)

CBAM **利用**(*credentials, concepts, privilege*)**來描述一個授權，在** *credentials* **部分是以使用者身份為授權單位，使用者身份可以是特定的使用者(藉由列舉使用者的識別)或包含特定特徵的使用者(藉由使用者身份表示式**(Credential Expression)**來表達)。使用者身份的正式定義如下：**

> **定義 2.2 (Credential Specification).** *A credential specification is either a set of user identifiers in* 2^U*, or a credential expression in CE.*

例如可以限制某些物件為必須大於 18 **歲的使用者方可使用，或必須具備** legal research analyst **這個身份且參與** p_1 **這個計畫才能使用，分別用**(X.age > 18)**和**(legal research analyst(X) ^ X.project = p_1)**來表示大於** 18 **歲和具備** legal research analyst **身份且參與** p_1 **計畫的人，**(X.age > 18)**即為一個使用者身份表示式。若某些使用者作為判斷條件的屬性欄位為空值，以** *Undef* **來表示，則這些使用者不屬於滿足該條件的使用者之集合；另外用** *Denotes* **表示滿足該條件的使用者之集合，以先前的例子來說：**

Denotes(employee) = {Ann, Bob};

Denotes(X.age > 18) = {Ann};

Denotes(legal research analyst(X) ^ X.project = P125) = {Ann};

Undef(X.age > 18) = {Bob};

Undef(employee) = ϕ .

2.3.3 **權力**(Privileges)

權力分為瀏覽的和著作的權力，瀏覽的權力有三種不同的類型：觀看(View)**、鏈結**(Link)**和觀看全部**(View-All)**，著作的權力也有三種不同的類型：參考**(Refer)**、附加**(Append)**、更新**(Update)**，其意義如表** 5 **所述。觀看全部內含觀**

看和鏈結的權力，更新內含參考和附加的權力。

類型	權力	意義
瀏覽的 (Browsing)	view	觀看物件或區段的資訊，不包括鏈結的資訊
	link	觀看鏈結的資訊
	view-all	觀看物件、區段或鍊結的資訊
著作的 (Authoring)	refer	為物件或區段加入鏈結
	append	為物件或區段寫入資訊，但不能刪除原有的資訊
	update	修改或刪除物件的內容

表 5: 權力的種類及其意義

2.3.4 授權模型(Authorization Model)

一個完整的授權包含四個元素：使用者、實體、權力和正/負向符號，定義如下所述：

> **定義 2.3 (Authorization).** *An authorization is a 4-tuple(crd-spec, ent-spec, priv, sign), where* crd-spec *is a credential specification denoting the users to whom the authorization is granted,* ent-spec *is an entity specification denoting the contents, objects, and/or the slots or links to which the authorization refers, priv* ∈ *P is the privilege for which the authorization is granted, and sign* ∈ {+,-} *indicates whether the authorization is positive*(+) *or negative*(-).

要表示適用某授權 A 之人及物分別用 Denoted Users 及 Denoted Objects 來表示，利用 2.3.2 所述 *Denotes* 及 *Undef* 來作定義，以 Denoted Users 為例來說明，若已明確描述所套用使用者識別並不會發生問題；而用使用者身份表示式來描述所套用的使用者時，假設如下情形：

Denotes(X.age > 18) = {Ann}

Undef(X.age > 18) = {Bob}

則當此授權為十八歲以上方能使用(正向授權)時，只有 Ann 可以使用；當此授權為十八歲以上禁止使用(負向授權)時，則 Ann 不能使用是無庸置疑的，但 Bob 則因為其年齡為空值，依保守原則，我們也應將此授權套用至 Bob 身上，換句話說，Bob 亦不能使用該物件。Denoted Users 與 Denoted Objects 的正式定義如

下：

定義 2.4 (Denoted Users). *Let A be an authorization. The set of users to which A applies, denoted as U_A, is defined as follows:*

- If crd-spec(A) ∈ 2^U, then U_A = crd-spec(A);
- If crd-spec(A) ∈ *CE*, then:
  If sign(A) = "+", then U_A = *Denotes*(crd-spec(A));
  If sign(A) = "-", then U_A = *Denotes*(crd-spec(A)) ∪ *Undef*(crd-spec(A)).

定義 2.5 (Denoted Objects). *Let A be an authorization. The set of IDs of objects to which A applies, denoted as O_A, is defined as follows:*

- If co-spec(ent-spec(A)) ∈ 2^{OI}, then O_A = co-spec(ent-spec(A));
- If co-spec(ent-spec(A)) is a conceptual expression, then O_A = Obj(co-spec(ent-spec(A)));
- If co-spec(ent-spec(A)) = ϕ, then:
  1. if slot-spec(ent-spec(A)) ? ϕ, then O_A = {i' | (i, slots, links, concepts) ∈ *OB*, i = i' and slots ∩ slot-spec(ent-spec(A)) ? ϕ};
  2. if link-spec(ent-spec(A)) ? ϕ, then O_A = {i' | (i, slots, links, concepts) ∈ *OB*, i = i' and links ∩ link-spec(ent-spec(A)) ? ϕ}.

The *Object Base*, denoted by *OB*, is the set of all dlos.

雖然負向授權並非一定要存在，但使用負向授權可以使存取控制描述更具彈性，例如一個簡單的授權可以輕易地使用反向授權來描述，如前述十八歲以上方能使用的授權(正向授權)為例，我們將之改成十八歲以下不能使用(負向授權)，兩者為等意的授權；但一個用來描述例外的授權若使用同樣的方式將會使過程變得瑣碎，例如有個授權為具 legal research analyst **身份的人不能使用，若不使用負向授權，則必須使用** n-1 **個(假設共有** n **種情形)具有其他身份可以使用的授權來描述。雖然使用負向授權有很多好處，但相對也帶來問題，當一個使用者對一個物件作操作時，若同時擁有正向及負向授權，則依據正向授權該使用者可操作該物件，依據負向授權該使用者卻不能操作該物件，此類衝突必須有效解決，才能使存取控制機制更有效地運作。在** CBAM **中，定義了較具權威性的授權**(Stronger

Authorization)(**定義** 2.8)**作為解決的辦法，所謂較具權威性指的是該授權擁有較高的優先權，但在這之前，必須先定義較具權威性的使用者描述**(Stronger Credential Specification)(**定義** 2.6)**和實體描述**(Stronger Entity Specification)(**定義** 2.7)**，分別敘述如下：**

定義 2.6 (Stronger Credential Specification). *Let u be a user. Let* A_1 *and* A_2 *be two authorizations such that* $u \in U_{A1} \cap U_{A2}$. crd-spec(A_1) *is stronger than* crd-spec(A_2) *with regard to user u, written* crd-spec(A_1) $>_u$ crd-spec(A_2), *iff one of the following conditions holds:*

1. crd-spec(A_1) $\in 2^U$ *and* crd-spec(A_2) $\in$ *CE*.
2. crd-spec(A_1), crd-spec(A_2) $\in$ *CE and* $\forall$ $ct_2 \in$ *C_types*(crd-spec(A_2)) $\cap$ *CT*(u) $\exists$ $ct_1 \in$ *C_types*(crd-spec(A_1)) $\cap$ *CT*(u) such that $ct_1 \pi_{CT} ct_2$.

