

以鏈結資料建置圖書館線上公用目錄系統之研究

A Study of Building Online Public Access Catalog with Linked Data

方凱鴻(Fang, Kai-Hong)

國立臺灣師範大學圖書資訊學研究所

柯皓仁(Ke, Hao-Re)

國立臺灣師範大學圖書資訊學研究所

摘要

圖書館的館藏目錄一直擔任著指引讀者館藏資訊的角色，隨著科技的進步，館藏目錄自實體卡片目錄演變為線上公用目錄系統，然而提供的資訊卻大多侷限於圖書館編目的傳統書目資料。儘管不少圖書館線上公用目錄系統採用了 Web 2.0 概念的混搭技術，運用應用程式介面技術整合外部資訊，然而應用程式介面的不一致卻會增加系統維護的困難。

Tim Berners-Lee 提出了語意網的概念，期望藉由建構一個優良的資料取用方式，以利於資料的整合與再利用。而後更進一步提出了所謂的「鏈結資料」，做為實現語意網的最佳實務。妥善運用鏈結資料技術，勢必能豐富圖書館線上公用目錄的外部資訊，並減少系統維護的困難。

為實踐鏈結資料在圖書館線上公用目錄系統的應用，本研究依循鏈結資料生命週期建置一套線上公用目錄系統，除呈現傳統書目資料外，並串聯 DBpedia、VIAF 等既存的鏈結資料集，提供使用者豐富的作品與作者資訊。

Abstract

Library catalogs serve as an essential role to guide readers the positions and information of library holdings. With the advance of science and technology, library catalogs evolve from card catalogs into online public access catalogs (OPACs). In order to give readers additional information, libraries have used APIs a lot into OPACs. However, the inconsistent APIs may increase the difficulties of maintaining OPACs.

The concept of Semantic Web aims at facilitating the integration and re-use of machine-readable and machine-understandable data. To realize Semantic Web, the principles of linked data has been proposed. To ameliorate the difficulties of

maintaining OPACs caused by inconsistent APIs, and bring plentiful information to library patrons at the same time, linked data may be a feasible solution.

This study aims to establish a linked-data OPAC system by following the life cycle of linked data. In addition to traditional bibliographical information, the proposed system integrates existent linked data sets such as DBpedia and VIAF to provide users abundant work and author information.

關鍵詞彙：鏈結資料、線上公用目錄、語意網

Keywords: Linked Data, OPAC, Semantic Web

壹、緒論

自 1990 年代, Tim Berners-Lee 發明了全球資訊網(World Wide Web, 簡稱 Web) 之後, 網路已成為人們獲取資訊的重要管道。智慧型代理人(Intelligent Agent)技術的突破, 更使機器(電腦)一定程度能自主運作、模仿人類社會的行為 (Franklin & Graesser, 1997)。Tim Berners-Lee 繼而提出語意網(Semantic Web)的概念, 他認為語意網就是資料的網路(Web of Data), 充滿著機器可理解(Machine-Understandable)的資訊, 從而促成電腦間的協同工作(Berners-Lee, 1998)。要達到語意網的第一步, 便是以電腦能夠理解的方式來描述任何具體或抽象、實體或虛擬的事物(thing), 以及事物間的關聯性, 而鏈結資料(Linked Data)便是實現語意網的最佳實務(Linked Data, n.d.; 柯皓仁、陳亞寧, 2013)。

圖書館目錄一直擔任著協助讀者查詢館藏資訊的任務, 圖書館書目紀錄具有聚集以及指引的功用, 因此館藏目錄是使用圖書館資源的入口。隨著科技的進步, 圖書館目錄從卡片目錄演變到線上公用目錄(Online Public Access Catalog, OPAC), 由電腦來管理圖書館的館藏資源, 不僅使圖書館方便作業, 也讓讀者能過透過不同的檢索點以較少的時間取得他們所需要的資源。儘管 Google 等搜尋引擎廣受人們歡迎, 但不可否認地, 線上公用目錄系統仍是讀者接觸圖書館館藏的重要橋樑。然而, 線上公用目錄常為讀者所詬病地乃是其僅能得知館藏的書目資料, 較難獲得書籍目次、簡介、書評, 甚至作者相關資訊。即使透過應用程式介面(Application Programming Interface, API)進行資料的混搭(Mashup)來豐富線上公用目錄的內容與功能, 如書籍封面圖片、地圖或是社群網站等, 然而當其混搭的資料愈來愈多, 所需要串連的 API 網址也會越來越多, 由於不同的 API 提供者需要使用不同的參數, 當線上公用目錄使用的 API 數量增加, 對於線上公用目錄的開發與維護也將會愈來愈困難與複雜。

本研究主旨為基於鏈結資料技術，建置圖書館線上公用目錄系統，利用鏈結資料提升檢索結果頁面的資訊量，在使用者進行館藏檢索時，能給予使用者較為多元的資訊，除了該作品本身的資訊還提供了其他作品的關聯，讓使用者在館藏檢索系統中不僅能夠檢索館藏還能夠探索館藏。此外透過鏈結資料統一且一致的查詢語言來簡化資料混搭的步驟，亦能降低圖書館線上公用目錄系統管理與維護的成本。

本文結構如下，第貳節探討與本研究相關的文獻，包含語意網、鏈結資料，以及圖書館線上公用目錄；第參節闡述系統開發環境與架構；第肆節具體說明系統建置步驟與成果；第五節為結論與建議。

貳、文獻探討

本節就語意網、鏈結資料，以及圖書館線上公用目錄等三方面加以說明。

一、語意網

在網際網路和全球資訊網蓬勃發展之後，大量資料在網際網路湧現，無論在資料的發現、取用、整合及再利用皆帶來許多困難。Tim Berners-Lee (Berners-Lee, 2000)認為要解決此一問題，其基礎在於將資料以電腦能夠理解的形式發布於 Web 環境中，形成一個讓電腦能夠直接或間接處理的資料網(Web of Data)，此即語意網 (Semantic Web)的開端。語意網的實現必須仰賴多項網路相關技術，包含：URI、UNICODE、XML、RDF、RDFS、OWL、SPARQL 等，如圖 1 所示(Wikipedia, 2015; 柯皓仁、陳亞寧，2013)，並說明如下：

- (一) 以統一資源識別符(**U**niform **R**esource **I**dentifier, URL)標示任何具體或抽象、實體或虛擬的資源(resource)或事物(thing)。
- (二) 以萬國碼(Unicode)做為文字編碼。
- (三) 以可延伸標記式語言(**eX**tensible **M**arkup **L**anguage, XML)做為電腦間交換資訊的語法。
- (四) 以資源描述框架(**R**esource **D**escription **F**ramework, RDF)描述資源屬性(property)以及資源間的關聯性(relationship)。三元組(triple)是 RDF 資料內容的基本結構，以主詞(subject)、述語(predicate)、受詞(object)的形式來描述資源屬性或資源間的關聯。
- (五) 以 RDF 綱要(RDF **S**chema, RDFS)為 RDF 定義詞彙，建立種類(class)與屬性的階層架構，建構分類法(taxonomy)。

