

詮釋資料與數位圖書館系統互通性之探討
The Study of Interoperability of Digital Libraries with Metadata

柯皓仁 **Hao-ren Ke**

交通大學圖書館副教授兼數位圖書資訊組組長
Associate Professor, Head of Digital Library and Information Division,
National Chiao Tung University Library
E-mail: claven@lib.nctu.edu.tw

黃夙賢 **Su-shang Huang**

交通大學資訊科學研究所博士生
Ph. D Student, Institute of Computer and Information Science,
National Chiao Tung University
E-mail: sshuang@cis.nctu.edu.tw

楊維邦 **Wei-pang Yang**

交通大學圖書館館長
Professor and Director of Library,
National Chiao Tung University
E-mail: wpyang@cis.nctu.edu.tw

【摘要】

詮釋資料通常用來描述數位圖書館藏品資源的特性，本文則是取詮釋資料最廣義的解釋—Data About Data，並將詮釋資料應用在數位圖書館系統間的互通。本文首先針對數位圖書館互通性下一定義，並介紹數位圖書館互通性的相關議題和方案。其次則是以 Stanford 數位圖書館計畫為例，說明詮釋資料在數位圖書館系統互通性上能夠扮演的角色。文中並闡述交通大學圖書館為解決數位圖書館互通性問題所提出的詮釋資料建構語言，並以交通大學圖書館虛擬聯合目錄系統來驗證詮釋資料建構語言的可行性。

【Abstract】

The term "Metadata" is often used by information professions to refer to character the resources contained in a digital library. This paper adopts the broadest meaning of "Metadata"-

-Data about Data, and applies metadata to facilitate the inter-operation among digital library systems. It first defines digital library interoperability, its related issues and approaches. Then, citing as an instance the Stanford Digital Library Project to describe the roles that metadata can play in handling digital library interoperability. This paper also describes how National Chiao Tung University has tried to solve the problem of handling digital library interoperability by proposing the *Metadata Modeling Language* (MML), which is a language to define the metadata of digital-library contents and services, and define the translation of two kinds of metadata. Finally, a virtual union catalog system based on MML, called *VUCS@NCTU*, is implemented to vindicate the feasibility of MML.

關 鍵 詞：數位圖書館；詮釋資料；數位圖書館互通性；詮釋資料建構語言
Keywords：Digital library; Metadata; Digital library interoperability; Metadata modeling language

壹、緒論

一、數位圖書館互通性概論

近年來由於電腦和網路科技的蓬勃發展，數位圖書館已成為圖書館界致力追尋的目標之一。然而，究竟何謂數位圖書館呢？數位圖書館和傳統圖書館又有何區別呢？文獻上關於數位圖書館的定義很多（註1）（註2）（註3）（註4）（註5），筆者認為 American Digital Library Federation 在 1998 年提出的數位圖書館定義（註6）（註7）與圖書館的使命和功能最為貼近：

數位圖書館乃是擁有相關資源(包含軟硬體設備、網路、專業人士等)以執行下列任務的機構：對數位形式的館藏進行挑選、組織、提供使用、解譯、傳播、保持完整性、長期保存

等工作，並使這些數位形式的館藏能為特定讀者群快速且經濟地運用。（註8）

由此可知，其實數位圖書館與傳統圖書館具有相同的目標，只是較為偏重數位格式的館藏，但若把數位圖書館的定義侷限在「數位化的館藏」，乃是忽視了圖書館從業人員在技術服務和讀者服務上的努力。

不可否認的，電腦和網路科技在數位圖書館的發展上扮演了極為重要的角色。若從電腦和網路科技專家的角度來看，數位圖書館則是：

- 一種分散式的技術環境，可大幅降低個人或機構在資訊的創造、散佈、操縱、儲存、整合和再利用上的障礙。（註9）
- 一種讓特定讀者群藉由一致的方法來使用龐大且具有組織的資訊和知識寶庫的系統。（註10）

- 一種整合了計算、儲存和通訊等方面的軟硬體服務環境，用以重現、模擬和延伸傳統圖書館所提供的以紙本或其他方法的各種服務。（註 11）
- 一群獨立發展的元件（Component），這些元件必須彼此溝通、配合，才能達成使用者的需求。（註 12）

Cleveland（註 13）在「Digital Libraries: Definitions, Issues, and Challenges」一文中說明數位圖書館在技術架構上所須面對的挑戰時，曾提及：

在真實情況下，數位圖書館絕非是一套單一完整的數位系統，而是一群異質系統或資源透過網路連接，並整合在單一的界面下。這些異質系統或資源可能包含指向紙本或數位媒體的書目資料庫、索引與資料找尋工具、目錄、相片、電子期刊、數據資料庫…等。儘管這些資源或許存在於不同的系統或資料庫中，但對讀者而言，它們卻有如是存在於單一系統內。（註 14）

Cleveland 的這段話點出了數位圖書館的一個重要目標：如何把散佈在各地、獨立運作與維護的數位圖書館系統和資源（Resource）整合在一起，讓讀者毋需了解所要取得的資源到底實際上是分散各地還是整合在一個系統下，亦即數位圖書館資源和服務的所在地對讀者而言是不重要的、透明的（Transparency），讀者僅需透過單一的系統或界面即可取得需要的所有資訊。要實踐前述目標，首先必須達成數位

圖書館系統間的互通（Interoperability），也是本文所要深入探討的主題。值得注意的是，當探討數位圖書館系統間的互通時，我們事實上是把數位圖書館視為是一群獨立發展的元件，這些元件必須彼此溝通、配合，才能達成使用者的需求。

廣義而言，互通性並非是數位圖書館環境下獨有的議題，而是在傳統圖書館時代就存在了，以下用一個簡單的情節（Scenario）來點出「互通性」的意涵：當讀者所要的資料為其所屬圖書館的館藏時，讀者可直接取得利用；但若資料並非其所屬館的館藏時，讀者可向參考諮詢組（以交通大學圖書館為例）提出需求，由參考諮詢組在了解讀者需求後向他館申請館際合作或是向文獻傳遞服務機構訂購，在這種情形下，參考諮詢組扮演了讀者（或是讀者所屬館）和其他館或文獻傳遞服務機構間互通的橋樑，藉由這個橋樑的協助，在理想狀況下，讀者毋需了解其所需資料的所在地即可取得資料。現在，讓我們把每一個圖書館或文獻傳遞服務機構都視為是獨立的數位圖書館系統，則先前描述的這段情節就如同多個數位圖書館系統間的互通，參考服務組就好比使用者所面對的單一界面，透過這個單一界面便可取得“所有”（在理想狀況下）數位圖書館系統的資訊。

二、數位圖書館互通性相關議題

數位圖書館系統間要達成互通所

必須面對的議題包含檢索的表述 (Query Formulation)、資源的發現 (Resource Discovery)、檢索的轉譯 (Query Translation)、結果的整合分析 (Result Analysis) 等 (註 15)，在 Table 1 中以傳統圖書館和數位圖書館對照的方式來敘述這些議題。

當然，數位圖書館系統間互通性的相關議題很多，除了 Table 1 所列出的議題之外，尚有使用者認證 (User Authentication)、付款機制 (Payment Mechanism)、智財權管理 (Right Management) 等。

三、數位圖書館互通性方案

長久以來，異質系統 (Heterogeneous System) 間如何達成互通一直是資訊科技專家致力研究的課題之一，而數位圖書館系統可說是異質系統的一種，因此異質系統間互通性的研究成果應能作為在探討數位圖書館系統間如何達成互通時的借鏡。在進一步探討異質系統、數位圖書館系統的互通性方案之前，我們先談談要如何評估互通性方案的成功與否呢？

事實上，隨著應用領域 (Application

Table 1 數位圖書館系統間互通性的相關議題

互通性相關議題	傳統圖書館環境下	數位圖書館環境下
檢索的表述	由讀者本人或參考諮詢組館員將讀者所要找尋資料的特性明確地表述出來	由讀者將其所要找尋資料的特性，表述成其所面對的單一系統或界面之檢索功能
資源的發現	根據檢索的表述，參考資源組館員必須有能力知道讀者要的資料存在於哪些圖書館或文獻傳遞服務機構	根據檢索的表述，單一系統或界面必須有能力知道讀者要的資料存在於哪些數位圖書館系統
檢索的轉譯	參考諮詢館員必須將先前得到的檢索表述，轉譯成要去索取資料的圖書館或文獻傳遞服務機構所能夠瞭解的辭彙，這個過程可能包括使用語言、專有名詞…等的轉譯	單一系統或界面必須將先前得到的檢索表述，轉譯成要去索取資料的數位圖書館系統所能夠瞭解的辭彙，這個過程可能包含檢索功能、詮釋資料…等的轉譯
結果的整合分析	參考諮詢館員從去索取資料的圖書館或文獻傳遞服務機構處找到資料後，必須經過整合、分析等工作，再將結果交給讀者	單一系統或界面從去索取資料的數位圖書館系統處找到資料後，必須將資料整合、分析後再交給讀者

Domain)、設計目標 (Design Goal)、設計前提 (Design Assumption) 的不同, 要去評估一個互通性方案是否成功、或是比較兩個互通性方案的優劣是非常困難的。儘管如此, 基本上我們還是可以用下列六項準則來評估互通性方案 (註 16) :