定義 2.7 (Stronger Entity Specification). *Let dlo be an object. Let* A_1 *and* A_2 *be two authorizations such that i(dlo)* $\in O_{A1} \cap O_{A2}$. ent-spec(A_1) *is stronger than* ent-spec(A_2) *with regard to object dlo, written* ent-spec(A_1) $>_{dlo}$ ent-spec(A_2) *iff one of the following conditions holds:*

1. co-spec(ent-spec(A_1)) $\in 2^{OI}$, co-spec(ent-spec(A_2)) $\in 2^{OI}$, ent-spec(A_1) *and* ent-spec(A_2) *are of the form* ent-spec.slot-spec *and* slot-spec(ent-spec(A_1)) $\neq \phi$, *whereas* slot-spec(ent-spec(A_2)) $= \phi$.
2. co-spec(ent-spec(A_1)) $\in 2^{OI}$ *and* co-spec(ent-spec(A_2)) *is a conceptual expression.*
3. co-spec(ent-spec(A_1)) *and* co-spec(ent-spec(A_2)) *are conceptual expressions and* $\forall cp_2 \in$ Concepts(co-spec(ent-spec(A_2))) $\cap$ C(dlo) $\exists cp_1 \in$ Concepts(co-spec(ent-spec(A_1))) $\cap$ C(dlo) *such that* $cp_1 \pi_{CP} cp_2$.
4. co-spec(ent-spec(A_1)) *and* co-spec(ent-spec(A_2)) *are conceptual expressions such that the condition in point 3 does not hold, and* slot-spec(ent-spec(A_1)) $\neq \phi$, *whereas* slot-spec(ent-spec(A_2)) $= \phi$.
5. ent-spec(A_1) *is of the form link-spec, whereas* ent-spec(A_2) *is of the form* co-spec.slot-spec, *with* slot-spec $= \phi$.

定義 2.8 (Stronger Authorization). *Let u be a user, and let dlo be an object. Let* A_1, A_2 *be two authorizations such that* $u \in U_{A1} \cap U_{A2}$ *and i(dlo)* $\in O_{A1} \cap O_{A2}$. *Authorization is stronger than authorization with regard to u and dlo, written* $A_1 >_{u,dlo} A_2$, *iff one of the following conditions holds:*

1. crd-spec(A_1) $>_u$ crd-spec(A_2).
2. crd-spec(A_1) *and* crd-spec(A_2) *are incomparable with regard to the* $>_u$ *relation and* ent-spec(A_1) $>_{dlo}$ ent-spec(A_2).
3. crd-spec(A_1) *and* crd-spec(A_2) *are incomparable with regard to the* $>_u$ *relation,* crd-spec(A_1) *and* crd-spec(A_2) *are incomparable with regard to the* $>_{dlo}$ *relation, and* priv(A_1) π_P priv(A_2).
4. crd-spec(A_1) *and* crd-spec(A_2) *are incomparable with regard to the* $>_u$ *relation,* crd-spec(A_1) *and* crd-spec(A_2) *are incomparable with regard to the* $>_{dlo}$ *relation,* priv(A_1) *and* priv(A_2) *are incomparable with regard to the* π_P *relation and* sign(A_1)="-," *whereas* sign(A_2)="+."

第三章　以詮釋資料為基礎之存取控制

存取控制用來管理使用者對物件是否擁有存取權，本論文提出適用於數位圖書館環境中以詮釋資料為基礎之存取控制(Metadata-Based Access Control, 簡稱MBAC)。MBAC主要修改CBAM中以概念來描述物件的方法，改以詮釋資料來描述物件。本章第一節介紹適合此方法的詮釋資料模型及利用詮釋資料來描述存取控制中的物件，第二節則介紹完整的存取控制模型及衝突解決策略。

第一節　基本概念

3.1.1 詮釋資料模型(Metadata Model)

數位圖書館利用詮釋資料來描述數位物件，但是通常只用來描述藏品原件(Original)的特性，代理物件(Surrogate)如圖檔、聲音和影片等的詮釋資料則因為在資料檢索上較不重要，所以在著錄藏品資料的過程中常常會被忽略。但將詮釋資料應用在存取控制時，代理物件的詮釋資料則非常重要，因為存取控制的對象往往為代理物件。本論文中用於存取控制的詮釋資料不只包括原件的詮釋資料(如創作者、權限範圍)，也包括代理物件的詮釋資料(如創作者、權限範圍、資料格式)，但所操作的對象(指用來作存取控制的物件)，則以代理物件為主。綜合上述，以原件的詮釋資料為存取控制的條件時，所有該原件相關的代理物件皆受到該條件所控制；當以代理物件的詮釋資料為存取控制的條件時，則只有符合該條件的代理物件受到該條件控制。舉例來說，若一存取控制對象為所有蘇森墉的作品，則此乃以原件的創作者為限制條件，因此所有作品包括樂譜圖檔、聲音、影像...等，都將受該存取控制所限制；在另一方面，代理物件也有創作者，例如圖檔掃描製作者等，但一般來說不具有作存取控制的意義。而以資料格式來說，雖然原件也有此欄位，例如紙張、錄影帶等，但反過來說原件的資料格式通常亦不具作存取控制的意義。本論文的討論將以原件的詮釋資料(如創作者、權限範圍)

及代理物件的詮釋資料(如資料格式)來作討論，依據 Swetland 對詮釋資料的分類，本模型需要原件的描述性詮釋資料和代理物件的技術性詮釋資料。此外，若必須使用原件及代理物件相同的詮釋資料欄位來作存取控制(如創作者)，則必須明確描述，以識別其為以原件或代理物件的詮釋資料為根據。

因為 MBAC 將利用原件及代理物件的詮釋資料，所以在著錄藏品資料的過程中，除了描述原件藏品外，亦需為代理物件建立詮釋資料。但是若為大量藏品建立其代理物件的詮釋資料，將使著錄的工作份量倍增，如此將耗費大量人力及時間，因此為求實用性，代理物件的詮釋資料可盡量簡化，或只著錄 MBAC 所需利用的詮釋資料即可。圖 12 為此詮釋資料模型分別針對單一代理物件及多重代理物件的簡單示意圖，其中多重代理物件和 2.1.2 所述之超物件類似，但所注錄的詮釋資料則有所不同，例如代理物件的詮釋資料除了結構性詮釋資料外，還加上技術性詮釋資料。

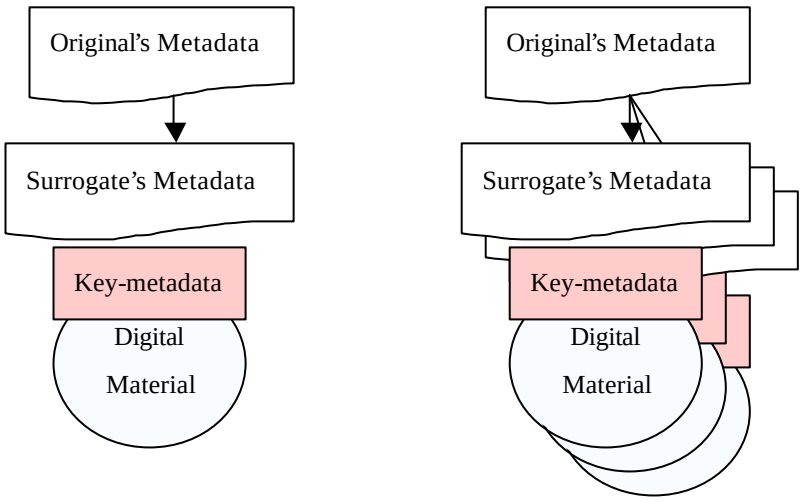


圖 12: 單一代理物件 及 多重代理物件 詮釋資料模型示意圖

3.1.2 實體描述

DC 的格式雖然簡單，但為了符合使用者的不同需求，DC 具有延展性及可變性，為了能更精確對代理物件作存取控制，本詮釋資料模型支援 DC 的欄位修飾語，我們為個別 DC 欄位建立修飾語階層(Qualifier Hierarchy)。階層內的修飾