- (六) 以 **Web Ontology Language (OWL)** 進一步擴展 RDFS，用以建構知識本體，使語意網具備推理能力。
- (七) 以 **SPARQL (SPARQL Protocol and RDF Query Language)** 做為檢索 RDF 資料庫的檢索語言。

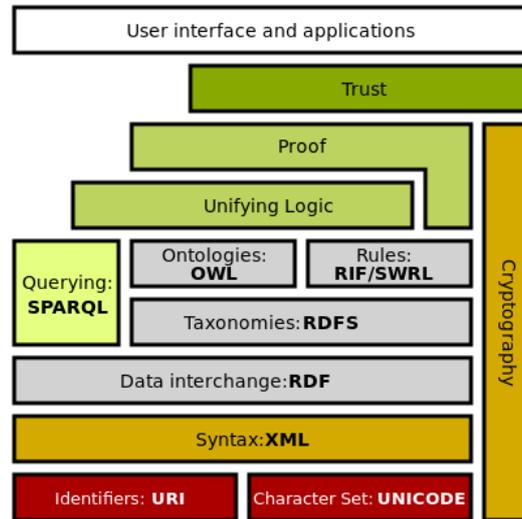


圖 1 Semantic Web Stack

資料來源：Wikipedia. (2015). Semantic Web Stack. Retrieved from https://en.wikipedia.org/wiki/Semantic_Web_Stack.

二、鏈結資料

為了具體實踐語意網，2006 年 Tim Berners-Lee 進一步提出了所謂的「鏈結資料」(Linked Data)，企圖賦予全球資訊網結構化的資料，使資料得以在網際網路上自由互連，彙整同一資源的相關資訊。鏈結資料乃是一組實現語意網的最佳實務 (Linked Data, n.d.)，用以將結構化資料發布在 Web 上，並加以串聯成為一個機器能理解、富含語意和結構化的資料網路。鏈結資料包含以下四大原則 (Berners-Lee, 2006; Heath & Bizer, 2011; 柯皓仁、陳亞寧，2013)：

- (一) 利用 URI 為資源或事物命名。
- (二) 以 HTTP 做為客戶端和伺服器端之間查詢及傳送 URI 的機制，使人或電腦可以查詢特定 URI 所代表事物的相關資訊。
- (三) 伺服器端使用 RDF 與 SPARQL 等標準，提供更多的資訊。當伺服器端接獲客戶端對於特定 URI 的請求時，伺服器端會以標準格式將該 URI 所代表事物的相關資訊傳回給客戶端。若客戶端是「人」，則伺服器端可回

傳 HTML 格式的文件；若客戶端是「應用程式」，則伺服器端可回傳 RDF 格式的文件，以方便客戶端的應用程式再利用這些資訊。

- (四) 對於特定 URI 所代表事物的相關資訊中，應包含與其他相關事物的連結（連結到該事物的 URI），使得事物間得以串連，以達成構築全域資料空間的目標。

鏈結資料的運用包含資料的發佈(Data Publication)以及資料的使用(Data Consumption)。Bauer & Kaltenböck(2001)提出了資料發佈與使用的流程，資料發佈可分為下列八個步驟：(1)分析資料、(2)精簡資料、(3)建立資料模型、(4)選擇適宜的詞彙、(5)指定授權許可、(6)轉換為 RDF 格式、(7)連結其他資料集，以及(8)發佈且推廣至鏈結開放資料雲；資料使用的流程則包含以下七個步驟：(1)具體描述使用狀況、(2)評估相關資料集、(3)確認使用許可、(4)建立使用模式、(5)管理以及更新資料集、(6)建立綜合服務界面及應用服務，以及(7)建立穩定持續的合作關係。

圖 2 為鏈結資料發佈以及使用的流程，依流程順序簡介如下(Thurner, 2013)：

- (一) 擷取(Extraction)：擷取到的資料可分為兩種，非結構式資料與結構式資料，其處理過程有所不同，非結構式資料需經過自然語言處理的三個步驟再轉換成 RDF 格式：

1. 命名實體識別：從文字中取出實體的標籤名稱。
2. 關鍵詞擷取：辨識出主題核心的關鍵詞彙。
3. 關係探勘：從資料來源取出描述實體與關鍵詞的關聯性。

而結構式資料，如關聯式資料庫以及 XML 格式，可以直接撰寫由轉換工具所提供的對應語言來將資料轉換為 RDF。

- (二) 儲存/查詢(Storage/Querying)：擁有大量 RDF 資料之後，必須建立儲存空間以產生索引檔讓查詢更具效益。Endpoint 是能夠接受 SPARQL 語言查詢的伺服器，目前已有不少相關的產品。
- (三) 修訂/創作(Revision/Authoring)：創造新的結構化資訊或是修正現有資訊。
- (四) 連結/融合(Interlinking/Fusing)：資料間互相鏈結，當有不同資料出版者提供相同或相似於某實體的資訊時，應當對這些資訊建立連結。
- (五) 分類/加值(Classification/Enrichment)：由於鏈結資料主要是實例資料，因此較缺乏分類或組織，此時可透過加入高等結構來助於聚合資料以及有效率地查詢。

- (六) 品質分析(Quality Analysis)：建立一套資料品質評估標準，針對提供資料的專家、來源出處和網路資源的信賴度提供品質的參考數據。
- (七) 演進/修復(Evolution/Repair)：建立鏈結資料更新與修復策略。
- (八) 搜尋/瀏覽/發現(Search/Browsing/Exploration)：對鏈結開放資料雲(Linked Open Data Cloud)進行搜尋、瀏覽以及探勘，接著將資料回傳給使用者。