1. 元件自主性 (Component Autonomy) ; 儘量不需要遵循共同的規範來設計數位圖書館系統或元件¹。
2. 建置基礎建設的代價 (Cost of Infrastructure) ; 儘量不需花費龐大的成本 (含軟硬體設備、人力…等成本) 來建置支援元件互通的基礎建設。
3. 元件的加入 (Contributing Components) ; 要很容易地讓新的元件加入互通的行列。
4. 元件的使用 (Using Components) ; 元件的使用愈簡單愈好。
5. 元件功能的複雜度和廣泛度 (Task Breadth and Complexity) ; 元件所能達成的功能儘量多樣化, 並且愈複雜愈好。
6. 元件數目的成長空間 (Scalability in the Number of Components) ; 所能支援參與互通的元件數量愈多愈好。

在真實情況下, 前述六項準則並非互相獨立, 甚至是互相衝突的。因此當要在多個互通性方案間做選擇時, 必須先確認這六個準則的優先順序, 再根據這六個準則來評估各個互

通性方案。

過去幾年來, 學者專家已經提出許多異質系統間互通的可行方案, 而這些方案間也有彼此整合的趨勢, 以下茲概要描述這些方案 (註 17) :

1. 單一共同規範 (Strong Standard) ;

異質系統間互通的最傳統方案是要求元件採用共同規範。TCP/IP、Z39.50、ISO 802 等皆是共同規範的例子。此一方案的優點在於系統發展廠商 (Vendor) 甚至一般的程式設計師都能夠明確地依循規範來發展互通元件, 也很容易地便能以符合規範的模組為基礎向上發展新服務; 缺點則在於共同規範的制定往往曠日費時, 此外亦違反元件獨立自主的原則。

2. 規範群 (Family of Standards) ;

顧名思義, 此種方案乃是讓元件設計者擁有自由度從多個共同規範中去選用適合的規範, 例如電子商務系統能選用 SET 或 SSL 等一或多種付款的安全機制。此種方案減輕了單一共同規範的缺點但仍維持它的優點, 然而當一個元件要與其他元件互通時, 必須先以人工或自動的方法獲知其他元件採用的規範。

3. 外部協調機制 (External Mediation) ;

若要維持元件全然的獨立自主, 最好是將互通的機制移到元件之外, 亦即元件在設計時毋需考慮到互通的問題。外部協調機制的主要功能在於

1 本文在討論數位圖書館的互通時是將數位圖書館系統視為一群獨立發展的元件, 是以本文中「數位圖書館系統」和「數位圖書館元件」這兩個詞的意義是相同的。

轉譯不同系統元件所採用的資料格式（如詮釋資料）以及交談模式（Interaction Modes），前者如 Dublin Core 和 USMARC 間的轉換，後者如 HTTP（Stateless）和 Z39.50（Session-based）間的互通。這種方案的優點在於元件能保持自主性、可互通的元件在數量和種類上有較大的成長空間、元件的使用較為容易；缺點則是要為每一種加入互通的元件設計其外部協調機制（換句話說，如果有M種元件要和其他N種元件互通，需要設計出M*N種外部協調機制）。Stanford 數位圖書館計畫（註 18）和本文第參節提出的詮釋資料建構語言（Metadata Modeling Language, MML）就是採用此種互通的方案。

4. 以規格描述為基礎的元件互動（Specification-based Interaction）；

此種方案係指藉由完整地描述所有資料（Data）、作業（Operation）的語意（Semantics）、結構（Structure）來達成元件間的互通。此一方案毋需藉由事先安排（如規範群方法）或外部調解的機制即能達成元件互通。在做法上必須有一套機制或語言，用宣告性的形式（Declarative Form）完整描述每一元件的設計前提、需求、界面、提供的服務等，讓每一個元件經由檢視（Inspect）與推論（Reason）其他元件的規格後達成互通。採用此一方案的所有元件必須使用一致的知識描述、推理、檢索、交換機制，如 Agent Communication Language (ACL)。

（註 19）此種方案的優點在於元件獨立自主的程度很高，且不需要外部協調機制即能確保元件數目和種類的成長；而其缺點在於要完整地描述元件的規格並非易事，在某些應用領域下甚至是不可能的任務。Michigan 數位圖書館計畫就是採用此種互通的方案。（註 20）

在大略探討數位圖書館系統互通性的相關議題和可行方案之後，相信大家已經領會到何謂數位圖書館系統互通性了。接下來，筆者在第二節將敘述詮釋資料和數位圖書館互通性之間的關連，並概述 Stanford 數位圖書館計畫（簡稱 Stanford DL）如何利用詮釋資料來達成數位圖書館系統的互通；然後，筆者在第三節將詳細闡述交通大學圖書館的數位圖書館計畫所提出的詮釋資料架構，以及此一架構如何實踐數位圖書館系統的互通。

貳、詮釋資料與數位圖書館系統互通性

一、詮釋資料簡介

在資訊爆炸的網路環境裡，如何快速且正確地找尋資料乃是一重要的研究課題，這也是詮釋資料（Metadata）之概念出現的主要原因。一般所熟悉的詮釋資料，是指在經由資訊組織與整理工作後，所定出的對藏品資源屬性、內容、特色的描述，目的在促進資訊系統中對資源的檢索、管理、分析。（註 21）（註 22）（註 23）（註 24）

(註 25) 詮釋資料的種類很多，諸如機讀編目格式 (MARC)、目錄交換格式 (DIF)、政府資訊指引服務 (GILS)、都柏林核心集 (Dublin Core) 等都是應用在不同領域的詮釋資料。

如果我們從詮釋資料最廣義的定義—Data about Data—來看，其實詮釋資料的作用並不侷限於描述藏品資料的屬性、內容、特色；

1. **Data about Data** 中的第一個 Data 不單只是資料而已，還包含了資訊 (Information)、知識 (Knowledge) 的意涵在內。亦即，詮釋資料可進

一步解釋為“Information about Data”、“Knowledge about Data”。

2. **Data about Data** 中的第二個 Data 不單只是藏品而已，還可衍伸為數位圖書館系統的軟硬體、儲存的資料類型、提供的服務、採用的規約 (Protocol)、收費方式、與其他系統互通的方式等。

Anne J. Gilliland-Swetland 將詮釋資料根據其用途劃分為管理、記述、保存、技術、利用等五種類型 (註 26)，如 Table 2 所示。

此外，Anne J. Gilliland-Swetland 亦

Table 2 詮釋資料的類型 (譯自 Anne J. Gilliland-Swetland)

類型 (Type)	定義 (Definition)	範例 (Examples)
管理 (Administrative)	用於管理藏品資源的詮釋資料	探訪資訊 智財權與重製追蹤 記載合法使用藏品資源之必要條件的文件 藏品資源所在地資訊 數位化的選擇基準 版本控制
記述 (Descriptive)	用於敘述或識別藏品資源的詮釋資料	編目記錄 搜尋輔助工具 特定學門的索引 藏品資源間的超連結關係 使用者附加的註解
保存 (Preservation)	與藏品資源之保存管理相關的詮釋資料	資源實體保存狀況的描述 記述在保存藏品資源之實體或數位化版本時所採取的行動 (例如資料更新或轉移)
技術 (Technical)	與數位圖書館系統之運作、詮釋資料之作用相關的詮釋資料	記載系統軟硬體的檔案 數位化相關資訊，如格式、壓縮比等 系統回應時間的歷程記錄 授權與保密資料，如密碼、加密公鑰私鑰
利用 (Use)	與藏品資源利用的層次、類型相關的詮釋資料	展覽記錄 使用者和其使用的記錄 資源再利用與多重版本的相關資訊

根據詮釋資料的來源、建立方式、專業性、狀態、結構化、語意、層次等七種特質來敘述詮釋資料的特性（註

27），如 Table 3 所示。

爲了讓詮釋資料能夠應用在電腦和網路世界裡，甚至在不同的數位圖

Table 3 詮釋資料的特質（譯自 Anne J. Gilliland-Swetland）

特質 (Attribute)	特性 (Characteristics)	範例 (Examples)
詮釋資料的來源 (Source of Metadata)	在藏品資源初次建立或數位化之時便產生的內部詮釋資料	檔名或檔頭資訊 目錄架構 檔案格式與壓縮技術
	在藏品資源建立或數位化之後才產生的外部詮釋資料	登錄或編目記錄 智財權或其他法律相關資訊
詮釋資料的建立方式 (Method of Metadata Creation)	電腦自動產生	關鍵字索引 使用者交易記錄
	人工建立	記述性詮釋資料，如編目記錄與 Dublin Core
詮釋資料的專業性 (Nature of Metadata)	不需由主題或資訊組織專家建立的詮釋資料	HTML 檔的 <META> 標籤 個人檔案管理系統
	需由主題或資訊組織專家建立的詮釋資料	特定學門的主題編目 MARC
詮釋資料的狀態 (Status)	靜態詮釋資料，一旦建立後便不會更改	藏品資源的題名、起源、建立日期
	動態詮釋資料，在藏品資源利用或處理後可能會更改	目錄架構 使用者交易記錄 影像解析度
	長期詮釋資料—確保數位文件能被取得與利用的資訊	技術格式與處理資訊 智財權宣告資訊
	短期詮釋資料	與保存管理相關的資訊
詮釋資料的結構 (Structure)	符合規範（無論標準與否）的結構性詮釋資料	MARC、TEI、EAD 資料庫的綱要 (Schema)
	不合乎規範的非結構性詮釋資料	不具結構性的備註或使用者附加的註解
詮釋資料的語意 (Semantics)	符合標準化辭彙或權威格式的詮釋資料	AAT ULAN AACR2
	不符合標準化辭彙或權威格式的詮釋資料	自由文字 (Free-text) 註解 HTML 文件的 <META> 標籤
詮釋資料的層次 (Level)	與整體藏品資源 (Collection) 相關的詮釋資料	主題索引
	與個別藏品資源相關的詮釋資料	影像標題與日期資訊 格式資訊