語具有局部次序的關係，以 π_{MQ} 表示，若修飾語 mq_1 具有比 mq_2 更特定的描述，以 $mq_1\ \pi_{MQ}\ mq_2$ 表示。圖 13 及圖 14 分別是以資料格式及創作者來建立修飾語階層的範例，其中，資料格式因為媒體類型(Medium)不同而分為 JPEG、TIFF、WMV...等，而 JPEG 則又因解析度的不同分為高解析度及低解析度，高解析度為代理物件的高品質圖檔，低解析度則為代理物件的縮圖；創作者則可分為作詞者(Songwriter)、作曲者(Composer)、編曲者(Arranger)...等；其中具有

Low res. π_{MQ} JPG 、 Low res. π_{MQ} Format 和 Songwriter π_{MQ} Creator 等關係。

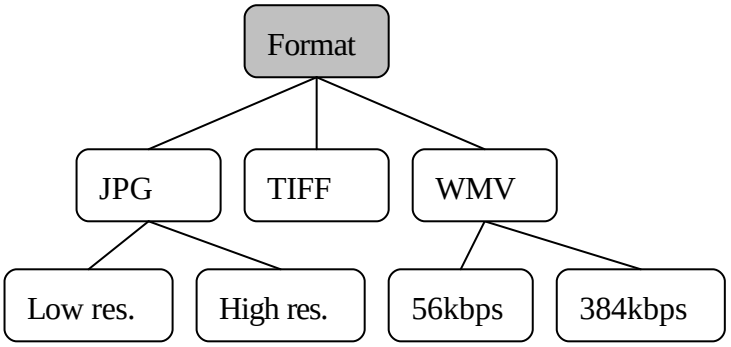


圖 13: 資料格式修飾語階層(Qualifier Hierarchy of FORMAT element)

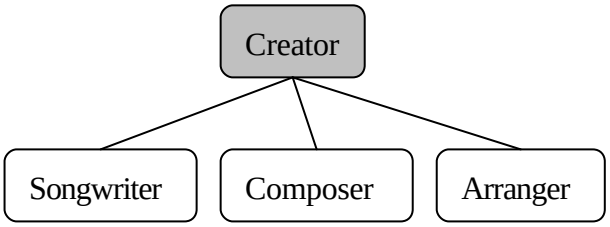


圖 14: 創作者修飾語階層(Qualifier Hierarchy of CREATOR element)

如 3.1.1 所述，描述一個數位物件包含了原件及代理物件的詮釋資料，我們用(*o_id, o_name, state*)來表示，*o_id* 為數位物件的識別；*o_name* 為此物件的名稱；*state* 為(a_1=v_1, ... a_n=v_n)的集合，a_1, ... a_n 為詮釋資料修飾語名稱，v_1, ... v_n 為對應的值。例如以下 Ex.1、Ex.2 及 Ex.3 三個數位物件：

天烏烏 樂譜 圖檔

(S004014，**天烏烏**，(Arranger=’**蘇森塘**’，Medium=’ JPEG’))..... [Ex.1]

天烏烏 合唱 影片

(M004014，**天烏烏**，(Arranger=’**蘇森塘**’，Medium=’WMV’，

BitRate=’384kbps’))..... [Ex.2]

竹中校歌 合唱 影片

(M003001，**竹中校歌**，(Arranger=null，Medium=’WMV’，

BitRate=’384kbps’))..... [Ex.3]

MBAC **利用**(*credentials, metadata, privilege*)**來描述一個授權，在** *metadata* **部分是以實體**(Entity)**為授權單位，實體可以是特定的物件**(**藉由列舉**(List)**物件的識別**)**或包含特定詮釋資料的物件**(**藉由詮釋資料表示式來表達**)**。我們修改定義** 2.1 **重新定義實體為：**

> **定義 3.1 (實體描述, Entity Specification)。物件描述可能為一個物件集合** 2^{OI} **或者為詮釋資料表示式。** 2^{OI} **為物件識別的集合**

詮釋資料表示式為一個物件集合，例如所有蘇森塘的作品，以

Composer=’**蘇森塘**’ ∨ Songwriter=’**蘇森塘**’ ∨ Arranger=’**蘇森塘**’ ..... [C.1]

或 Creator=’**蘇森塘**’..... [C.2]

來表示；而若要表示所有蘇森塘的表演影片，則可以

Creator=’**蘇森塘**’ ∧ Medium=’WMV’ ..... [C.3]

來表示。

由於存取控制利用詮釋資料的欄位或修飾語來作限制，當有些欄位或修飾語為空值時，則此物件不屬於該特定描述的集合，例如 Ex.1、Ex.2 **符合** C.1 **所描述的集合，**Ex.2 **符合** C.3 **所描述的集合，但是** Ex.3 **卻都不屬於** C.1 **及** C.3 **的集合。在此，我們利用** CBAM **所述** *Denotes* **及** *Undef* **來表示，例如：**

Denotes(Creator=’**蘇森塘**’) = {Ex.1、Ex.2}

Denotes(Medium=’WMV’) = {Ex.2、Ex.3}

Denotes(Creator=’**蘇森塘**’ ∧ Medium=’WMV’) = {Ex.2}

Undef(Creator=’**蘇森塘**’ ∧ Medium=’WMV’) = {Ex.3}

第二節　存取控制模型

3.2.1　存取控制描述

在 MBAC 中，一個完整的授權如同 CBAM 中的一樣，包含四個元素：使用者(定義 2.2)、實體、權力和正/負向符號，其中我們重新對實體作定義(定義 3.1)，授權的定義如下所述(沿用定義 2.3)：

> **定義 3.2 (授權, Authorization)**[1]。一個存取控制由四個元素所構成 (crd-spec, ent-spec, priv, sign)，其中 crd-spec(Credential Specification)用來描述該授權所授予權力的使用者；ent-spec(Entity Specification)用來描述該授權所套用的物件；priv(Privilege)表示該授權所授予的操作權限為何；sign 則用來表示該授權為正向或負向授權，分別用"+"及"-"來表示。

所以，在 MBAC 中，授權亦可為正向授權或負向授權，正向授權表示可允許存取，負向授權表示拒絕存取。底下是授權的範例：

1: (School=' NCTU' ∧ Occupation=' UnderGraduate'，

Creator='蘇森塘' ∧ Medium=' JPG' ∧ Resolution=' LOW'，View，+)

2: (School=' NTHU'，

Creator='蘇森塘' ∧ Medium=' JPG' ∧ Resolution=' LOW'，View，-)

授權 1 表示所有交大大學部的學生可以瀏覽蘇森塘低解析度的樂譜圖檔，授權 2 則表示所有清大的學生不可以瀏覽蘇森塘低解析度的樂譜圖檔。因為存取控制除了可以明確利用識別資料作辨識外，亦可利用使用者的身分資料或是物件的詮釋資料來做權限控管，所以確認存取控制所套用的人及物是非常重要的。若有使用者或實體描述所參考的資料欄位為空值的情形，我們將無法判斷該人/物是否應該被該授權所控制。在 CBAM 中利用 Denoted Users 及 Denoted Objects 來表示受存取控制所規範的人及物，根據 MBAC 的需求，我們修改定義 2.5 重新定義 Denoted Objects 為：

定義 3.3 (Denoted Objects)。假設 A **為一個存取授權，所有符合** A **的數位物件必須具備下述條件，用** O_A **來表示：**

- If ent-spec(A) ∈ 2^{OI}, then O_A = ent-spec(A)
- If ent-spec(A) ∈ entity expression, then:
  If sign(A) = "+", then O_A = *Denotes*(ent-spec(A));
  If sign(A) = "-", then O_A = *Denotes*(ent-spec(A))
  ∪ *Undef*(ent-spec(A)).