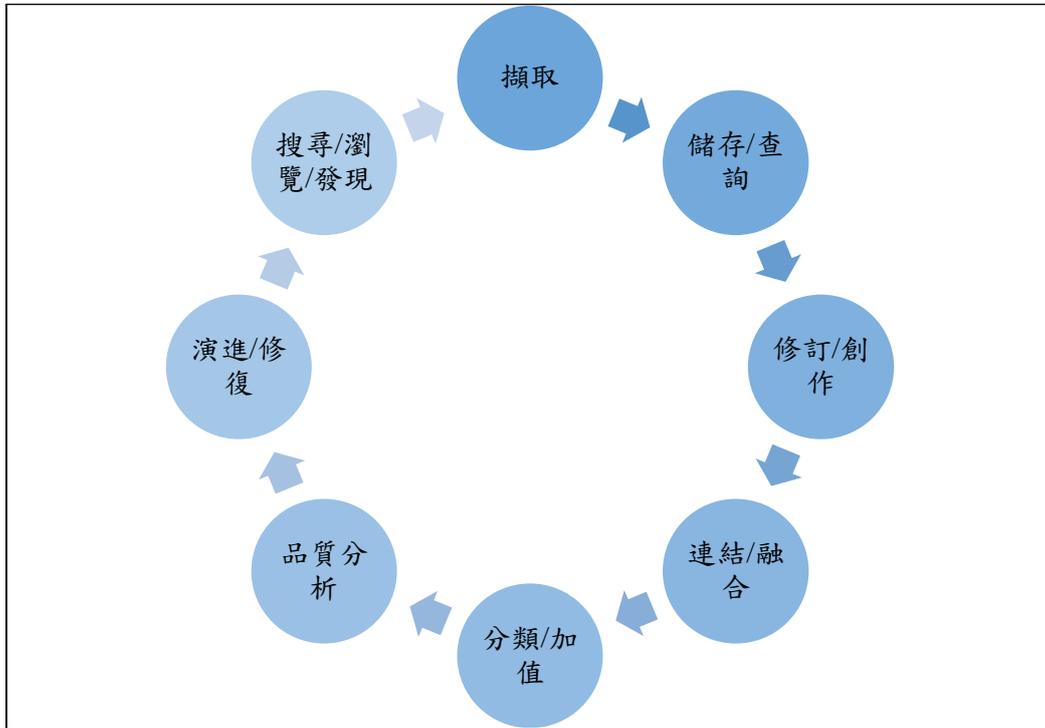


圖 2 LOD Life Cycle

資料來源：Thurner, T. (2013). The Stack. Retrieved from <http://stack.lod2.eu/blog/>

目前鏈結資料已開始廣泛運用於各種領域，如生命科學、地理、社群網路、媒體等各領域，截至 2014 年 8 月為止，LOD 已收錄至少 1014 個資料集 (Schmachtenberg, Bizer, & Paulheim, 2014)。Google 和 Yahoo 等搜尋引擎公司也開始利用鏈結資料來豐富他們的檢索結果。可見透過鏈結開放資料的重新組合可以創造出新的知識與對事物的理解，而這些可帶來在全新的領域上的運用。

至於鏈結資料在圖資界的應用方面，美國國會圖書館將其擁有的各式主題詞 (subject headings)、權威檔 (authority file)、分類表 (classification)、國家代碼 (country)、語言 (languages) 與圖像資料索引典 (Thesaurus for Graphic Materials) 等不同類型的控制詞彙以鏈結資料的方式提供，使用者可以批次下載所有資料。此外，使用者可以輸入關鍵詞彙進行檢索，檢索結果除以網頁格式呈現，亦提供使用者以 RDF/XML、JSON、N-Triples 等格式下載單一檢索結果 (柯皓仁、陳亞寧，2013)。

不僅美國國會圖書館發布鏈結資料，英國大英圖書館(British Library)亦將書目資料以鏈結資料的形式釋出，目前書目資料涵蓋範圍包含已出版書籍和連續性出版品，未來會繼續延伸至不同類型作品。作為 LOD 的一環，大英圖書館鏈結資料的事物屬性描述來自 Bibliography Ontology、Bio、British Library Terms、Dublin Core、FOAF、ISBD、MADS/RDF、RDA、SKOS、WGS84 Geo Positioning 等多種語彙，且能連結至 ISNI、VIAF、LCSH、GeoNames 等鏈結資料集。此外，大英圖書館提供讀者使用 SPARQL 進行語意查詢，讓讀者得以利用這些檢索結果(British Library, n.d.)。

Hannemann & Kett (2010)認為圖書館書目資料多採用圖書館專屬的機讀格式，因為該格式不易被異質機構利用，若採用鏈結資料格式，可讓圖書館資料更易於其他領域交換與再利用。Tillett (2012)認為對於被重複翻譯或發行的作品，系統應讓編目員能快速連結到全球網路中存在的作品/表現形描述資料，並且連結到現存的名稱及主題權威控制詞彙。若採用鏈結資料格式，註冊之詞彙集可重複使用於資料之描述工作，一次建置，所有人共享及共同維護。因此鏈結資料能夠促進資料利用，提升資料之發現，快速取用其他機構產生的資料，將有關資訊進行最大程度的連結，促進網路效益及資料加值。

三、線上公用目錄

自圖書館提供檢索系統讓讀者查找館藏開始，檢索系統就一直有不同的稱呼，一般較為人知的有：電腦目錄(computer catalog)、線上目錄(online catalog)、自動卡片目錄(automated card catalog)、讀者檢索系統(patron access catalog)以及線上公用目錄(online public access catalog)(Hildreth, 1985)，本研究概以線上公用目錄稱之。

依據圖書館學與資訊科學線上字典(Online Dictionary for Library and Information Science, ODLIS)對線上公用目錄的定義是：「一個記錄圖書館館藏之書目資料的資料庫...大部分線上目錄可以查詢作者、書名、主題和關鍵詞，並提供列印、下載或是輸出書目紀錄等功能」(Reitz, n.d.)。

有感於搜尋引擎的興起，線上公用目錄在網路時代逐漸不受到使用者的喜愛，學者開始研究線上公用目錄的改善，並稱在 2000 年之後的線上公用目錄為新世代公用目錄(Next-generation Catalogue)。在 2007 年 Marshall Breeding 在 Library Technology Reports 提出了公用目錄應該有的功能，讓新世代公用目錄增加不同議題，他認為新世代線上公用目錄系統應該具有以下十二項功能：(1)單一的入口網站、(2)有美感的網頁界面、(3)豐富的內容、(4)層面檢索、(5)簡單的關鍵字檢索界

面、(6)相關排序、(7)自動猜測與建議、(8)推薦資料、(9)使用者參與貢獻、(10)RSS 訂閱、(11)結合社群網站、(12)永久連結(Breeding, 2007)。

由於讀者身處數位資源豐富和網路工具取得容易的環境，大多數的線上公用目錄已經不能滿足讀者的需求，使得圖書館的線上公用目錄使用次數逐漸下降，這反映了讀者現在需要更多樣廣泛的內容，而鏈結資料技術或許可為線上公用目錄系統帶來新的樣貌。

參、系統開發環境與架構

本節闡述本研究運用鏈結資料技術建置圖書館線上公用目錄系統(以下簡稱本系統)時的系統開發環境與架構。

在資料來源方面，本系統使用 INEX(Initiative for the Evaluation of XML retrieval, INEX) 2013 年的競賽資料，共 280 萬餘筆書目紀錄，作為測試資料集，用以模擬圖書館之館藏環境。INEX 資料集之每一筆書目紀錄皆是一份 XML 文件，書籍資料包含一般圖書館使用的資料欄位，如國際標準書號、題名、出版者、頁數、出版日期以及杜威分類號等，然而為了凸顯鏈結資料可提供豐富資訊這項特色，本系統並未將所有欄位儲存於資料庫中，資料庫欄位列表將於第肆節中說明。

本系統使用之作業系統為 Linux Mint，以 PHP 程式語言撰寫程式，前述 INEX 書目資料則使用關聯式資料庫 MySQL 儲存。其他相關的開發語言還有 HTML 5、CSS 3 以及 Java Script 等，主要編寫工具為作業系統內建之文字編輯器。

由於本系統支援鏈結資料，儲存於關聯式資料庫中的資料必須轉換為 RDF 三元組，為此，本研究使用 D2R 伺服器將關聯式資料庫模擬成三元組。透過 D2R 伺服器的轉換機制，可以在不影響原有資料庫的情形下進行 SPARQL 查詢、瀏覽或是儲存成 RDF 格式。

由於 INEX 資料集十分龐大，為提升檢索效能與支援層面檢索(faceted search)，本研究使用由 Apache 基金會發行的 Solr 來建置索引，Solr 是開放原始碼的全文檢索伺服器，以 Java 的函式庫 Lucene 為核心；而 Lucene 則是一個基於 Java 的全文檢索函式庫，它可為應用程式提供索引和搜索功能。