書館系統間交流，必須將詮釋資料表示成電腦能夠瞭解 (Machine-Understandable) 的格式，此時有三個層面必須考慮，即詮釋資料的語意 (Semantics)、結構 (Structure)、和語法 (Syntax)；語意是指詮釋資料的意涵，多人 (或團體) 在運用同一種詮釋資料時必須對詮釋資料的意涵取得一致，才不至於「雞同鴨講」。例如說，若要用 Creator 這個詞來著錄一本書時，必須先對 Creator 究竟是代表書的作者還是書的出版者取得共識；結構則是指詮釋資料的資料模型，即如何把資源的屬性關係定義出來，目前在網路上多使用 RDF (Resource Description Framework) 來定義詮釋資料的結構；語法則是用電腦能夠了解的文法來呈現詮釋資料的結構，目前在網路上多使用

XML (Extensible Markup Language) 做為詮釋資料的語法。

二、詮釋資料與數位圖書館互通性的關聯

在【壹-二】節中，筆者概略說明了數位圖書館系統互通性的相關議題，若是我們想要用詮釋資料的概念來處理這些議題，究竟需要哪些詮釋資料呢？A. Paepcke 等人進行 Stanford DL，曾經陳述在處理檢索的表述、資源的發現、檢索的轉譯、結果的整合分析等議題時所需要用到的詮釋資料 (註 28)，茲彙整於 Table 4。

在考慮前述互通性的相關議題之後，Stanford DL 認為需要下列幾種詮釋資料機制來處理這些議題：

1. 真實世界裡各種詮釋資料 (如

Table 4 處理數位圖書館互通性相關議題所需要的詮釋資料

互通性相關議題	需要用到的詮釋資料
檢索的表述	<ul style="list-style-type: none"> ●讀者面對之單一系統或界面所採用的詮釋資料 ●讀者面對之單一系統或界面的系統功能
資源的發現	<ul style="list-style-type: none"> ●每一個提供藏品資源的數位圖書館系統之內容摘要 (Content Summary)
檢索的轉譯	<ul style="list-style-type: none"> ●每一個提供藏品資源的數位圖書館系統所採用的詮釋資料、可資檢索與擷取的詮釋資料屬性、支援的檢索運算 (Boolean、Truncation、Stemming)、停用字、索引…等 ●把讀者面對之單一系統或界面所採用的詮釋資料轉譯成每一個提供藏品資源的數位圖書館系統所採用的詮釋資料
結果的整合分析	<ul style="list-style-type: none"> ●把每一個提供藏品資源的數位圖書館系統所採用的詮釋資料轉譯成讀者面對之單一系統或界面所採用的詮釋資料

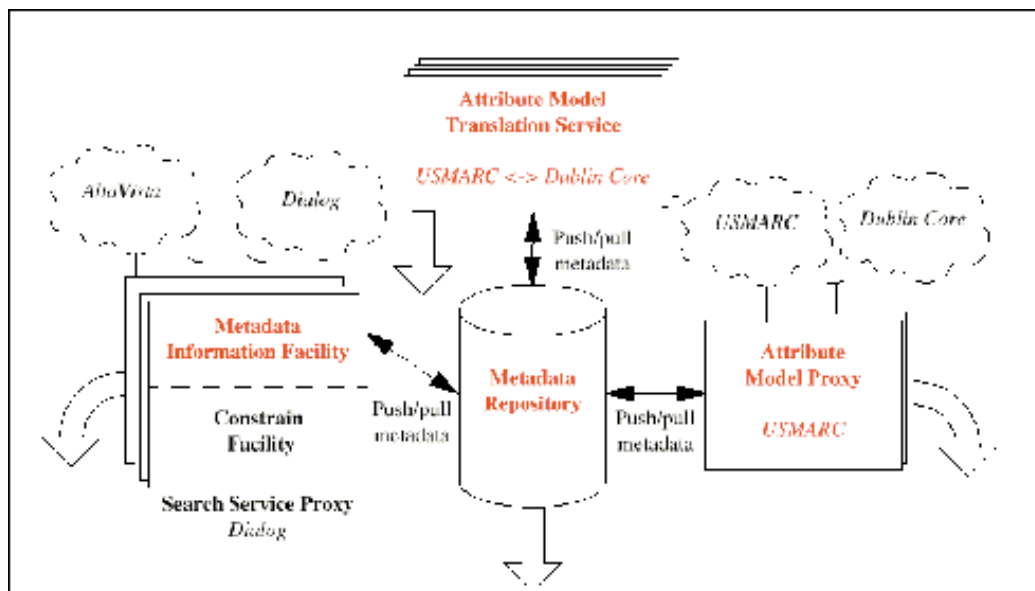


Figure 1 Stanford數位圖書館計畫的詮釋資料架構（資料來源：（註29））

1. 讓各數位圖書館元件能夠輸出（Export）其系統性能（Capability）的機制。
2. 二或多種詮釋資料間轉換的機制。
3. 讓各數位圖書館元件能夠輸出（Export）其系統性能（Capability）的機制。
4. 讓各數位圖書館元件能夠輸出其藏品資源之內容摘要（Content Summary）的機制。

Stanford DL 據此提出了其詮釋資料架構（Metadata Architecture），如 Figure 1 所示，共可分為 *Attribute Model Proxy*（用來實現機制 1）、*Attribute Model Translation Service*（用來實現機制 2）、*Search Service Proxy Metadata Information Facility*（用來實現機制 3 和 4）、*Metadata Repository* 等四部

份。接下來，我們就概略說明這四個部份的作用。

◆ Attribute Model Proxy

Attribute Model Proxy 將真實世界裡的詮釋資料表示成物件（Object），在概念上，我們可以將各種詮釋資料，例如 US-MARC、Dublin Core、Z39.50 Bib1、GILS 等，都用 Attribute Model 的方式來描述。*Attribute Model Proxy* 並提供詮釋資料相關資訊的查詢，例如詮釋資料的說明、屬性的形態（Type）與值域（Value）、屬性間的關連等。

詮釋資料中的每個屬性則是以 *AttributeItem* 物件來表示，包含有下列性質：

- attrModelName: String (所屬詮釋資料的種類)
- attrName: String (詮釋資料屬性的名稱)
- attrValueType: String (詮釋資料屬性的形態)
- attrDocumentation: String (詮釋資料屬性的說明)

當要建構一種包含 Title、Creator、Description 等三個屬性的詮釋資料（姑且稱此詮釋資料為 Simple_Dublin_Core）時，則表示如下：

```
Simple_Dublin_Core:{
inst1:{
  attrModelName -> 'Simple_Dublin_Core'
  attrName -> 'Title'
  attrValueType -> 'String'
  attrDocumentation -> 'This is a title for a work.}'
inst2: {
  attrModelName -> 'Simple_Dublin_Core'
  attrName -> 'Creator'
  attrValueType -> 'SEQUENCE of String'
  attrDocumentation -> 'first-name/last-name'}
inst3: {
  attrModelName -> 'Simple_Dublin_Core'
  attrName -> 'Description'
  attrValueType -> 'String'
  attrDocumentation -> 'Description of this work'}
}
```

◆ Attribute Model Translation Service

顧名思義，Attribute Model Translation Service 的任務在於執行不同詮釋資料間的轉換。詮釋資料間的轉換所涉及的層面很多，除了屬性的對應 (Attribute Mapping) 之外，還需要考

慮到屬性形態 (Type) 的不同，在大部份的情況中，詮釋資料間的轉換是困難且無法百分之百正確的。Stanford DL 是以 Method 的方式來實現詮釋資料轉換機制，如下所示：

- getNameMapping(toAttrModel, fromAttrModel, from AttrName): SEQUENCE of String
(給定原始詮釋資料[fromAttrModel]中的一個屬性名稱[fromAttrName]，傳回此屬性對應到目標詮釋資料[toAttrModel]的屬性名稱)
- getValueMapping(toAttrModel, fromAttrModel, toAttrName, fromAttrName, fromAttrValue): Any
(給定原始詮釋資料中的一個屬性名稱和屬性值[fromAttrValue]，以及此屬性對應到目標詮釋資料的屬性名稱[toAttrName]，傳回對應到目標詮釋資料的屬性值)
- getToAttrModels(): SEQUENCE of string
(傳回 Translation Service 能夠轉換的目標詮釋資料種類)
- getFromAttrModels(): SEQUENCE of string
(傳回 Translation Service 能夠轉換的原始詮釋資料種類)
- getDocumentation(): String
(傳回 Translation Service 相關說明)