crd-spec(A), ent-spec(A), sign(A) **分別表示一個授權** A **中的使用者身份描述、實體描述及正向或負向授權。**

3.2.2 衝突解決策略

負向授權使存取控制描述更具彈性，但是也會帶來問題，當一個存取要求同時擁有正向及負向授權時，系統將無法決定是否要接受或拒絕此要求。此類衝突必須有效解決，才能使存取控制機制更有效地運作。本論文提出一套衝突解決策略，利用使用者身份類型、詮釋資料及權力的階層觀念來解決這個問題，此策略包含下列四點原則：

1. **根據使用者身份類型階層，擁有越特定描述的授權擁有較高的優先權。**
2. **詮釋資料中擁有越特定描述的授權擁有較高的優先權。**
3. **根據權力階層，擁有越特定描述的授權擁有較高的優先權。**
4. **當以上都無法解決時，負向授權優先採用。**

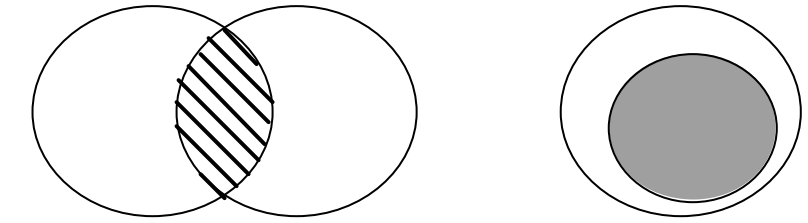
假設授權 3 **如下**

3: (School='NCTU'，
Creator='**蘇森塘**' ∧ Medium='JPG' ∧ Resolution='LOW'，View，-)

則若有一交大大學部的學生欲瀏覽蘇森塘低解析度的樂譜圖檔，此時授權 3 **和授權** 1 **同時符合此存取要求，因兩者一為正向授權，另一為負向授權，此時會產生衝突，但是因為** NCTU Undergraduate π_{CT} NCTU **，所以根據第** 1 **點原則，授權** 1 **擁有較高的優先權。**

因為詮釋資料修飾語階層不像使用者身份或權力為單一階層，單一欄位及欄

位修飾語自成一個階層，如圖 13 及圖 14，因此詮釋資料中何謂擁有越特定的描述？首先舉例說明詮釋資料表示式的衝突，及兩種衝突的類型(如圖 15)，再進而闡述何謂越特定的詮釋資料表示式。



(a)inter-hierarchy (b)intra-hierarchy

圖 15: 跨階層及單一階層的衝突

例如有下列兩個存取控制分別為：

4: (School=’ NCTU’，Creator=’蘇森塘’，View，+)

5: (School=’ NCTU’，Medium=’WMV’，View，-)

則若一個使用者要瀏覽蘇森塘的影片，系統將無法判斷是否要同意其存取要求，此情形屬於圖 15(a)的衝突(FORMAT 和 CREATOR 兩種階層的衝突)；而若是如下列兩個存取控制：

6: (School=’ NCTU’，Medium=’WMV’，View，+)

7: (School=’ NCTU’，Medium=’WMV’ ∧ BitRate=’ 384kbps’，View，-)

則依據資料格式修飾語階層，授權 7 擁有越特定的描述，因此拒絕使用者的存取要求，此情形屬於圖 15(b)的衝突(FORMAT 階層內的衝突)。由此可知跨兩個階層以上的衝突比單一階層的衝突較難克服，為了解決跨階層的衝突，底下為四種可行的方法：

1. 為個別存取控制設定優先權
2. 為詮釋資料中的欄位設定優先權
3. 依據修飾語階層，只比較衝突的存取控制中相同的欄位
4. 為詮釋資料中的修飾語設定權重

方法 1 為最根本的解決辦法，若不存在兩個具有相同優先權的存取控制，則衝突

即可迎刃而解，但是此方法也大幅增加了管理上的負擔，如優先權的給定。方法 2 則修改方法 1，改為設定詮釋資料欄位的優先權，若一存取控制含有較高優先權的詮釋資料欄位，則優先採用該存取控制，但因為 DC 中每個欄位實為獨立的描述，彼此相關性極小，因此如何設定優先權將是一個困難的課題。方法 3 則類似單一階層的衝突解決辦法，比較有所衝突的存取控制中相同的欄位階層，但若沒有相同的欄位則無法用此方法判斷。方法 4 則改採計權重的方式，欄位修飾語階層中每個子節點權重皆為父節點權重的 a 倍，每個存取控制的詮釋資料表示式權重為所有修飾語權重的總和，擁有較高的權重其優先權也越高，也就是說，擁有越特定的詮釋資料表示式，正式定義為：

> **定義 3.4 (較特定的詮釋資料表示式)。** 假設 dlo 為數位物件，A_1 和 A_2 為兩個授權，ent-spec(A_1)和 ent-spec(A_2)皆為詮釋資料表示式，且 dlo 為 A_1 和 A_2 的 denoted object，若 ent-spec(A_1)中的詮釋資料修飾語權重$\sum W_1$ 高於 ent-spec(A_2)中的權重$\sum W_2$，則說 ent-spec(A_1)比 ent-spec(A_2)具有更特定的詮釋資料描述。

如資料格式修飾語的權重給定 a = 10，則前述授權 6 和授權 7 其權重依圖 16 計算分別為 1 和 11。

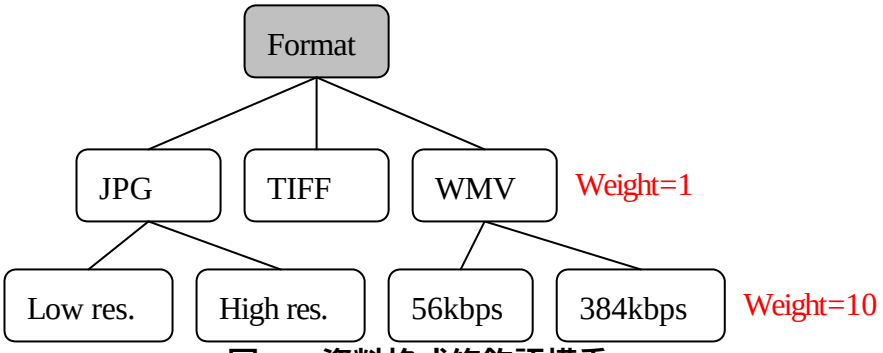


圖 16: 資料格式修飾語權重(a=10)

a 必須適當地給定大小，給定太小會產生困擾－含有 a 個權重為 1 的授權和含有一個權重為 a 的授權將具有相同的優先權。所以在給定 a 的大小時，必須考量每一欄位的修飾語可能個數，給予一個大於個數的值，本論文所有的舉例都以 a = 10 來討論。

衝突解決策略中的四點原則可利用較具權威性的授權來作定義，所謂較具權

威性指的是該授權擁有較高的優先權，利用 CBAM 中的定義 2.6 和 2.8，及修改定義 2.7 並重新定義較具權威性的實體描述為定義 3.5，定義 3.6 則沿用定義 2.8 中對較具權威性授權的定義。

> **定義 3.5 (較具權威性的實體描述, Stronger Entity Specification)。** 假設 dlo 為數位物件, A_1 和 A_2 為兩個授權，且 dlo 為 A_1 和 A_2 的 denoted object，若符合下述條件，則說 ent-spec(A_1)比 ent-spec(A_2)具有較具權威性的物件描述，以 ent-spec(A_1) $>_{dlo}$ ent-spec(A_2)來表示：
>
> 1. ent-spec(A_1) $\in 2^{OI}$ 且 ent-spec(A_2)為詮釋資料表示式。
> 2. ent-spec(A_1)和 ent-spec(A_2)皆為詮釋資料表示式，兩者為單一階層的衝突，且 ent-spec(A_1)比 ent-spec(A_2)具有較特定的修飾語描述，即具有 $mq_{A_1}\ \pi_{MQ}\ mq_{A_2}$ 的關係。
> 3. ent-spec(A_1)和 ent-spec(A_2)皆為詮釋資料表示式，兩者為跨階層的衝突，且 ent-spec(A_1)比 ent-spec(A_2)具有更特定的詮釋資料表示式。