綜上所述，本研究所使用的系統發展工具如表 1。

表 1 本系統建置工具彙整

| 工具 | 功能簡述 |
|----|------|
|----|------|

| | |
|-----------------------------|--------------------------------|
| Linux Mint | 作業系統環境 |
| PHP、JavaScript、HTML 5、CSS 3 | 程式語言 |
| MySQL | 儲存書目紀錄的關聯式資料庫 |
| D2R | 轉換關聯式資料庫為三元組之工具，以及提供 SPARQL 查詢 |
| Solr | 建置索引、層面檢索 |
| Lucene | 全文檢索函式庫 |

本系統的整體運作流程圖如圖 3 所示，分為「資料後端處理」與「使用者前端檢索」兩階段。在「資料後端處理」階段，首先運用本研究自行撰寫的 XML 處理器(XML Parser)將原始書目資料讀入關聯式資料庫，接著使用索引工具與轉換工具建置索引庫(Index DB)與三元組儲存庫(Triple Store)。在「使用者前端檢索」階段，系統根據使用者輸入的關鍵字，將符合的書目紀錄取出(Record Match)；接著將紀錄中的關鍵字送往其他資料集(如 DBpedia)進行二次查詢(Endpoint Query)，鏈結資料集使用 SPARQL 語法查詢，非鏈結資料集使用 RDF 包裝器(RDF Wrapper)完成之；再來將所取得的資料進行整合(Data Integration)並且呈現(Data Display)給使用者。若使用者對於感興趣的關鍵字進行點擊(Interaction)，本系統會將點擊的關鍵字再次送入系統重複執行上述流程。

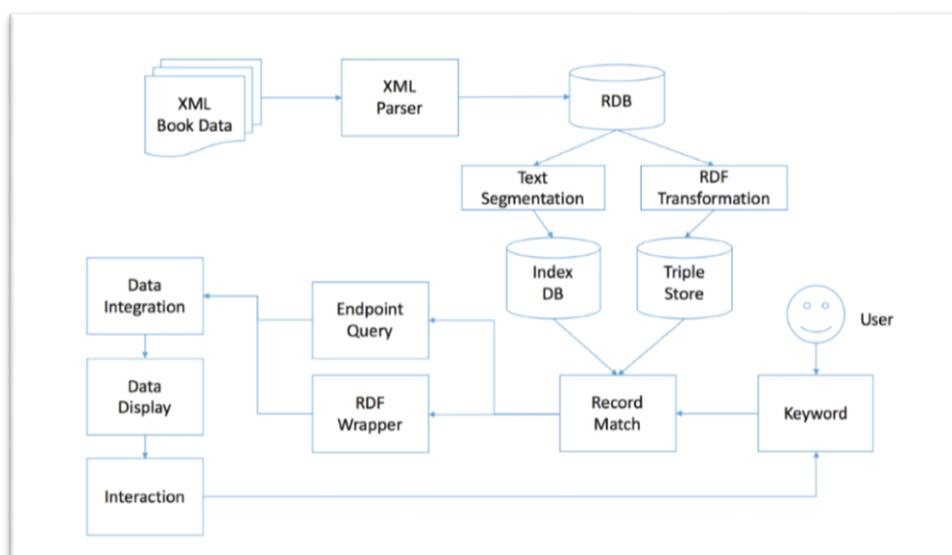


圖 3 系統運作流程圖

圖 4 為本系統之系統架構圖，包含資料層、資料存取整合層、應用層與網路資源等四部分，分別說明如下。

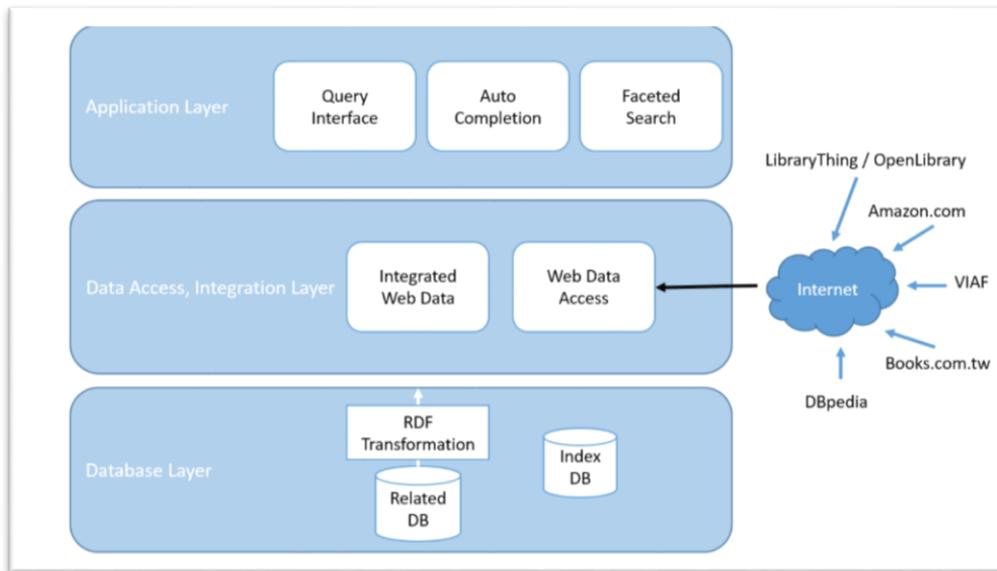


圖 4 系統架構圖

- 一、資料層：本系統的資料來自兩個資料庫，其一為存放書目資料的關聯式資料庫；另一個則是經過斷詞過後的索引庫，因為書目紀錄數量龐大，為了要快速取得符合關鍵字的书目紀錄，乃採用 Solr 建置索引庫。
- 二、資料存取整合層：本系統使用鏈結資料集作為額外資料來源，因此需要設定 SPARQL 查詢和彙整回傳的資訊。
- 三、應用層：本系統在此層級中提供檢索介面、層面分類檢索與自動建議。檢索介面負責接受使用者檢索詞和呈現檢索結果，並與使用者互動；層面檢索幫助使用者過濾檢索結果；自動建議可幫助使用者輸入檢索詞彙時自動補上剩餘字串。
- 四、網路資源：本系統需要存在於網路上的各類鏈結資料集提供相關資料呈現給使用者，而各個資料集又彼此相串連成為資料網，使本系統得以提供使用者除了作品本身的額外資訊。

依據上述之系統架構，本系統最終可達成以下功能：

- 一、使用者使用關鍵字進行檢索，系統回傳之查詢結果可結合 DBpedia 與 LibraryThing 等語意網服務來產生相關作品簡介、書籍封面、作者介紹、套書介紹、電影改編資訊等。

- 二、本系統具有自動建議和自動修正功能，故使用者在檢索過程中無需擔心輸入錯誤或不清楚使用正確主題詞。
- 三、透過虛擬國際權威檔(Virtual International Authority File, VIAF)輔助，可使用不同語言檢索同一物件實體，使用者無需思考該實體之原始名稱。
- 四、書目紀錄使用 RDF 方式儲存，提供讀者下載。每一筆紀錄皆擁有靜態 URI 供使用者方便查找，也可讓開發者再利用。