◆ Search Service Proxy Metadata Information Facility

Search Service Proxy Metadata Information Facility 提供了一個機制，讓提供服務的數位圖書館元件輸出 (Export) 系統的性能、藏品資源的內容摘要。

在輸出系統的性能方面，Stanford DL 是以其自定的表格 (Local Table) 方式來表達元件的系統性能，如藏品資源採用的詮釋資料、檢索功能…等等。在 Table 5 中我們摘錄比較重要的系統性能供讀者參考。

至於藏品資源的內容摘要方面，係用來描述藏品資源的特性、索引。Stanford DL 採用 Harvest SOIFs (<http://www.ukoln.ac.uk/metadata/resources/soif>。

html) 的格式作為描述內容摘要之用。Figure 2 是一個簡單的範例，從這個例子我們可以看到在 Title 這個檢索欄位中 “algorithm” 出現了 53 次、“analysis” 出現了 23 次，根據這些索引資訊，我們毋需真正去存取某個數位圖書館元件，就能夠初步判斷此元件中是否有使用者要的資料。

◆ Metadata Repository

Metadata Repository 為一 Stanford DL 的中央 (Central) 資料庫，其內容為彙整前面三部份的詮釋資料相關資訊，作為單一窗口來回答所有與詮釋資料和數位圖書館元件相關的問題。

Table 5 Stanford DL 數位圖書館系統性能的部份描述

系統性能描述	說明
CollectionName	數位圖書館元件名稱
AttriModelNames	支援的詮釋資料種類 (在 Attribute Model Proxy 中定義，可以有多种詮釋資料)
AttriNames	支援的詮釋資料屬性 (在 Attribute Model Proxy 中定義，對於一種詮釋資料不需要支援其全部屬性)
BooleanOps	支援的布林邏輯運算
Proximity	支援的鄰近詞查詢
Truncation	支援的切截方式
StopWordList	停用字列表
Languages	藏品資源的語言
ContentSummaryLinkage	連接到藏品資源的內容摘要 (Figure 2)

```
@SContentSummary{
Version{10}: STARTS 1.0
Stemming{1}: F
StopWords{1}: F
CaseSensitive{1}: F
Fields{1}: T
NumDocs{3}: 892
Field{5}: Title
DocFreq{11023}: "algorithm" 53
DocFreq{analysis" 23
...
Field{6}: Author
DocFreq{1211}: "Ullman" 11, "Knuth" 15
...
}
```

Figure 2 Stanford DL 內容摘要範例

參、DL@NCTU 的詮釋資料架構與數位圖書館互通性

一、DL@NCTU 簡介

DL@NCTU 是交通大學圖書館正在進行的一項數位圖書館計畫。DL@NCTU 的目標主要有四：

- (1) 建立一個基礎建設 (Infrastructure)，以支援各式數位圖書館系統元件間的互通；
- (2) 規劃一套機制，以根據使用者不同的需求協尋並配置適當的數位圖書館服務；
- (3) 提出一個方法，使不同格式的資訊能夠互通；

(4) 探討數位圖書館環境下的經濟相關議題。

為了達成此四項目標，我們規劃了 DL@NCTU 的核心架構 (Figure 3)，並以詮釋資料技術做為此核心架構的基礎。從 Figure 3 可看出 DL@NCTU 為三層式 (Three-Tier) 的架構，此三層分別為：

1. **Metadata Manager**：負責詮釋資料的處理 (Metadata Manipulation)、數位圖書館服務協尋與配置 (Service Allocation)、藏品資源內容檢索 (Content Query)。此層還包括一些便利詮釋資料製作、管理、轉換的工具，如詮釋資料編輯器 (Metadata Editor)、詮釋資料轉換編輯器 (Translation

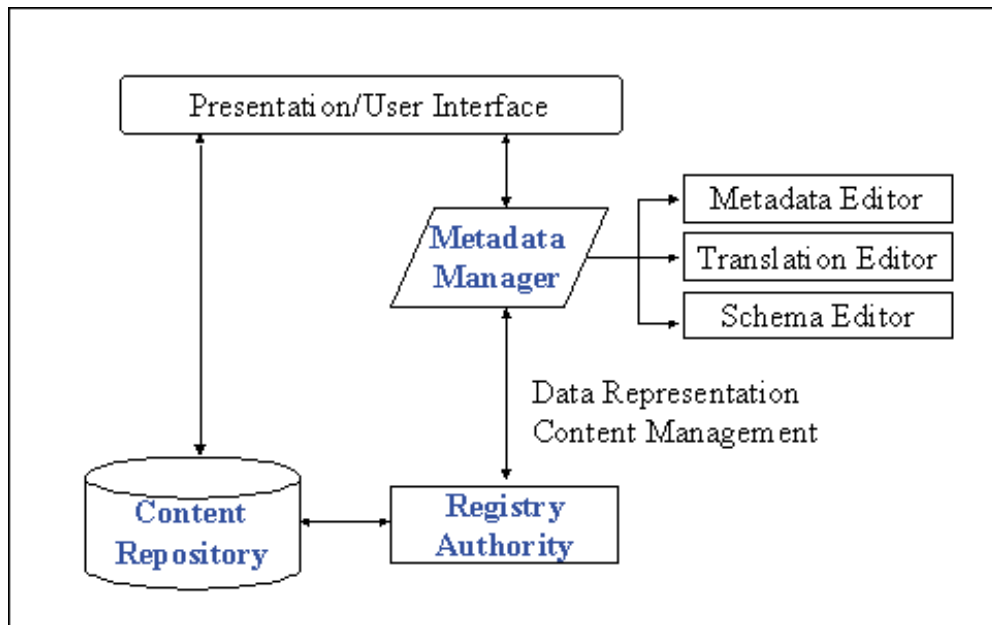


Figure 3 DL@NCTU 系統架構

Editor)、詮釋資料綱要編輯器 (Schema Editor) 等。

2. **Registry Authority**: 負責管理藏品資源及其詮釋資料、所有參與互通運作的數位圖書館元件之相關資訊。
3. **Content Repository**: 為詮釋資料綱要、詮釋資料、藏品資源等三種資訊的貯藏庫，並負責產生識別代碼 (Handle)。

之所以將 DL@NCTU 的系統架構分為三層，主要的考量如下：

1. **Metadata Manager** 負責各式詮釋資料的概念模型設計 (Conceptual Model Design)。
2. **Registry Authority** 提供了一個將藏品資源、各類詮釋資料 (包含藏品資

源和數位圖書館元件的系統性能) 分開管理的機制，讓系統的設計更為透明化、簡單化。此外 **Registry Authority** 還具有索引 (Index) 的功能，提供各類資訊 (藏品資源及其詮釋資料、所有參與互通運作的數位圖書館元件之相關資訊) 的快速取得。

3. **Content Repository** 提供了藏品資源、各類詮釋資料貯存的空間，並賦予每個詮釋資料唯一的本地系統識別代碼 (Native Unique Handle)。
4. 各類詮釋資料的識別由 **Registry Authority** 和本地系統識別代碼達成，不需建置困難度頗高的全域識別代碼產生器 (Global Handle Generator)。

如前所述，DL@NCTU 的架構是以詮釋資料技術為核心，透過詮釋資料技術來協調、整合參與運作的數位圖書館系統元件，以達成數位圖書館系統互通性的目的。而 DL@NCTU 採用的詮釋資料技術具有以下三項特性：

- (1) 為 Container-based 的詮釋資料；
- (2) 擁有一套詮釋資料建構語言 (Metadata Modeling Language, MML, 在【參-二】節中說明)，提供一致且高互通性的方法來表達藏品資源的詮釋資料以及數位圖書館元件的系統性能；
- (3) 擁有一個轉換 (Translation) 的機制，以達成系統元件間資料與服務的互通，此轉換機制同樣由 MML 來表示。

Figure 4 為 DL@NCTU 詮釋資料技術之概念示意圖，可分為四個部份，其中 1-3 屬於 Metadata Manager 負責，4 則為 Registry Authority 與 Content Repository 負責。以下茲說明 1-3 這三項與詮釋資料技術相關的部份：

1. 詮釋資料建構 (Metadata Modeling)；對於要參與 DL@NCTU 運作的所有藏品資源或系統元件都要利用 MML 定義其詮釋資料的種類 (如 Dublin Core 或 EAD) 或系統性能。此部份所要達成的功能與 Stanford DL 的 Attribute Model Proxy、Search Service Proxy Metadata Information Facility 類似。