若有如下兩個授權，根據定義 3.5 中的第 3 點，則我們可以說授權 8 具有比授權 9 更具權威性的實體描述，因為授權 8 中詮釋資料修飾語權重為 11，而授權 9 的修飾語權重只有 1 而已。

8: (School=' NCTU' ∧ Occupation=' Graduate'，

Medium=' JPG' ∧ Resolution=' HIGH'，View，+)

9: (School=' NCTU' ∧ Occupation=' Graduate'，Creator='蘇森塘'，View，+)

> **定義 3.6 (較具權威性的授權, Stronger Authorization)**[1]。假設 u 為使用者, dlo 為數位物件，A_1 和 A_2 為兩個授權，且 u 為 A_1 和 A_2 的 denoted user, dlo 為 A_1 和 A_2 的 denoted object，若符合下述條件，則說 A_1 比 A_2 具有較具權威性的授權，以 $A_1 >_{u,dlo} A_2$ 來表示：
>
> 1. crd-spec(A_1) $>_u$ crd-spec(A_2)。
> 2. 無法用第 1 項比較，但存在 ent-spec(A_1) $>_{dlo}$ ent-spec(A_2)的關係。
> 3. 無法用第 1 和第 2 項比較，但存在 priv(A_1) π_P priv(A_2)的關係。
> 4. 無法用第 1、第 2 和第 3 項比較，但存在 sign(A_1) = "-", sign(A_2) = "+" 的關係。

以授權 8 和授權 9 為例，因為兩者無法用定義 3.6 的第 1 點作判斷，而授權 8 含有較具權威性的物件描述，所以根據定義 3.6 第 2 點，我們說授權 8 為較具權威性的授權。

第四章　系統架構及實作

本章將闡述本論文所提以詮釋資料為基礎之存取控制的系統架構及實作系統。第一節敘述此系統架構及相關演算法，第二節說明實作系統。

第一節　系統架構

存取控制系統接受使用者的存取要求並依適當的程序使資訊僅被經由授權的使用者使用，當一個使用者對存取控制系統提出存取要求時，若沒有適合的授權可採用，則依保守的原則拒絕該存取要求，否則便依據相關授權所授予的權力讓使用者對物件適當地進行操作。系統中主要有兩大模組：

1. **存取控制**(Access Control Module)：存取控制模組利用授權資料庫(Authorization Base)來決定是否允許使用者存取物件，若允許則透過數位物件貯藏庫(Dlo Repository)及詮釋資料庫(Dlo Metadata)提供數位物件及詮釋資料。授權資料庫用來存放系統管理者所訂定的授權；數位物件貯藏庫及詮釋資料庫則分別存放數位物件及詮釋資料，兩者皆為數位圖書館的構成元件。
2. **衝突解決**(Conflict Resolver)：若存取控制模組無法判斷是否應給予使用者存取權，也就是說，存在授權衝突的情形時，衝突解決模組依據 3.2.2 所述策略及使用者身份類別階層、詮釋資料階層及權力階層計算出較具權威性的授權。

存取控制系統架構圖如圖 17 所示：

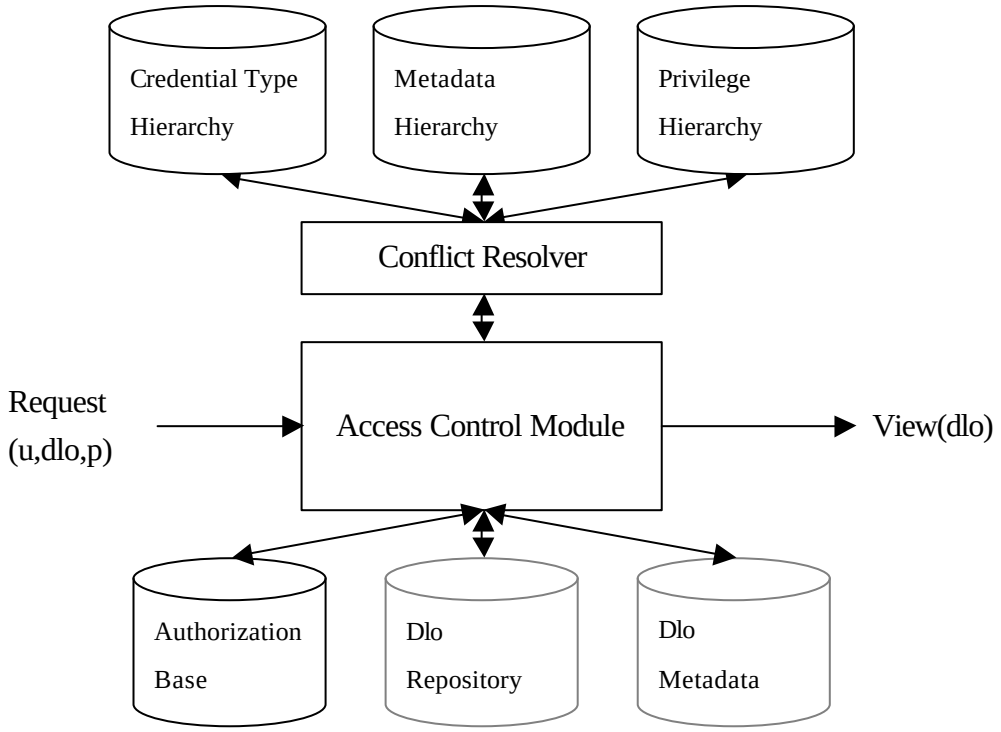


圖 17: MBAC **系統架構圖**

為了確認授權資料庫中有哪些授權可以套用在某一個存取要求，我們利用在[1]**中所提出的授權資料庫投影**(Projection)**來找出所有符合該存取要求中提及之人、物及權力的授權，定義如下：**

> **定義 4.1 (Authorization Base Projection).**[1] Let (u,dlo,p) be an access request and let AB be an authorization base. The projection of AB with regard to (u,dlo,p), denoted $?_{(u,dlo,p)}(AB)$, is the subset of AB, such that, $\forall A \in AB$, $A \in ?_{(u,dlo,p)}(AB)$ iff $u \in U_A$, $i(dlo) \in O_A$, and $priv(A) = p$ or $p \pi_P priv(A)$.

若經投影產生的授權數為零個，代表沒有任何授權描述使用者是否可以對該物件行使特定的權力，則依據保守的原則拒絕該存取要求；若為一個或多個相同符號的授權(同為正向(或負向)授權)，則根據該授權所述來操作；而若有多個授權相衝突時(同時具有正向和負向授權)，則依據定義 3.6 **來找出較具權威性的授權。上述過程其演算法如演算法** 4.1 **所述，其中步驟** 2**、**3 **和** 4 **分別針對使用者身份、詮釋資料和權力來計算較具權威性的授權，步驟** 7 **為投影授權數為零個的情形。**

演算法 4.1 (存取控制)。假設存取要求為(u, dlo, p)**，授權資料庫** AB**，**$?_{(u,dlo,p)}(AB)$**為授權資料庫投影。存取控制流程如下：**