肆、系統建置步驟與成果

本節主要介紹本系統的建置步驟，並且展示本系統最後所呈現出的樣貌。茲依鏈結資料集與索引建置、系統建置步驟、建置成果之順序闡述。

一、鏈結資料集與索引建置

由於原始資料均是以 XML 格式提供，為了使此資料集存放於資料庫中，故須先處理 XML 文件。本研究透過 PHP 內建的函式庫 SimpleXml 讀取 280 萬筆 XML 檔案，本研究撰寫一份 PHP 程式讀取每一筆書目紀錄，過濾出適合的欄位後將所有資料寫入 MySQL 資料庫。本系統的資料庫命名為 books，資料庫內有一張資料表 books，表 2 為本系統的書籍資料欄位列表。

表 2 書籍資料欄位列表

| 欄位名稱 | 資料型態 | 標籤說明 | 附註 |
|-----------------|--------|--------|-------|
| id | bigint | 流水號 | 設定為主鍵 |
| title | text | 題名 | |
| author | text | 作者 | |
| isbn | bigint | 國際標準書號 | |
| publicationdate | int | 出版年 | |

本研究使用鏈結資料技術建置線上公用目錄系統，為支援 RDF，因此透過 D2R 伺服器建置 SPARQL Endpoint。D2R 伺服器是一個將關聯式資料庫發布為鏈結資料的專用工具，支援目前主流的關聯式資料庫如 Oracle、MySQL、PostgreSQL、Microsoft SQL Server 等，D2R 伺服器會在存取關聯式資料庫時建立一份對映檔，使關聯式資料庫中的表格和欄位彼此產生關聯，在使用者檢索之時得以進行語意檢索。對映檔使用 mapping language 編寫，是一種類似 RDF 以及知識本體概念的

語言。對映檔可依照需求進行更動，管理者需要分析表格中欄位之間彼此的關聯，選擇適合的知識本體進行描述，使物件之間的關聯得以完善。而 SPARQL Endpoint 是一個網路服務，用於接收 RDF 的查詢語言 SPARQL，然後再回傳查詢結果。透過 D2R 伺服器的轉換機制，可以不影響原有的資料庫形態來進行 SPARQL 查詢、瀏覽或是儲存成 RDF、JSON 等主流格式。D2R 主要進行兩個步驟建構 SPARQL Endpoint，分述如下。

(一) 建立對映檔

為了使關聯式資料庫得以使用 SPARQL 查詢，需建立對映檔，對映檔可透過一般文字編輯器編寫，表 3 為本系統對映檔的局部內容。在對映檔的結構上，一個資料表視為一個類別(class)，資料表中的每一筆紀錄視為該類別中的實體(individual)，而資料表中的欄位則被視為該實例之屬性(attribute)。

為了將資料庫中的資料轉換為具有一致標準描述標籤的 RDF 格式資料，必須在對映檔中選擇適當的描述詞彙替關聯式資料庫的欄位建立關聯，本研究參考已發布鏈結資料之各國國家圖書館的知識本體，如大英國家圖書館、德國國會圖書館等，配合本系統所選用的資料欄位，得出如表 4 的鏈結資料欄位對映表。其中，欄位 label 與 amazon 並不存在於 MySQL，而是從對映檔設定中產生而成。欄位 label 中註記書目紀錄中的流水號，欄位 amazon 中記載網址 URL，URL 是本系統抓取亞馬遜網路書店的資訊頁面。

(二) 執行 D2R 伺服器

D2R 伺服器須透過終端機啟動，終端機為 Linux 中的文字操作介面，透過使用者輸入文字指令來進行各項操作，輸入指令後即可啟動 D2R 伺服器。完成啟動後，伺服器將會讀取對映檔來建立鏈結資料服務，使得關聯式資料庫中的資料，經過伺服器轉換後能以 RDF 格式呈現，使用者對 D2R 伺服器下 SPARQL 查詢指令也可正常運行。此外，D2R 伺服器還提供多種存取方式，包含 URI 參照以及鏈結資料的瀏覽與下載。

本系統採用 Solr 以提升檢索效能與實踐層面檢索，Solr 伺服器的建置可依照以下流程進行：

- (一) 環境設定：欲實踐 Solr 的檢索功能，首先必須將 MySQL 的資料匯入至 Solr，而 Solr 需要經過設定才能讀取 MySQL 之資料。Solr 的設置是透過內部的 XML 文件進行操控，相關的 XML 設定檔皆放置於 conf 資料夾底

下，schema.xml 是用於設定 Solr 如何處理資料欄位，而 solrconfig.xml 可用來設定 Solr 伺服器的各項功能。

- (二) 開啟終端機，輸入指令以執行 Solr 伺服器。
- (三) 資料匯入：完成設定兩個檔案後使 Solr 得以取用 MySQL 資料庫。匯入 Solr 伺服器的書目紀錄將會同時建立索引檔。
- (四) 建立查詢：完成伺服器啟動與資料匯入後，Solr 伺服器會接收來自使用者的查詢詞，再回傳查詢結果給使用者，而本系統將解析查詢結果，產生完整的檢索畫面呈現給使用者。

表 3 對映檔局部內容

```
@prefix map: <#> .
@prefix db: <> .
@prefix vocab: <vocab/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix d2rq: <http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/0.1#> .
@prefix jdbc: <http://d2rq.org/terms/jdbc/> .
@prefix bibo: <http://purl.org/ontology/bibo/> .
@prefix dc: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
map:database a d2rq:Database;
    d2rq:jdbcDriver "com.mysql.jdbc.Driver";
    d2rq:jdbcDSN "jdbc:mysql://localhost/books";
    d2rq:username "root";
    d2rq:password "XXXXXXXX";
    jdbc:autoReconnect "true";
    jdbc:zeroDateTimeBehavior "convertToNull";
```

```

.
# Table books
map:books a d2rq:ClassMap;
    d2rq:dataStorage map:database;
    d2rq:uriPattern "books/@@books.id@@";
    d2rq:class vocab:books;
    d2rq:classDefinitionLabel "books";
.

map:books__label a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:books;
    d2rq:property rdfs:label;
    d2rq:pattern "books #@@books.id@@";
.

map:books_title a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:books;
    d2rq:property dc:title;
    d2rq:propertyDefinitionLabel "books title";
    d2rq:column "books.title";

```

表 4 鏈結資料欄位對映表

| 欄位名稱 | RDF 屬性 | 說明 |
|-----------------|------------|-------------------|
| id | -- | 系統號不轉換 |
| label | rdfs:label | 使用 RDF Schema 的標籤 |
| title | dc:title | 使用都柏林核心集的題名 |
| author | dc:creator | 使用都柏林核心集的創作者 |
| isbn | bibo:isbn | 使用書日本體的 ISBN |
| publicationdate | dc:issued | 使用都柏林核心集的發行時間 |

| | | |
|--------|----------------|-----------|
| amazon | bibo:annotates | 使用書日本體的註記 |
|--------|----------------|-----------|