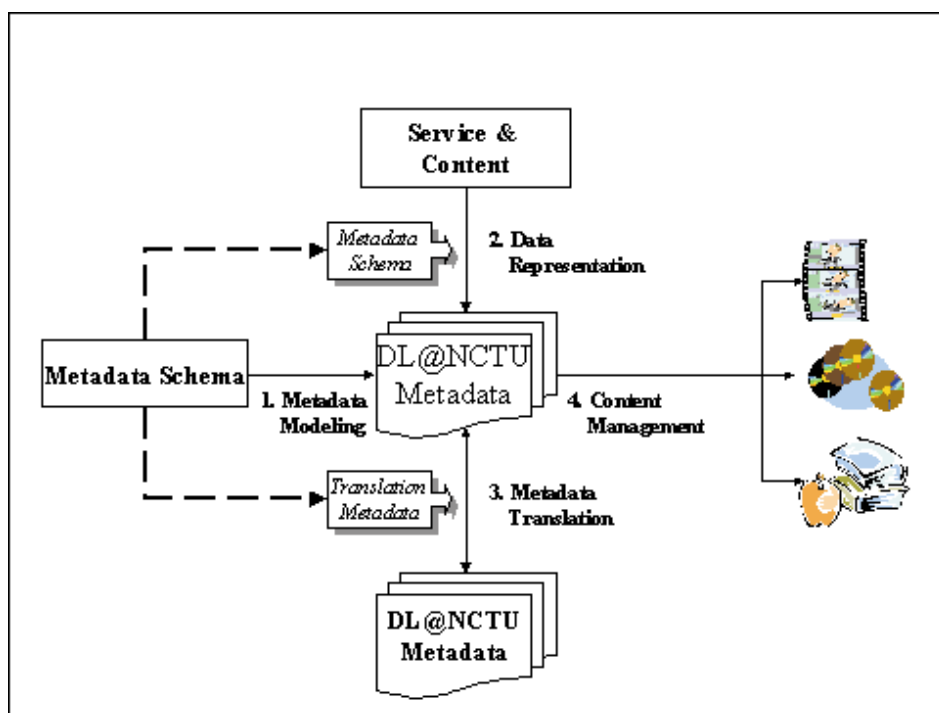


Figure 4 DL@NCTU 詮釋資料技術概念示意圖

2. 資料表示 (Data Representation) ; 在定義好藏品資源的詮釋資料種類或者系統元件的性能之後, 將藏品資源和系統元件性能真正用詮釋資料表示出來。
3. 詮釋資料轉換 (Metadata Translation) ; 當多種藏品資源的詮釋資料或系統元件的性能要互通時, 透過詮釋資料轉換的機制達成。此部份所要達成的功能和 Stanford DL 的 Attribute Model Translation Service 類似。

二、詮釋資料建構語言 (Metadata Modeling Language)

DL@NCTU 中詮釋資料技術的發展理念是希望能夠結合(1)外部協調機制 (External Mediation) 和(2)以規格描述為基礎的元件互動 (Specification-based Interaction) 這兩種互通性方案的優點。為了達成這個目標, 在 DL@NCTU 中我們提出了詮釋資料建構語言 (Metadata Modeling Language, MML) (註 30), MML 擁有強大的能力, 提供標準、一致性、高互通性的語法來描述藏品資源的格式 (Data Format)、系統元件性能 (System Capability)、轉譯規則 (Translation Rule)、不同詮釋資料間的轉譯機制 (Translation Mechanism)。

MML 採用的語法為 XML (Extensible Markup Language), 而 MML 的資料模型 (Data Model) 則是將 RDF (Resource Description Framework) 的資料模型加以簡化並新增 RDF 資料模型的功能。MML 的架構如 Figure 5 所示, 主

要包含 MML 資料模型 (MML Data Model) 和 MML 轉譯服務 (MML Translation Service) 等二部份, 資料模型是用來表示數位圖書館藏品資源的詮釋資料或是數位圖書館系統元件的性能; 轉譯服務則是執行原始詮釋資料 (Source Metadata) 和目標詮釋資料 (Target Metadata) 間的轉換。

Figure 5 也展示了不同種類的藏品資源詮釋資料如何達成互通。舉例而言, 假設我們要將以 MARC 格式著錄的詮釋資料 (Figure 5 中的 Source Database) 轉換成 Dublin Core 格式的詮釋資料 (Figure 5 中的 Target Database), 首先我們必須先用 MML 定義出 MARC 和 Dublin Core 格式 (換句話說就是用 MML 宣告 MARC 和 Dublin Core 的屬性, 這就是所謂的綱要 (Schema) 定義); 其次, 我們根據 MARC 的 MML 綱要 (Figure 5 中的 Source Schema), 將所有 MARC 格式著錄的詮釋資料都用 MML 的方式表達出來 (Figure 5 中的 Source Metadata); 接下來, 我們必須定義出 MARC 和 Dublin Core 這兩種格式轉換的機制, 並同樣用 MML 來表示轉換的機制 (Figure 5 中的 MML Translation Service); 最後, 藉由轉換的機制把用 MARC 格式著錄的 MML 詮釋資料轉換成以 Dublin Core 格式著錄的 MML 詮釋資料 (Figure 5 中的 Target Metadata)。在整個過程中, 藏品資源的詮釋資料和不同詮釋資料間的轉換機制都是用 MML 表述。

在對 MML 有了初步的認識之後,

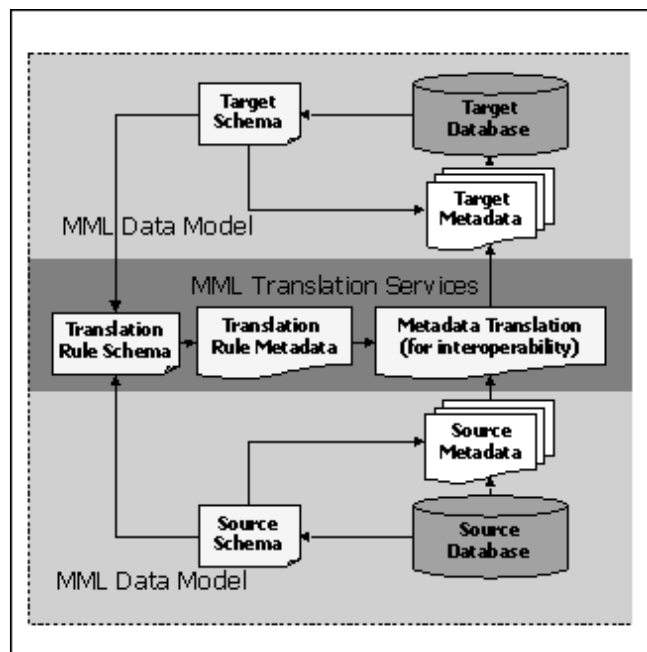


Figure 5 詮釋資料建構語言架構示意圖

接下來我們將詳細介紹 MML 的資料模型、如何用正規的方式來表達 MML 資料模型、並說明如何以 MML 來定義系統元件間的轉換機制。

(一) MML 資料模型

MML 採用階層式 (Hierarchical-

based)、關連式 (Relation-based) 的資料模型，其最基本的單位為資源 (Resource)。MML 資料模型是由實體 (Entity)、屬性 (Attribute)、和值 (Value) 等三部份所構成，而這三部份都是資源，因此能夠用一致的方法來表示 (如 Figure 6)。

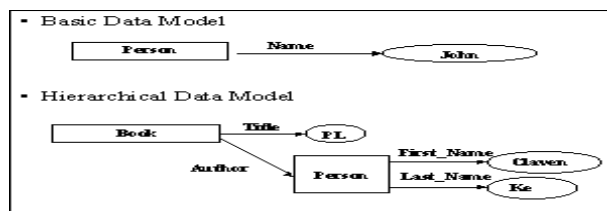


Figure 6 MML 資料模型

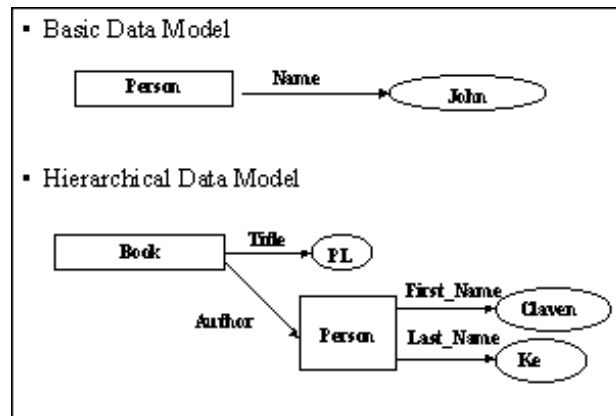


Figure 7 MML 資料模型範例

Figure 7 展示如何利用 MML 來建構一個實體；每一個實體擁有許多屬性，每個屬性都有其特定的值，例如在 Figure 7 的基本資料模型範例中，實體 Person 擁有一個屬性 Name，而此屬性的值為 John。一個資源能夠藉由屬性和另一個資源發生關連，從而發展出階層式的資源關係，例如在 Figure 7 的階層式資料模型中，Book 和 Person 這兩個資源就是透過 Author 的這個屬性發生關連。

(二) 資源的正規描述

要用電腦來運用 MML 時必須用正規的方法來描述 MML 資料模型。在此，我們給資源一個正規的定義：所謂的資源是擁有十個註解式屬性 (Annotation Property) 的基本單位 (參見 Table 6)。(註 31) 之所以要用這種看似複雜的方法來定義資源，主要是

為了將詮釋資料的描述標準化，有助於系統間對於詮釋資料的互通並促進詮釋資料定義方法的一致。在這十個屬性中，Name 和 Identifier 都可用來作為 MML 中識別資源之用。Figure 8 則是以 XML 的語法來描述 MML。

有了資源的正規描述方法之後，我們就可以用 MML 來描述詮釋資料了。在此之前我們必須先區分 Schema (詮釋資料綱要) 和 Metadata (詮釋資料) 這兩個名詞。以物件導向的概念來說，Schema 就代表物件的種類 (Object Class)，而 Metadata 就代表某種物件的個體 (Object Instance)，所以像 Dublin Core 或 MARC 等詮釋資料的著錄格式就是一種 Schema，而利用某個格式來著錄一藏品資源所得到的就是此藏品資源的 Metadata。Schema 和 Metadata 的關係又可以用 Figure 9 表示；Schema 指的是“實體—屬性”這