1. **計算授權資料庫投影中授權的個數：**
   - ✓ 0: **跳至步驟** 7
   - ✓ 1: **跳至步驟** 6
   - ✓ 1+: **若有衝突，跳至步驟** 2**；若無，跳至步驟** 6
2. **刪除** $?_{(u,dlo,p)}(AB)$**中較不具權威性的授權，令** $?'_{(u,dlo,p)}(AB) = \{A \mid A \in ?_{(u,dlo,p)}(AB)$ **且** $\nexists A' \in ?_{(u,dlo,p)}(AB)$ such that crd-spec(A) $>_u$ crd-spec(A')$\}$**，並計算投影授權個數：**
   - ✓ 1: **跳至步驟** 6
   - ✓ 1+:**若有衝突，跳至步驟** 3**；若無，跳至步驟** 6
3. **刪除** $?'_{(u,dlo,p)}(AB)$**中較不具權威性的授權，令** $?''_{(u,dlo,p)}(AB) = \{A \mid A \in ?'_{(u,dlo,p)}(AB)$ **且** $\nexists A' \in ?'_{(u,dlo,p)}(AB)$ such that ent-spec(A) $>_{dlo}$ ent-spec(A')$\}$**，並計算投影授權個數：**
   - ✓ 1: **跳至步驟** 6
   - ✓ 1+: **若有衝突，跳至步驟** 4**；若無，跳至步驟** 6
4. **刪除** $?''_{(u,dlo,p)}(AB)$**中較不具權威性的授權，令** $?'''_{(u,dlo,p)}(AB) = \{A \mid A \in ?''_{(u,dlo,p)}(AB)$ **且** $\nexists A' \in ?''_{(u,dlo,p)}(AB)$ such that priv(A)π_P priv(A')$\}$**，並計算投影授權個數：**
   - ✓ 1: **跳至步驟** 6
   - ✓ 1+:**若有衝突，跳至步驟** 5**；若無，跳至步驟** 6
5. **採用負向授權** #
6. **若** sign(A)="+"**，則採用正向授權；若** sign(A)="-"**，則採用負向授權**#
7. **拒絕存取**#

在對授權資料庫作投影時，存有潛在的效率問題[6]**，必須花費大量時間計算每個授權是否可套用在使用者的存取要求中。所以** Ferrari **等人提出預先計算**(Precomputation)**來解決這個問題，存取控制系統事先計算每個授權所套用的人及物，即計算定義** 2.4 **及定義** 3.3 **的集合，並將之儲存在授權資料庫中，當有一存取要求進入系統而要作投影時，可以輕易找出授權資料庫中所有含有該特定人及物的授權，再比對是否具有** priv(A)=p **或** p π_P priv(A)(p **為存取要求中使用者所欲操作的權力，**priv(A)**為授權中所規範的權力**)**的關係即可完成對授權資料庫的投影。預先計算策略的架構圖如圖** 18 **所示：**

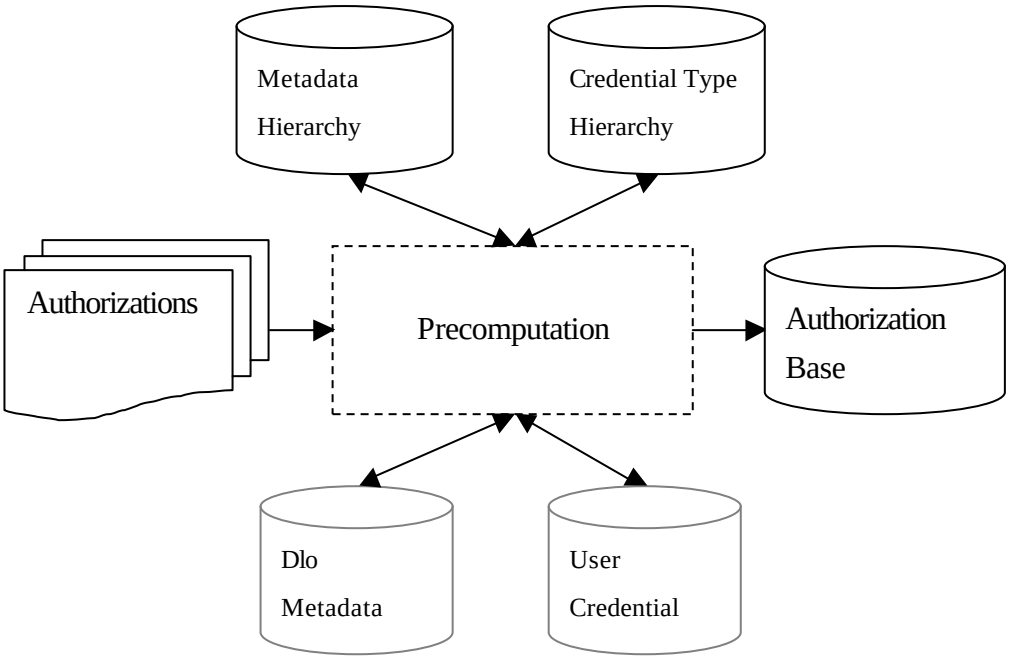


圖 18: **預先計算策略架構圖**

在預先計算授權所套用之人的部分，我們參考[6]**所提出的演算法** 4.2**；在計算授權所套用之物的部分，配合** MBAC **中詮釋資料的特性，我們修改演算法** 4.2 **為演算法** 4.3 **來計算授權所套用之物件。**

演算法 4.2 (預先計算 I)[6]
#**計算** denoted users
If crd-spec(A) is a set of subject ids: U_A = crd-spec(A)
If crd-spec(A) is a credential expression:
If sign(A) = +: U_A = *Denotes*(crd-spec(A))
else U_A = *Denotes*(crd-spec(A)) ∪ *Undef*(crd-spec(A))
endif

演算法 4.3 (預先計算 II)
#**計算** denoted objects
If ent-spec(A) is a set of subject ids: S_A = ent-spec(A)
If ent-spec(A) is a metadata expression:
If sign(A) = +: S_A = *Denotes*(ent-spec(A))
else S_A = *Denotes*(ent-spec(A)) ∪ *Undef*(ent-spec(A))
endif

第二節　實作系統

為驗證本論文提出之以詮釋資料為基礎之存取控制機制，本論文實作一套含有以詮釋資料為基礎的存取控制機制之數位圖書館(以下稱為本系統)，使用的方法如同瀏覽一般的網站，但是當點選到特定物件時，系統便會根據使用者登入的身份和點選的物件決定是否給予存取權力，或者提供適當的錯誤訊息。開發本系統之硬體配備為 Intel Pentium III 1.13Ghz 、系統記憶體 512MB，作業系統為 Microsoft Windows 2000 Server，使用 Microsoft .NET Framework 1.1 作為軟體開發工具。為便於說明起見，在此建立了一個包含 9 個使用者(如表 6)、23 個數位物件(如表 7)(包含蘇森塘和楊英風的作品)及 9 個授權(如表 8)的虛擬使用環境，這 9 個授權所描述的為一個具備存取控制機制的數位圖書館，其中，所有的交大學生可以觀賞低解析度的圖檔和影片(1,4，參照表 8 之 ID)，而交大教授和資科系的學生不受此限(2,3,5,6)，但交大外文系的使用者則都不能觀看影片(8)，而蘇森塘的作品則都可以給交大的使用者使用(7)，另有一條規定明確指出交大學生不能觀看高解析度的影片(9)。此外，我們尚利用使用者 aloha 和物件 M002001s 作為 *Undef* 的測試，所以有 *Undef* (school='NCTU') = {aloha}和 *Undef* (medium='WMV') = {M002001s}的關係。

使用者 ID	學校	系所	職業	年齡
aloha		CIS	Graduate	25
nctu1	NCTU	CSIE	Undergraduate	20
nctu2	NCTU	FL	Undergraduate	22
nctu3	NCTU	CIS	Professor	40
nctu4	NCTU	FL	Graduate	21
nthu1	NTHU	CS	Graduate	24
nthu2	NTHU	PME	Undergraduate	20
nthu3	NTHU	EE	Professor	45
ntu1	NTU	EE	Undergraduate	21