二、系統建構

完成 D2R 伺服器以及 Solr 伺服器的建置之後，鏈結資料服務的基本雛型已經完成。本研究以上述服務為基礎，為線上公用目錄系統實作鏈結資料的加值服務。本研究使用的電腦伺服器主機為國立臺灣師範大學圖書資訊學研究所所有，建立的系統與服務安裝於 <http://140.122.104.78/opac/j3.php>。圖 5 為本線上公用目錄系統之元件圖，逐一解說如下。

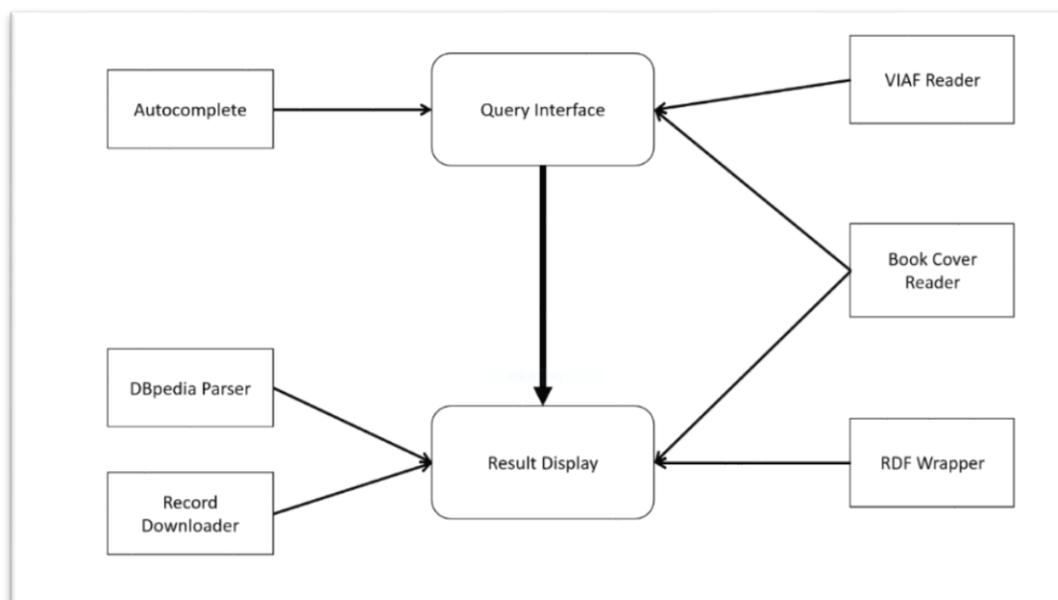


圖 5 系統元件圖

(一) 自動建議(Autocomplete)

在使用者開始輸入字串後，透過 JavaScript 即時傳送使用者所輸入的內容，並且將內容傳送至 Solr 伺服器中進行比對，將符合內容的前十筆紀錄回傳於系統，整理成一份清單供使用者選擇，使用者選擇清單中的項目之後，系統會自動填入使用者尚未輸入完畢的字串，讓使用者可以進行查詢。

(二) 讀取 VIAF(VIAF Reader)

若使用者輸入作者欄位進行查詢，系統會將關鍵字送往 VIAF 所提供之 API 比對最接近的人名，如表 5 所示為向 VIAF 檢索海明威(Hemingway)的範例。

表 5 VIAF 檢索語法(以海明威為例)

<http://www.viaf.org/viaf/AutoSuggest?query=Hemingway>

若有符合項目，VIAF 將回傳相關人物之 VIAF 編號，本系統會讀取來自 VIAF 的作者紀錄，將作者的出生日、國籍、性別以及著作呈現於系統中，如表 6 所示為海明威於 VIAF 中的紀錄。

表 6 海明威之 VIAF 紀錄

```
<ns2:VIAFCluster xmlns="http://viaf.org/viaf/terms#" xmlns:foaf="http://xmlns.com/foaf/0.1/" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:void="http://rdfs.org/ns/void#" xmlns:ns2="http://viaf.org/viaf/terms#">
<ns2:viafID>97006051</ns2:viafID>
<ns2:Document about="http://viaf.org/viaf/97006051/">
<ns2:inDataset resource="http://viaf.org/viaf/data"/>
<ns2:primaryTopic resource="http://viaf.org/viaf/97006051"/>
</ns2:Document>
<ns2:nameType>Personal</ns2:nameType>
<ns2:sources>
```

此外，本系統還會進行延伸查詢，透過 SPARQL 查詢語法取得 DBpedia 中該作者的生平資訊，如表 7 為向 DBpedia 檢索海明威生平資訊之 SPARQL 語法。

表 7 以 SPARQL 查詢 DBpedia 中的海明威生平資訊

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```

PREFIX dbp: <http://dbpedia.org/property/>

SELECT distinct ?title WHERE {{
?s foaf:isPrimaryTopicOf <http://en.wikipedia.org/wiki/Ernest_Hemingway>.
?x dbo:author ?s.
?x rdfs:label ?title.
}
union
{
?s foaf:isPrimaryTopicOf <http://en.wikipedia.org/wiki/Ernest_Hemingway>.
?x dbp:author ?s.
?x rdfs:label ?title.
}
FILTER ( lang(?title) = 'en' )
}

```

(三)讀取書籍封面(Book Cover Reader)

本系統運用以下方法取得使用者查詢書籍的封面。首先，若書籍資訊本身附帶 ISBN，本系統會直接連接 LibraryThing 或是 OpenLibrary 取回書籍封面。由於兩者會有書籍封面缺失的狀況發生，為減少此情況發生，本系統使用兩個來源互相補足，當本系統偵測到其一網站無法提供圖片，將會選擇另外一個來源，若兩者皆無法提供圖片，本系統亦會補上替代圖片告知使用者圖片無法取得。LibraryThing 與 OpenLibrary 皆有提供 API 讓開發者輸入書籍之 ISBN 即可取得圖片，表 8 為該 LibraryThing 和 OpenLibrary 網站所提供之 API 網址，其中 LibraryThing 須額外帶入開發者金鑰，加入會員後可免費取得該金鑰。

表 8 書籍封面來源網址

| 來源網站 | API 網址 |
|--------------|---|
| OpenLibrary | http://covers.openlibrary.org/b/isbn/{isbn}-L.jpg |
| LibraryThing | http://covers.librarything.com/devkey/{api-key}/large/isbn/{isbn} |

若資料庫僅有書名資訊，則是先透過 OCLC 所提供之 API 查詢該書籍之 ISBN，再使用該 ISBN 取得 LibraryThing 或是 OpenLibrary 的書籍封面。表 9

為在 OCLC 查詢《老人與海(the old man and the sea)》之範例。

表 9 連線 OCLC 查詢書籍資訊的語法