Table 6 定義資源時所用到的十個註解式屬性

Properties	Comment
Name	資源名稱
Type	資源種類。用來定義某一資源所採用的綱要 (Schema)
Document	資源相關說明
Identifier	指定給資源的單一識別碼。此識別碼可為資料庫的鍵值 (Key)、URI、或其他可資區分不同資源的識別碼。
Version	資源定義版本
Maximum	資源最多可出現的次數
Occurrence	
Obligation	指示此資源是否一定得出現
Definition	完整說明資源概念、特質的陳述，亦可用以說明資源的值域 (Domain & Value Range)
Language	資源定義的語言
Registration Authority	負責註冊管理此資源的組織

MML Syntax

```

MML ::= '<'mmlPrefix'>' resources* '</mmlPrefix'>'
mmlPrefix ::= 'Model' | 'Model_Instance'
resources ::= properties* | attributeItems* | stringvalue
properties ::= '<'propertyTag'>' propertyValue '</propertyTag'>'
propertyTag ::= 'MML_Name' | 'MML_Type' | 'MML_Doc' | 'Identifier' | 'MML_Version' | 'MML_RA' |
'MML_Language' | 'MML_Definition' | 'MML_Obligation' | 'MML_MO'
propertyValue ::= 'string' | 'integer' | 'float' | 'boolean' | stringvalue
attributeItems ::= '<attributeTag'>' resource '</attributeTag'>'
attributeTag ::= 'Attribute' | attributename

```

Figure 8 MML 語法

部份，而 Metadata 則包括“實體—屬性—值”。

在 Table 7 中我們用一個例子來說

明如何以 MML 來定義詮釋資料綱要和詮釋資料。在這裡我們定義了書和作者兩種資源的綱要：只要是書，就

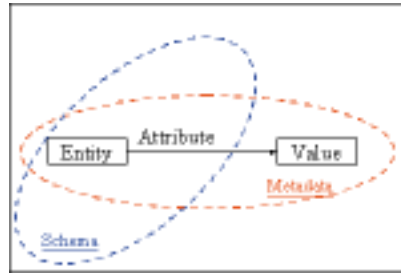


Figure 9 詮釋資料網要 (Schema) 與詮釋資料 (Metadata)

Table 7 以 MML 定義資源的詮釋資料網要 (Schema) 和定義資源個體的詮釋資料 (Metadata)

Book Schema (書籍綱要, 亦即定義“書”這種資源的詮釋資料—需要具備書名和作者這兩個屬性, 其中書名屬性的值為一字符串, 而作者屬性的值則為定義在 Author Schema 中的 Author 資源)	
<pre> <Model> <MML_Name>Book</MML_Name> <MML_Type>Book</MML_Type> <MML_Doc>定義書的詮釋資料綱要</MML_Doc> <Identifier>Book_ID1</Identifier> <MML_Version>Version_1</MML_Version> <MML_RA>NCTU_DB</MML_RA> <MML_Language>English</MML_Language> <MML_Definition>Unlimited</MML_Definition> <MML_Obligation>Unlimited</MML_Obligation> <MML_MO>Unlimited</MML_MO> </pre>	<pre> <Attribute> <MML_Name>Title</MML_Name> <MML_Type>String</MML_Type> <MML_Doc>定義書的屬性一：書名</MML_Doc> </Attribute> <Attribute> <MML_Name>Author</MML_Name> <MML_Type>@Author</MML_Type> <MML_Doc>定義書的屬性二：作者</MML_Doc> </Attribute> </Model> </pre>
Author Schema (作者綱要, 亦即定義“作者”這種資源的詮釋資料—需要具備名和姓這兩個屬性, 這兩個屬性的值均為字符串)	
<pre> <Model> <MML_Name>Author</MML_Name> <MML_Type>Author</MML_Type> <MML_Doc>定義作者的詮釋資料綱要</MML_Doc> <Identifier>Author_ID</Identifier> <MML_Version>Version_1</MML_Version> <MML_RA>NCTU_DB</MML_RA> <MML_Language>English</MML_Language> <MML_Definition>Unlimited</MML_Definition> <MML_Obligation>Unlimited</MML_Obligation> <MML_MO>Unlimited</MML_MO> </pre>	<pre> <Attribute> <MML_Name>First_Name</MML_Name> <MML_Type>String</MML_Type> <MML_Doc>定義作者的屬性一：名</MML_Doc> </Attribute> <Attribute> <MML_Name>Last_Name</MML_Name> <MML_Type>String</MML_Type> <MML_Doc>定義作者的屬性二：姓</MML_Doc> </Attribute> </Model> </pre>
Book Metadata—真正定義一本書, 這本書的書名為 PL、作者則為右欄中所定義的作者(Author_ID2)	真正定義一個作者, 姓和名分別為 Ke 和 Claven。這位作者為左欄中所定義的書籍(Book_ID2)的作者
<pre> <Model_Instance> <MML_Type>Book</MML_Type> <Identifier>Book_ID2</Identifier> <Book> <Title>PL</Title> <Author>Author_ID2</Author> </Book> </Model_Instance> </pre>	<pre> <Model_Instance> <MML_Type>Author</MML_Type> <Identifier>Author_ID2</Identifier> <Author> <First_Name>Claven</First_Name> <Last_Name>Ke</Last_Name> </Author> </Model_Instance> </pre>

應該要有書名 (Title) 和作者 (Author) 這兩種屬性；只要是作者就應該要有名 (First_Name) 和姓 (Last_Name) 這兩種屬性；有了綱要之後，我們就可以實際定義特定書 (Book Instance) 或特定作者 (Author Instance) 的詮釋資料，例如在 Table 7 中我們就定義了一本名為“PL”的書，這本書的作者為 Author_ID2；而根據書的綱要，我們發現 Author 這個屬性的值階層式地指向另一個定義在 Author Schema 中的資源 (<MML_Type>@Author</MML_Type>) (@表示指向另一種資源)，所以我們可以推論出這本書的作者為 Claven Ke。要強調一點，數位圖書館系統元件也是資源的一種，所以數位圖書館系統元件的性能也能夠用 MML 來描述。

(三) MML 轉譯服務

MML 轉譯服務藉由轉譯函數

(Translation Function) 和轉譯模板 (Translation Rule Template) 來達成二種詮釋資料綱要間的轉換，它可以用來執行兩種藏品資源著錄格式 (如 MARC 與 Dublin Core) 間的轉換，或者兩種數位圖書館系統元件性能的轉換。MML 轉譯模板也是 MML 綱要的一種，其綱要定義如 Table 8 所示，共包含 FromSchemaName、ToSchemaName、TranslationRule 等三個屬性，其中，FromSchemaName、ToSchemaName 分別為原始綱要名稱和目標綱要名稱，而 TranslationRule 則定義一條或以上由原始綱要轉換到目標綱要的轉譯規則。在 TranslationRule 中，我們可以對詮釋資料的屬性執行一些基本運算，目前我們實作的基本運算列在 Table 9 中。

Figure 10 為一個 MML 轉譯服務的範例。這個範例將 Book Schema (Table 7) 轉譯成 Book II Schema (Table 10)。Figure 10 共有兩條轉譯規則，其中第

Table 8 轉譯模板綱要定義

Translation_Rule_Template Schema (轉譯模板綱要)	
<Model>	<Attribute>
<MML_Name>Attribute_Model_Translation_Service</MML_Name>	<MML_Name>FromSchemaName</MML_Name>
<MML_Type>Attribute_Model_Translation_Service</MML_Type>	<MML_Type>String</MML_Type>
<MML_Doc>定義轉譯服務模板</MML_Doc>	</Attribute>
<Identifier>Translation_Template_ID1</Identifier>	<Attribute>
<MML_Version>Version_1</MML_Version>	<MML_Name>ToSchemaName</MML_Name>
<MML_RA>NCTU_DB</MML_RA>	<MML_Type>String</MML_Type>
<MML_Language>English</MML_Language>	</Attribute>
<MML_Definition>Unlimited</MML_Definition>	<MML_Name>TranslationRule</MML_Name>
<MML_Obligation>Unlimited</MML_Obligation>	<MML_Type>String</MML_Type>
<MML_MO>Unlimited</MML_MO>	<MML_MO>Unlimited</MML_MO>
	</Attribute>
	</Model>

Table 9 轉譯規則的基本運算

資源總類	基本運算
Integer	Add, Sub, Multiply, Divide
String	String Concatenate
Float	Add, Sub, Multiply, Divide
Boolean	And, Or

Translation Rule Metadata

```

<Model_Instance>
<MML_Type>Attribute_Model_Translation_Service</MML_Type>
<Version>1.0</Version>
<Identifier>Book_Model_I_To_Book_Model_II_Translation</Identifier>
<MML_Doc>Translate metadata of book model I into metadata of book model II</MML_Doc>
<FromSchemaName > Book</FromSchemaName >
<ToSchemaName >Book_II</ToSchemaName >
<TranslationRule>Book_II.Title = Book.Title</TranslationRule>
<TranslationRule>Book_II.Author = Book.Author.Last_Name +"," + Book.Author.First_Name
</TranslationRule>
</Model_Instance>