表 6: 實作系統使用者範例

物件 ID	物件名稱	作者	作詞	作曲	編曲	媒體類型	解析度
SP002005s	天烏烏		林福裕	福佬民謠	蘇森塘	JPG	low
SP002005	天烏烏		林福裕	福佬民謠	蘇森塘	JPG	high
SP003001	新竹中學校歌		辛志平	蘇森塘	蘇森塘	JPG	high
TMP0092	緣慧潤生	楊英風				JPG	high
M002001	謊歌		佚名	蘇森塘	蘇森塘	WMV	384kbps
M002001s	謊歌		佚名	蘇森塘	蘇森塘		
TMPV001	繪畫及銅雕系列創作精選	楊英風				WMV	384kbps
TMPV001s	繪畫及銅雕系列創作精選	楊英風				WMV	56kbps

表 7: 實作系統物件範例(僅含部分物件)

ID	使用者身份	物件詮釋資料	權力	正/負向
1	school='NCTU'	medium='WMV'^bitrate='56kbps'	View	Positive
2	school='NCTU'^occupation='Professor'	medium='WMV'	View	Positive
3	school='NCTU'^department='CIS'	medium='WMV'	View	Positive
4	school='NCTU'	medium='JPG'^resolution='low'	View	Positive
5	school='NCTU'^occupation='Professor'	medium='JPG'	View	Positive
6	school='NCTU'^department='CIS'	medium='JPG'	View	Positive
7	school='NCTU'	creator='蘇森塘'	View	Positive
8	school='NCTU'^department='FL'	medium='WMV'	View	Negative
9	school='NCTU'	medium='WMV'^bitrate='384kbps'	View	Negative

表 8: 實作系統授權範例

關於使用者身份的識別，常用的方法如利用帳號管理或是從上站來源來作判斷，本系統採用帳號來管理，每個使用者利用專有的密碼登入本系統，登入畫面如圖 19 所示：

圖 19: **登入畫面**

使用者登入後，即可進入本系統的瀏覽畫面，如圖 20 **所示：**

曲　名	作 詞 者	作 曲 者	編 曲 者
天鳥鳥 樂譜(高解析度)	徐志摩	蘇森墉	
天鳥鳥 樂譜(低解析度)	鄭愁予	蘇森墉	
新竹中學校歌(高解析度)	辛志平	蘇森墉	蘇森墉
謠歌(高解析度)	佚名	蘇森墉	蘇森墉
愁(高解析度)	辛棄疾	蘇森墉	
谷關之夜(高解析度)	蘇森墉	蘇森墉	
大江東去(高解析度)	蘇軾	青主	蘇森墉
天鳥鳥 影音(384kbps)	徐志摩	蘇森墉	
天鳥鳥 影音(56kbps)	徐志摩	蘇森墉	
謠歌(384kbps)	佚名	蘇森墉	蘇森墉
謠歌(56kbps)	佚名	蘇森墉	蘇森墉
谷關之夜(384kbps)	蘇森墉	蘇森墉	
谷關之夜(56kbps)	蘇森墉	蘇森墉	

圖 20: **數位圖書館瀏覽畫面**

點選物件將會產生存取要求，經由系統送入存取控制模組，若所投影出的授權數為 0**，也就是發生演算法** 4.1 **步驟** 7 **的情形時，根據保守的原則，系統拒絕該使**

用者的存取要求，並顯示適當的錯誤訊息。例如使用者 ntu1 點選了任一物件時，因為授權資料庫中並沒有適用於 ntu1 的授權，系統會拒絕 ntu1 操作該物件。拒絕存取的畫面如圖 21 所示：

圖 21: 拒絕存取畫面 I

而若為演算法 4.1 步驟 5 或步驟 6 的情形，當採用正向授權，則系統會將物件呈現在瀏覽畫面中，例如當使用者 nctu3 點選物件 SP003001 時，依據授權 5 所述，nctu3 可以瀏覽該物件，結果如圖 22 所示；當採用負向授權，則系統會帶出錯誤訊息，告知使用者依據哪條授權而拒絕該存取要求，例如當使用者 nctu2 點選物件 M002001 時，依據授權 8 所述，nctu2 不可以瀏覽該物件，結果如圖 23 所示。

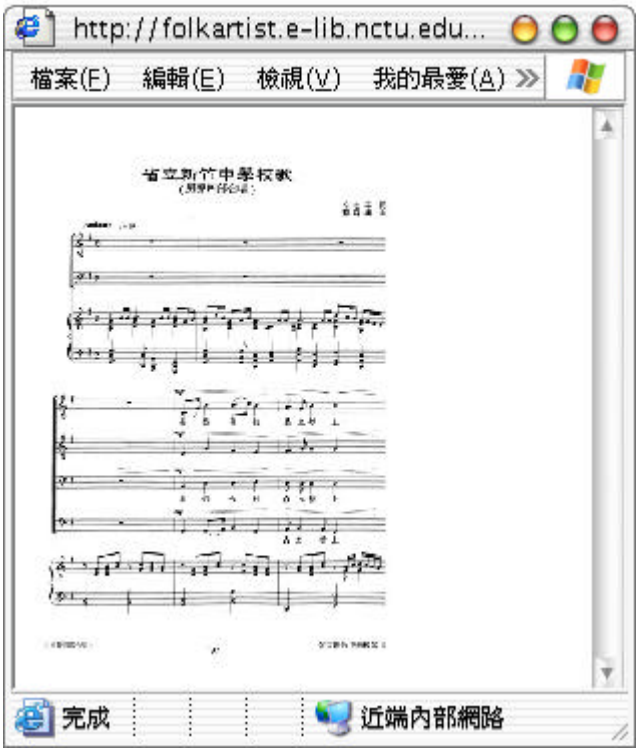


圖 22: **瀏覽結果畫面**

圖 23: **拒絕存取畫面** II

在系統管理介面中，允許系統管理者新增、刪除或修改授權，以新增授權為例，在 Credential specification **欄位輸入所欲控制使用者身份描述或識別** ID**；在** Entity specification **欄位輸入所欲控制物件詮釋資料表示式或識別** ID**；**Privilege **欄位選擇給予的權力；而在** Sign **欄位指定此為正向或負向授權。新增授權畫面如圖** 24 **所示：**

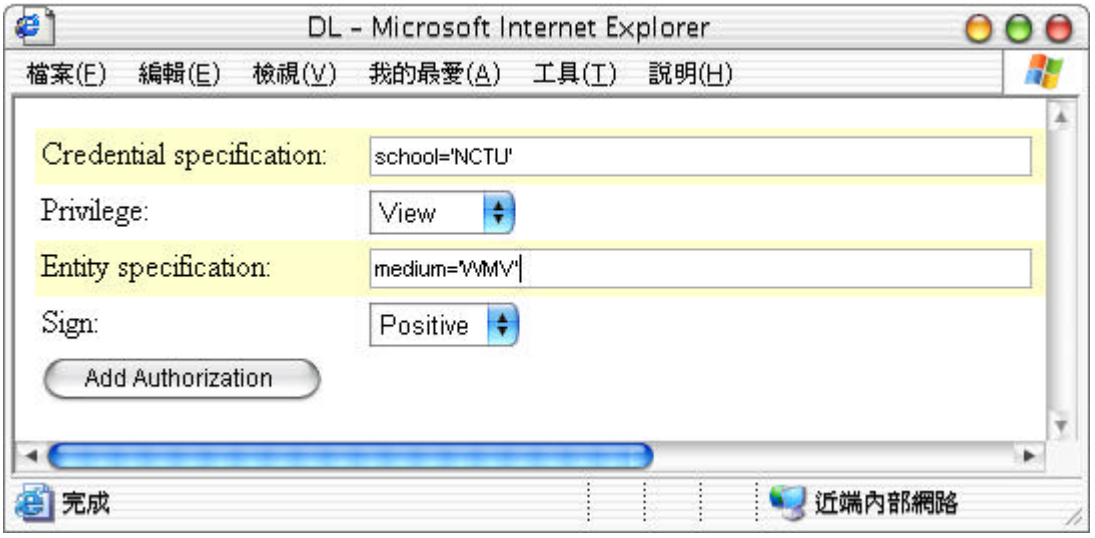


圖 24: **新增授權畫面**

另外在管理介面另一個重要的功能為執行預先計算，以表 6**、表** 7 **和表** 8 **為例，執行預先計算結果如表** 9 **所示，以** ID=1 **之授權為例，套用此授權的使用者有** nctu1、nctu2、nctu3 **和** nctu4 **這四人；物件則有** M002005s、M004002s、TMPV001s **和** TMPV002s。