```
http://classify.oclc.org/classify2/Classify?title=the+old+man+and+the+sea&summary=true
```

(四) 讀取 DBpedia (DBpedia Parser)

在詳細書目畫面中，除了原本儲存於本機端的資料之外，本系統也透過 SPARQL 查詢語言提供來自 DBpedia 的各類資訊，如書籍資訊、作者資訊。表 10 為查詢《老人與海》之作者資訊的 SPARQL 語法。此外，運用 SPARQL 還可延伸查詢套書與電影改編等相關資訊。套書資訊來自 DBpedia 中的「dbp:followedBy」與「dbp:precededBy」欄位，分別代表著「下一部作品」與「前一部作品」，表 11 為查詢《哈利波特:鳳凰會的密令(Harry Potter and the Order of the Phoenix)》上下部作品之 SPARQL 語法。改編資訊則來自 DBpedia 中的「dct:subject」欄位，該欄位註記該部作品所屬的主題，若該作品的所屬主題有「dbc:British_novels_adapted_into_films」，則表示該部作品的主題為「英國小說改編為電影」，然而其他國家亦會有作品改編為電影，因此可在 SPARQL 加入廣義詞的條件，使檢索主題不僅為「英國小說改編為電影」，而是「書籍改編為電影」(dbc:Books_adapted_into_films)，表 12 為查詢《哈利波特:鳳凰會的密令》電影改編資訊之 SPARQL 語法。

表 10 連線 DBpedia 查詢《老人與海》之作者資訊的 SPARQL 語法

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix dbp: <http://dbpedia.org/property/>
select ?u ?w where {
  ?t rdfs:label ?s.
  ?t dbp:author ?w.
  ?w rdfs:comment ?u.
  ?s bif:contains "\"The Old Man and The Sea\"".
  FILTER (lang (?u)='en').
}
```

表 11 連線 DBpedia 查詢《哈利波特:鳳凰會的密令》套書資訊的 SPARQL 語法

```

select distinct ?a ?b where {
  ?t rdfs:label ?s.
  OPTIONAL{ ?t dbp:precededBy ?a}
  OPTIONAL{ ?t dbp:followedBy ?b}
  ?s bif:contains "\" Harry Potter and the Order of the Phoenix \".
}

```

表 12 連線 DBpedia 查詢《哈利波特:鳳凰會的密令》電影改編資訊的 SPARQL 語法

```

prefix dbo: <http://dbpedia.org/ontology/>
prefix dbc: <http://dbpedia.org/resource/Category:>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
prefix dct: <http://purl.org/dc/terms/>
select distinct ?t (COUNT(?t) AS ?count) where {
  ?t rdfs:label ?s.
  ?t dbo:abstract ?u.
  ?t dct:subject ?r.
  optional{ ?t rdf:type dbo:WrittenWork}.
  ?r skos:broader{,3} dbc:Books_adapted_into_films.
  ?s bif:contains "\" Harry Potter and the Order of the Phoenix \".
  Filter (lang(?u)='en').
}

```

(五) RDF Wrapper

除了 DBpedia，本系統亦擷取來自網路書店(如亞馬遜與博客來)的商品資訊，透過網頁讀取技術，替書籍的相關資訊建立專屬頁面以及 URI。若未來該網路書店發佈了鏈結資料集，本系統也只需替換相對應的超連結即可繼續提供最新資訊。本研究所給予的 URI 原則如下：

http://140.122.104.78/opac/{provider}/{ISBN}

參數{provider}為資訊提供者，亞馬遜書店為 amazon，博客來書店為 bookstw；參數{ISBN}為該作品之 ISBN 編號。例如書籍《Harry Potter and the Order of the

Phoenix》的 ISBN 編號為「0747561079」，則來自亞馬遜書店的資訊將會給予 URI 為「<http://140.122.104.78/opac/amazon/0747561079>」。

(六) 書目下載功能(Record Downloader)

本系統所提供的書目下載格式為 RDF 以及 JSON。前者為鏈結資料常用格式，後者為網路程式開發所常用的格式。每一筆書目皆擁有靜態網址，其命名原則如下：

<http://140.122.104.78/opac/{資料格式}/{流水號}>

參數{資料格式}可選擇書目紀錄的資料格式，RDF 與 JSON 為該參數之選項；參數{流水號}則是該筆書目紀錄之流水編號。例如書籍《Harry Potter and the Prisoner of Azkaban》的靜態網址為「<http://140.122.104.78/opac/bookid/14154672319296>」，「<http://140.122.104.78/opac/RDF/14154672319296>」為該筆書目之 RDF 下載連結。兩者皆有提供靜態網址，可作為未來程式開發或資訊整合的媒介。

三、系統展示

系統的功能以整體而言可分為兩個部分，第一個是書籍檢索介面，第二個則是書目瀏覽介面。

(一) 書籍檢索介面

使用者進入系統之後，由上方文字方塊中選擇一種檢索點(書名或是作者)，輸入關鍵字進行查詢，系統會將關鍵字與本機端的書目資料進行查詢，之後將查詢結果呈現於畫面中。其中，若使用者進行作者檢索，本系統亦同時將關鍵字傳送至 VIAF 比對，並回傳該作者之相關資訊以及著作，將其整理為作者總覽圖表呈現於檢索結果上方。如圖 6 所示，位於中央的作者總覽圖表主要使用 VIAF，VIAF 可提供作者的基本訊息，如姓名、生日、國籍等，以及該作者的著作，另外，本系統額外從 DBpedia 補充該作者的簡介。



圖 6 書籍檢索畫面

(二) 書目呈現介面

使用者從檢索結果列表中，選定一則欲了解的書籍，點選書名連結可查看詳細書目頁面。此頁面分為三個區塊，如圖 7 所示，分述如下。第一為左上角的簡要書目區塊，該處資料是由本系統內建資料庫所提供，而書籍封面是由 LibraryThing 或 OpenLibrary 提供。第二為右上角的額外相關資訊區塊，該處資料是透過 SPARQL 至 DBpedia 即時查詢，為使用者補充書籍與其他資源的相關性資訊，亦提供書目紀錄下載，提供的格式為 RDF 與 JSON 格式。第三則是下方的詳細書目區塊，共有三個頁籤分別為「詳細資訊」、「作品介紹」以及「作者介紹」，「詳細資訊」頁籤使用亞馬遜網路書店的商品資訊來補足本系統原有資料庫中的資訊，「作品介紹」則依照作品的語言別選擇不同來源，外文書籍使用 DBpedia，中文書籍使用博客來網路書店，「作者介紹」則和作者資訊圖表相同使用 DBpedia 作為資料來源。本系統為來自此二網路書店的資訊建立 URI，使本系統未來可支援該資訊提供者的鏈結資料集。

| | | |
|--|---|---|
|  | 題名 Harry Potter and the Order of the Phoenix 作者 Rowling, J. K. 出版社 Bloomsbury 出版年 2004 駐記 http://140.122.104.78/opac/amazon/0747561079 ISBN-13 9780747561071 ISBN-10 0747561079 | 書目紀錄下載: RDF JSON 這部作品有被改編為電影, 請參見 維基百科 系列作品介紹 Harry Potter and the Goblet of Fire ↑↑ -本作品- ↓↓ Harry Potter and the Half-Blood Prince |
| | Book Cover From LibraryThing | |
| | 詳細資訊 作品介紹 作者介紹 | |
| | 媒體類型: Paperback 頁數: 768 pages 高: 19.812 cm 長: 12.9032 cm 寬: 4.0894 cm ISBN: 0747561079 | |
| | 說明: Biography J. K. (Joanne Kathleen) Rowling was born in the summer of 1965 at Yate General Hospital in England and grew up in Chipstow, Gwent where she went to Wyedean Comprehensive. Jo left Chipstow for Exeter University, where she earned a French and Classics degree, and where her course included one year in Paris. As a postgraduate she moved to London to work at Amnesty International, doing research into human rights abuses in francophone Africa. She started writing the Harry Potter series during a Manchester to London King's Cross train journey, and during the next five years, outlined the plots for each book and began writing the first novel. Jo then moved to northern Portugal, where she taught English as a foreign language. She married in October 1992 and gave birth to her daughter Jessica in 1993. When her marriage ended, she returned to the UK to live in Edinburgh, where "Harry Potter & the Philosopher's Stone" was eventually completed and in 1996 she received an offer of publication. | |
| amazon.com Customer Reviews Average Customer Review ★★★★★ (7,540 customer reviews) 7,540 Reviews 5 star 71% 4 star 19% 3 star 8% 2 star 1% 1 star 1% | | |