```

Figure10 MML 轉譯服務範例—從 Book schema 轉譯到 Book II Schema

Table 10 Book II Schema

Book II Schema (書籍原始綱要, 亦即定義“書”這種資源的詮釋資料—需要具備書名和作者這兩個屬性, 其中書名與作者的屬性值均為字串)	
<pre> <Model> <MML_Name>Book_II</MML_Name> <MML_Type>Book_II</MML_Type> <MML_Doc> 定義 Book_II 的詮釋資料綱要 </MML_Doc> <Identifier>BookII_ID1</Identifier> <MML_Version>Version_1</MML_Version> <MML_RA>NCTU_DB</MML_RA> <MML_Language>English</MML_Language> <MML_Definition>Unlimited</MML_Definition> <MML_Obligation>Unlimited</MML_Obligation> <MML_MO>Unlimited</MML_MO> </pre>	<pre> <Attribute> <MML_Name>Title</MML_Name> <MML_Type>String</MML_Type> <MML_Doc>定義書的屬性一：書名</MML_Doc> </Attribute> <Attribute> <MML_Name>Author</MML_Name> <MML_Type>String</MML_Type> <MML_Doc>定義書的屬性二：作者</MML_Doc> </Attribute> </Model> </pre>

Table 11 MML、RDF、Stanford DL Metadata Architecture 的比較

	MML	RDF	Stanford DL
資料模型	同 RDF 資料模型，並可直接用資源名稱參照到特定資源	階層式資料模型	階層式資料模型
轉譯機制	1. 透過轉譯模板完成 2. 轉譯模板同樣可用 MML 定義，達成詮釋資料與轉譯機制表示法的一致性 3. 轉譯規則具基本運算能力	未支持（由程式設計者自行設計轉譯規則）	僅定義轉譯機制界面，需由程式設計者自行設計轉譯規則
採用語法	採用 XML 語法，互通性較高	採用 XML 語法，互通性較高	採用專屬語法，互通性較低
資源參照	可用資源名稱或資源單一識別碼階層式地參照資源。	URI	未說明

一條規則表示經過轉譯後 Book II 的 Title 屬性值等於 Book 的 Title 屬性值；第二條規則表示轉譯後 Book II 的 Author 屬性值等於 Book 的 Author 屬性值以“Last_Name, First_Name”的形式呈現。

四) MML 與 RDF、Stanford DL Metadata Architecture 的比較

在 Table 11 中，筆者從資料模型 (Data Model)、轉譯機制 (Translation Mechanism)、採用語法 (Syntax) 和資源參照 (Resource Reference) 等幾個面向，來比較本文提出的 MML 和 RDF、Stanford DL Metadata Architecture 的差別，從這個比較中我們可以很明顯地看出 MML 的優點所在。

三、MML 案例研究：VUCS@NCTU

為驗證 MML 的可行性，我們將 MML 的概念應用在虛擬聯合目錄系統 (Virtual Union Catalog System, VUCS) 的設計上 (註 32)，並稱此一虛擬聯合目錄為交通大學圖書館虛擬聯合目錄系統 (VUCS@NCTU)。(註 33) 聯合目錄系統乃一整合多個圖書館館藏資訊的系統，其目的在使讀者僅需透過單一的使用界面即可得知多館的館藏資訊。聯合目錄系統的製作一般分為實體聯合目錄系統與虛擬聯合目錄系統等二種，前者係指有一集中式資料庫來儲存多館的館藏資訊，各參與聯合目錄的圖書館必須透過一特定的機制將其館藏資訊儲存到此集中式資料庫中，讀者則藉由檢索此集中式資料庫得知各館館藏資訊；後者則是由各館獨立維護本身的館藏

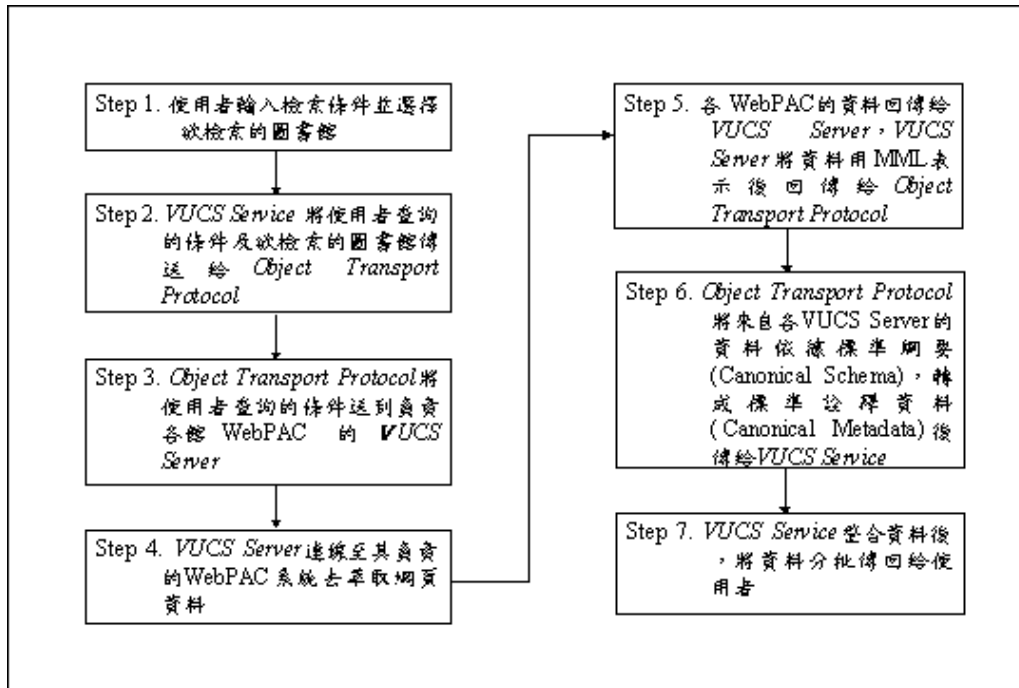


Figure 11 VUCS@NCTU 系統運作流程

查詢系統，當讀者在虛擬聯合目錄系統執行檢索時，系統透過網路即時到各參與館的自動化系統中檢索資料，再由虛擬聯合目錄系統彙整各參與館的檢索結果。

VUCS@NCTU 為數位圖書館系統互通的一個應用，它採用外部協調機制達成互通性，各館的館藏查詢系統可視為 VUCS@NCTU 下獨立運作的系統元件，透過 VUCS@NCTU 將這些元件整合起來。VUCS@NCTU 的特色在於當使用者檢索時，VUCS@NCTU 會即時連結至各館的 WebPAC 進行檢索，並針對各館 WebPAC 傳回的查詢

結果進行分析並萃取必要資訊，完全不需改變各館 WebPAC 的內部設定或與各館的底層資料庫做溝通。

Figure 11 為 VUCS@NCTU 的運作流程，Figure 12 則為本系統架構，共分為 VUCS Server、Object Transport Protocol、VUCS Service 等三部分：

1. VUCS Server; VUCS Server 在 VUCS@NCTU 中扮演外部協調機制的重要角色，其功用是和特定的 WebPAC 溝通。針對每個參與 VUCS@NCTU 運作的 WebPAC，我們必須賦予其專屬的 VUCS Server 設定，比如說各館 WebPAC 的網址或是檢索

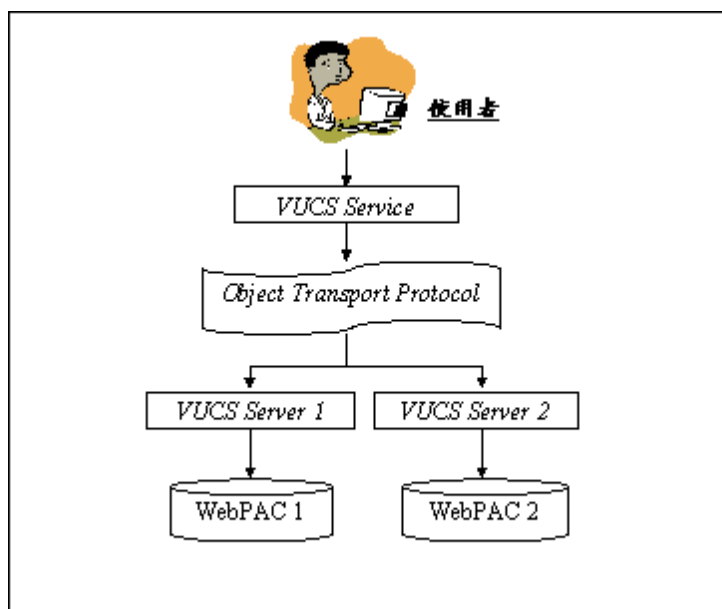


Figure 12 VUCS@NCTU 系統架構

程式所必須接受的參數等等，只要正確地設定好各館的 VUCS Server，則使用者只需在本系統（VUCS Service 層）執行檢索，本系統就會自動連結到已設定好的 VUCS Server 抓取資料。當 VUCS Server 接收到上層傳來的命令，便會參照設定到所負責的 WebPAC 抓取網頁資料。為了正確地分析出各館 WebPAC 的資料，我們必須為每各館 WebPAC 設定其專屬的 VUCS Server 以供分析並萃取資料，一旦某 WebPAC 有所改變，我們只需重新設定其專屬的 VUCS Server 即可。此外，由於各館 WebPAC 所提供的網頁資料格式不同，例如有些會提供書名、作