ID	Denoted Users	Denoted Objects
1	nctu1;nctu2;nctu3; nctu4	M002005s;M004002s;TMPV001s;TMPV002s
2	nctu3	M004002;M002005;M002005s;M002001;M004002s;TMPV001; TMPV001s;TMPV002;TMPV002s
3	nctu3	M004002;M002005;M002005s;M002001;M004002s;TMPV001; TMPV001s;TMPV002;TMPV002s
4	nctu1;nctu2;nctu3; nctu4	SP002005s
5	nctu3	SP002005s;TMP0092;SP002005;SP003001;SP002001;SP003002; SP004002;SP003005;TMP0080;TMP0086;TMP0145;TMP0330;TMP0304
6	nctu3	SP002005s;TMP0092;SP002005;SP003001;SP002001;SP003002; SP004002;SP003005;TMP0080;TMP0086;TMP0145;TMP0330;TMP0304
7	nctu1;nctu2;nctu3; nctu4	M004002;M002005;M002005s;M002001;M002001s;M004002s; SP002005s;SP002005;SP003001;SP002001;SP003002;SP004002;SP00300 5
8	nctu2;nctu4;aloha	M004002;M002005;M002005s;M002001;M004002s;TMPV001; TMPV001s;TMPV002;TMPV002s;M002001s
9	nctu1;nctu2;nctu3; nctu4;aloha	M004002;M002005;M002001;TMPV001;TMPV002;M002001s

表 9: 預先計算結果

第五章　結論與未來研究方向

本章總結本論文以及敘述未來研究方向，第一節說明本論文提出以詮釋資料為基礎之存取控制的優點，以及 MBAC 與 CBAM 的比較，第二節則說明本論文未來可能的研究發展方向。

第一節　結論

本論文針對數位圖書館環境提出以詮釋資料為基礎的存取控制架構，利用含資料屬性之詮釋資料建立存取控制管理策略(Policy)，大幅簡化使用傳統存取控制機制所需之人力物力，幫助數位圖書館快速建立存取控制機制。本論文所提出的方法具有下列特性：

1. 利用開發數位圖書館時不可或缺的詮釋資料來建立存取控制，減少建立過程的額外負擔(和 RBAC、CBAM 相比)，並兼顧實作上的可行性(和 ACM、ACL、Capability 相比)。
2. 因為用來作存取控制的物件多為代理物件，所以其詮釋資料廣泛應用於本論文所提出的方法中。
3. 利用給定權重的方式，解決授權發生衝突時，詮釋資料修飾語階層中多個階層的優先權判斷問題。

和 CBAM 相比，雖然 MBAC 不對物件內容萃取概念，但因所利用的詮釋資料即為描述資料的資料(Data About Data)，所以 MBAC 也提供和物件內容有關的存取控制方式。另外兩者皆可利用物件的識別資料作為實體描述，物件之能否存取和內容沒有關係，所以皆提供和物件內容無關的存取控制方式。而就存取控制的物件之結構性來比較，兩者皆可處理非結構性的物件，但 CBAM 礙於其方法的限制，必須要有文字性的資料內容，所以如圖檔或影片等物件，CBAM 必須透過詮釋資料來描述物件的內容。

雖然本論文所提出的方法建立在 CBAM 的基礎上，並具有 CBAM 所欠缺的

一些優點，但是在整個存取控制的需求上，物件內容的概念也是不可或缺的存取控制依據，所以本論文提出 MBAC 之目的並不在於取代 CBAM，而是提供數位圖書館開發者在考量存取控制機制時的另一種選擇。

第二節　未來研究方向

MBAC 和 CBAM 分別針對不同的需求而設計，都有其存在的必要性，一個整合兩者優點的模型將是未來的研究目標。透過詮釋資料中的簡述(Description)欄位來描述物件的內容摘要，整合 CBAM 的概念萃取模組，接下來的研究考量即為如何整合修飾語及概念兩種不同類型的階層，在互相衝突時如何判斷其優先順序？

另外，如何在 MBAC 中整合加密、數位簽章、隱私權、版權和電子商務...等，從而建立完整的數位版權管理架構，在數位版權管理逐漸受到重視的今天，亦是一個未來的研究方向。

參考文獻

[1] N.R. Adam, V. Atluri, E. Bertino, and E. Ferrari, "A Content-Based Authorization Model for Digital Libraries," *IEEE Trans. Knowledge and Data Eng.*, vol. 14, no. 2, 2002.

[2] W.Y. Arms, C. Blanchi, and E.A. Overly, "An Architecture for Information in Digital Libraries," *D-Lib Magazine*, February 1997.

[3] E. Bertino, P. Samarati, and S. Jajodia, "An Extended Authorization Model," *IEEE Trans. Knowledge and Data Eng.*, vol. 9, no. 1, pp. 85-101, 1997.

[4] R.S. Fabry, *Capability-based addressing*, *Communications of the ACM*, vol. 17, no. 7, pp. 403-412, July 1974.

[5] D. Ferraiolo and R. Kuhn, "Role-Based Access Control", 15th National Computer Security Conference, pp. 554-563, 1992.

[6] E. Ferrari, N.R. Adam, V. Atluri, E. Bertino, and U. Capuozzo, "An Authorization System for Digital Libraries," The VLDB Journal, Volume 11, issue 1, 2002.

[7] A.J. Gilliland-Swetland, "Setting the Stage: Defining Metadata" in *Introduction to Metadata: Pathways to Digital Information*, Murtha Baca, ed. Los Angeles: Getty Information Institute, 1998, http://www.getty.edu/research/institute/standards/intrometadata/2_articles/index.html

[8] E. Gudes, H. Song, and E.B. Fernandez, "Evaluation of Negative, Predicate, and Instance-Based Authorization in Object-Oriented Databases," *Database Security, IV: Status and Prospects*, 1991.

[9] R. Iannella, "Digital Rights Management (DRM) Architectures," *D-Lib Magazine*, June 2001.

[10] R. Kahn and R. Wilensky, "A Framework for Distributed Digital Object Services," May 1995, http://www.cnri.reston.va.us/home/cstr/arch/k-w.html

[11] H.R. Ke, "**數位圖書館概論**", http://www.cc.nctu.edu.tw/~claven/course/metadata89/dl_overview.pdf

[12] B.W. Lampson, *Protection. Operating Systems Review*, vol. 8, no. 1, pp. 18-34, January 1974.

[13] M.J. Moyer and M. Ahamad, "Generalized Role-Based Access Control", *21st Int'l Conf. Distributed Computing Systems*, pp. 391-398, 2001.

[14] 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.

[15] R.S. Sandhu and P. Samarati, "Access Control: Principles and Practice," *IEEE Communications Magazine*, vol. 32, no. 9, pp. 40-48, 1994.

[16] Dublin Core Metadata Element Set Resource Page, http://dublincore.org/

[17] The Digital Object Identifier System, http://www.doi.org/

[18] CNRI, Handle System, http://www.handle.net

[19] **資源組織與檢索之規範**, http://ross.lis.ntu.edu.tw/