圖 7 書目呈現畫面

伍、結論與建議

本研究以鏈結資料技術建置了線上公用目錄系統，展現了鏈結資料在書籍資料上的應用，方便使用者檢索書籍與作者，更進一步地引導使用者瀏覽線上公用目錄系統來接觸其他館藏。本研究採用鏈結資料原則，使用知識本體為書籍資料進行描述，透過 SPARQL 來連結其他資料集，豐富線上公用目錄系統所呈現的書籍資訊，並讓使用者能透過 RDF 或 JSON 下載書籍資料，使本系統同時扮演資料提供者與使用者的角色。此外透過鏈結資料統一且一致的查詢語言來簡化資料混搭的步驟，亦能降低圖書館線上公用目錄系統管理與維護的成本。針對本研究所開發的系統，未來將持續進行以下改善：

- 一、進行使用者評估：較傳統圖書館線上公用目錄，本系統運用鏈結資料技術整合網路上豐富的作品、作者相關資訊。為了解使用者對於這些資訊的接受度，未來將進行使用者評估。此外，尚可進一步了解讀者使用圖書館線上公用目錄系統的資訊需求，作為後續增加鏈結資訊的參考依據。
- 二、詞彙檢索的改良：本研究對於資料鏈結和映對方式侷限在單一詞彙，而無法針對一個完整句子進行斷詞、斷句和文字屬性的分析，若能將鏈結資料與自然語言分析技術相結合，勢必有助於提升網路上大量網頁資料鏈結和關聯，讓鏈結開放資料的各領域資料集更加豐富，對於具體實踐語意網一定會有莫大的助益。
- 三、與其他外部資料集的整合：本系統整合了亞馬遜、博客來網路書店以及 DBpedia 等的資訊，未來希望嘗試整合其他專業領域之資料集，讓使用者不僅僅只接觸書籍本身的資訊，更能探索至其他形式的相關作品。
- 四、中文鏈結資料的支援：由於目前中文的鏈結資料集仍然缺乏，在資料的呈現上需要使用英文資料來補足，甚至是無法呈現。因此未來國內的圖書館、博物館等其他資訊機構釋出足夠的鏈結資料集，將可使資訊更加適合華人閱讀。若是建置為多語系的鏈結資料集，還可以將資料集整合至鏈結資料雲供全世界使用。
- 五、提升檢索準確度：本系統之作者資訊總覽是透過 VIAF 查詢再彙整 DBpedia 資訊而成，其中在 VIAF 查詢方面，本系統採用 VIAF 回饋結果的第一筆紀錄，因此有可能產生查詢結果與檢索項目不符合的狀況，以英文姓名為例，若使用者只輸入姓氏海明威(Hemingway)，則本系統將會選擇厄尼斯特·海明威(Ernest Miller Hemingway)，因此其他名字的作者將不會顯示。本系統現階段使用自動建議功能進行輔助，但此並非強制

性選擇，因此成效有限，未來相關研究可以嘗試其他作法幫助使用者欲檢索的作者，使檢索準確度得以提升。

致謝

本研究為行政院科技部專題研究計畫「以 FRBR、Linked Data、Crowdsourcing 設計圖書館線上公用目錄系統之研究」(NSC 102-2410-H-003-121-MY3)之部分成果。

參考文獻

- Bauer, F., & Kaltenböck, M. (2011). *Linked Open Data: The Essentials*. Retrieved from <http://bls.buu.ac.th/~f55361/08Aug22/LOD/EBFFFd01.pdf>.
- Berners-Lee, T. (1998). Semantic Web Roadmap. Retrieved from <http://www.w3.org/DesignIssues/Semantic.html>.
- Berners-Lee, T. (2000). *Weaving the Web: The original design and ultimate destiny of the World Wide Web*. HarperBusiness, 1st edition.
- Berners-Lee, T. (2006). Linked Data - Design Issues. Retrieved from <http://www.w3.org/DesignIssues/LinkedData.html>.
- British Library (n.d.). Collection Metadata Data Services. Retrieved from <http://www.bl.uk/bibliographic/datafree.html>.
- Breeding, M. (2007). Next-Generation Library Catalogs. *Library Technology Reports*, 43(4). Retrieved from <http://www.librarytechnology.org/lrg-displaytext.pl?RC=12723>.
- Franklin, S., & Graesser, A. (1997). Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In *Intelligent Agents III Agent Theories, Architectures, and Languages* (pp. 21-35). Berlin: Springer.
- Hannemann, J., & Kett, J. (2010). Linked Data for Libraries. In Proc of the World Library and Information Congress of the Int'l Federation of Library Associations and Institutions (IFLA).
- Heath, T., & Bizer, C. (2011). Linked Data: Evolving the Web into a Global Data Space (1st edition). *Synthesis Lectures on the Semantic Web: Theory and Technology*, 1:1, 1-136. Morgan & Claypool. Retrieved from <http://linkeddatabook.com/editions/1.0/>.
- Hildreth, C. R. (1985). Online Public Access Catalog. *Annual Review of Information Science and Technology*, 20, 233.
- Turner, T. (2013). The Stack. Retrieved from <http://stack.lod2.eu/blog/>.
- Linked Data (n.d.). *What is the relationship between Linked Data and the Semantic Web? Frequently Asked Questions (FAQs)*. Retrieved from <http://linkeddata.org/faq>.
- Reitz, J. M. (n.d.). ABC-CLIO - ODLIS — Online Dictionary for Library and Information Science. Retrieved from http://www.abc-clio.com/ODLIS/odlis_o.aspx#opac.
- Schmachtenberg, M., Bizer, C., & Paulheim, H. (2014). State of the LOD Cloud 2014.

Retrieved from <http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/>.

Tillett, Barbara B. (2012). *Looking to the Future with RDA*. Retrieved from <http://catweb.ncl.edu.tw/userfiles/cat07/file/RDA/1338368798.pdf>.

Wikipedia. (2015). Semantic Web Stack. Retrieved from https://en.wikipedia.org/wiki/Semantic_Web_Stack.

柯皓仁、陳亞寧(2013)。鏈結資料在圖書館的應用。在全國學術電子資訊資源共享聯盟，海量資料：學術研究新境界研討會，臺北市。Retrieved from <http://concert.stpi.narl.org.tw/Images/upload/2013/11/HaoRenKe-Report.pdf>.