者和出版社，有些則提供書名、ISBN、作者、出版年及索書號，所以各 VUCS Server 萃取得到的資料格式也不盡相同，為此，在本系統中，我們利用 MML 為各館的 WebPAC 都制定其專屬的 MML 綱要，並據以將各館回傳的資料包裝成 MML 詮釋資料。

2. Object Transport Protocol; VUCS Server 所萃取的資料在根據各館的 MML 綱要包裝成 MML 詮釋資料後會上傳到 Object Transport Protocol，此層負責將不同綱要的詮釋資料轉換成標準綱要（Canonical Schema）的標準詮釋資料（Canonical Metadata），再將轉換完成的資料上傳至 VUCS



Figure 13 VUCS@NCTU 系統展示

Service，讓 VUCS Service 彙整來自各 WebPAC 的資料並供使用者檢閱。為了達成詮釋資料轉換的任務，在此層必須用 MML 定義標準綱要，並用 MML 制定標準綱要和各館專屬綱要間的轉換機制。

3. VUCS Service: VUCS Service 的任務為資料的收集與整合，將傳送上的標準詮釋資料彙整後編成網頁文件供使用者檢視，同時也提供一個整合的使用介面讓使用者下達檢索命令。

由 VUCS@NCTU 的系統架構可知，MML 在 VUCS@NCTU 中所扮演的角色有：(1) 定義各館 WebPAC 的資料格式（即詮釋資料綱要），(2) 定義 VUCS@NCTU 標準綱要，(3) 定義各館 WebPAC 資料與標準詮釋資料間的轉換機制。Figure 13 為 VUCS@NCTU 的雛形系統展示。

肆、結語

本文針對數位圖書館系統的互通性議題以及詮釋資料在其中所扮演的

角色做了廣泛而深入的探討。文中並介紹交通大學圖書館發展的DL@NCTU系統架構，並提出一套為達成數位圖書館系統互通性而發展的詮釋資料建構語言（Metadata Modeling Language, MML）。MML 是由資料模型（MML Data Model）和轉譯服務（MML Translation Service）等二部份構成，前者用來定義數位圖書館藏品資源所使用的詮釋資料格式、以及數位圖書館系統元件的性能，後者則用來轉換不同格式的詮釋資料。MML 採用與 RDF 類似的資料模型，並使用 XML 語法。文中也從資料模型、轉譯機制、採用語法和資源參照等幾個角度，來分析 MML 和 RDF、Stanford DL Metadata Architecture 間的差異。我們也將 MML 的概念應用在交通大學圖書館虛擬聯合目錄系統（VUCS@NCTU）的實作

上，驗證 MML 的可行性。

關於數位圖書館系統的互通性，Cleveland 在「Digital Libraries: Definitions, Issues, and Challenges」一文中（註 34），曾提及數位圖書館系統的互通性很難有一套通盤的解決之道，往往一種互通性方案僅能應用在一定範疇的數位圖書館系統中。（註 35）儘管如此，藉由全球的專家學者不斷地探討互通性議題，筆者深信未來不同數位圖書館系統間的互通將會愈來愈容易，從而讓使用者有更便利的數位圖書館整合服務可以使用。

伍、致謝

感謝國科會對本論文研究所提供的一切補助。計畫編號 NSC89-2218-E009-009。

註釋

- 註 1：陳亞寧，「另類圖書館：電子圖書館綜觀」，資訊傳播與圖書館學 5 卷 3 期（民國 88 年 3 月），頁 141-150。
- 註 2：吳政歡，「電子圖書館時代的大學圖書館」，大學圖書館 1 卷 2 期（民國 86 年 4 月），頁 111-122。
- 註 3：薛理桂，「美英兩國電子圖書館之發展：兼論我國發展之途徑」，資訊傳播與圖書館學 3 卷 3 期（民國 86 年 3 月），頁 23-33。
- 註 4：吳政歡，「淺談電子圖書館的發展」，圖書館學刊 25 期（民國 85 年 6 月），頁 23-31。
- 註 5：陳亞寧，「化危機為契機：試論虛擬圖書館對圖書館界的衝擊」，中國圖書館學會會報 54 期（民國 84 年 6 月），頁 61-73。
- 註 6：G. Cleveland, "Digital Libraries: Definitions, Issues and Challenges," UDT Occasional paper #8, 998, <<http://www.ifla.org/VI/5/op/udtop8/udtop8.htm>>
- 註 7：Digital Library Federation, "A Working Definition of Digital Library," 1998, <<http://www.clir.org/diglib/dldefinition.htm>>

- 註 8：原文為 “Digital libraries are organizations that provide the resources, including the specialized staff, to select, structure, offer intellectual access to, interpret, distribute, preserve the integrity of, and ensure the persistence over time of collections of digital works so that they are readily and economically available for use by a defined community or set of communities.”
- 註 9：E. Fox ed. , “Source book on Digital Libraries,” TR 93-95, Dept. of Computer Science (Virginia Tech., 1993)
- 註 10：C. Lynch and H. Garcia-Molina, “Interoperability, Scaling, and the Digital Libraries Research Agenda,” IITA Digital Libraries Workshop (1995)
- 註 11：H. M. Gladney et al. , “Digital Library: Gross Structure and Requirements: Report from a March 1994 Workshop,” IBM Research Report RJ9840(IBM Almaden Research Center, 1994)
- 註 12：A. Paepcke, C-C K. Chang, H. Garcia-Molina and T. Winograd, “Interoperability for Digital Libraries WorldWide,” Communications of the ACM, 41:4(1998):33-43.
- 註 13：同註 6。
- 註 14：原文為 “One thing digital libraries will not be is a single, completely digital system that provides instant access to all information, for all sectors of society, from anywhere in the world. This is simply unrealistic. This concept comes from the early days when people were unaware of the complexities of building digital libraries. Instead, they will most likely be a collection of disparate resources and disparate systems, catering to specific communities and user groups, created for specific purposes. They also will include, perhaps indefinitely, paper-based collections. They will be a collection of disparate systems and resources connected through a network, and integrated within one interface, most likely a Web interface or one of its descendants.”
- 註 15：M. Baldonado, C-C K. Chang, L. Gravano and A. Paepcke (1997), “The Stanford Digital Library Metadata Architecture,” International Journal on Digital Libraries, 1:2:108-121.
- 註 16：同註 12，頁 37。
- 註 17：同註 12，頁 37-42。
- 註 18：同註 15。
- 註 19：M. Genesereth and S. Ketchpel, “Software Agents,” Communications of the ACM 37:7 (1994):48-53.
- 註 20：P. C. Weinstein, W. P. Birmingham and E. H. Durfee, “Agent-Based Digital Libraries: Decentralization and Coordination,” IEEE Communications Magazine, 37:1(1999): 110-115.
- 註 21：陳雲華，「數位時代資訊組織之發展趨勢」，中國圖書館學會會報 64 期（民國 89 年 6 月），頁 49-60。

- 註 22：陳雪華，「網路資訊組織與 MetaData 之發展」，圖書館學刊 12 期（民國 86 年 12 月），頁 21。
- 註 23：吳政敷，「從元資料看未來資料著錄的發展趨勢」，資訊傳播與圖書館學 4 卷 2 期（民國 86 年 12 月），頁 42-52。
- 註 24：陳雪華、張玉欣，「臺大電子圖書館與博物館數位化資源之組織」，圖書館學與資訊科學 23 卷 2 期（民國 86 年 10 月），頁 51-65。
- 註 25：吳政敷，「三個元資料格式的比較分析」，中國圖書館學會會報 57 期（民國 85 年 12 月），頁 35-45。
- 註 26：A. J. Gilliland-Swetland, "Defining Metadata," in Introduction to Metadata: Pathways to Digital Information, ed. M. Bace (Los Angeles: Getty Research Institute, 1998), 1-8.
- 註 27：同前註。
- 註 28：同註 15。
- 註 29：同註 15，頁 114。
- 註 30：S. S. Huang, H. R. Ke. W. P. Yang, "Interoperability of Cooperative Databases with Metadata," Fourth World Multiconference on Systemics, Cybernetics and Informatics (2000).
- 註 31：MML 資料的正規描述是採用 ISO 11179 標準（和 Dublin Core 相同），請參見 ISO 11179 Specification and Standardization of Data Elements, Parts 1-6. <<ftp://sdct-sunsv1.ncsl.nist.gov/x3l8/11179/>>
- 註 32：K. Coyle, "The Virtual Union Catalog: A Comparative Study," D-Lib Magazine (March 2000).
- 註 33：曾志軒、黃夙賢、柯皓仁、楊維邦，「虛擬聯合目錄系統中擁有共同結構之網頁文件資料萃取」，2000 年臺灣區網際網路研討會 (TANET 2000)（民國 89 年 10 月），頁 475-481。
- 註 34：同註 6。
- 註 35：原文為 "Interoperability across digital libraries-of technical architectures, metadata, and document formats-will also only likely be possible within relatively bounded systems developed for those specific purposes and communities."