



Integrated searching in Taiwan: the LIPS-DOI experiment

Kuang-hua Chen

National Taiwan University, Taipei, Taiwan

Refereed article received
25 July 2006
Approved for publication
12 September 2006

Abstract

Purpose – There is an active effort by major libraries in Taiwan to offer integrated searching as part of their information services. The purpose of this paper is to report a low-cost and high-flexibility system, < LIPS-DOI >, which can carry out integrated searching with respect to resource management.

Design/methodology/approach – The paper first reviews the related techniques and then designs an integrated search system based on the concept of resources management. The reported system, < LIPS-DOI >, is composed of three modules: enumeration module, description module, and resolution module. The various digital contents are first imported into < LIPS-DOI > and thereafter the system is put into operation.

Findings – A low-cost and high-flexibility system for integrated searching can be implemented and put into operation. In addition to digital objects, physical objects could also be managed and searched in the proposed < LIPS-DOI > system. This system will redirect users to the original system in which these physical objects reside.

Research limitations/implications – Due to the nature of management, registered users have to manage their own objects using the features provided by < LIPS-DOI > .

Originality/value – Such a system will empower library end users to find materials of mixed formats residing in disparate locations from a single interface. It was also designed with an eye toward integration with its DOI counterpart in the future.

Keywords Object-oriented methods, Resource management, Computer software, Taiwan

Paper type Technical paper

Introduction

The concept “Digital Library” was formulated to embody using the browser as a portal into all library resources through the use of integrated directory and search technologies (Marshall, 1997). In such an internet-powered environment the difficulty lay beyond the mere mechanics of preparing the data sources and managing their access. Instead, the difficulty lay in performing these tasks against a data source of massive size. In other words the enormous amount of data encompassed in these resources has created another level of complexity in what is otherwise a moderate task. How to provide a system on which the resource holder can easily manage their data and search against contents of disparate formats across different locations has become an important research topic. In addition how to provide all of these in a simple and integrated search interface is equally important. In the past some used the Z39.50 protocol technology to enable integrated searching (Lynch, 1997). However, such an



approach drew little attention due to its large overhead and complicated deployment. The Open Archives Initiative (OAI), on the other hand, is considered a lightweight protocol which provides a set of simple verbs or operations (Van de Sompel and Lagoze, 2000). Both methods are of the shared protocol variety. Alternatively, OpenURL is a kind of search technology that allows users to perform searches against scattered resources (Van de Sompel and Hochstenbach, 1999a, b, c).

This paper takes a different approach and address integrated searching against scattered resources in the interest of resource management. This method can be used to complement existing systems that already have shared protocols or search technologies in place. Furthermore, this system provides management for digital rights. Such a system will be able to manage digital and physical content as well as provide an integrated directory for library end-users to perform integrated searches. In the real world resource management systems are already commonplace in the publishing industry. These systems are capable of providing functions such as integrated management, integrated searching and document retrieval. Take the digital object identifier (DOI) system as an example. The DOI system is based on the use of DOI, which has the quality of being persistent. The DOI system provides a framework for managing intellectual property and is used in areas such as digital publishing. Presently, the development of the DOI system is overseen by the International DOI Foundation (IDF), which charges a handsome fee for DOI membership. There are currently no organisations in Taiwan that are DOI members.

This paper addresses how to build a low-cost and high-flexibility digital resource management system, < LIPS-DOI > (LIPS stands for Language & Information Processing System, <http://lips.lis.ntu.edu.tw>). < LIPS-DOI > will use a commonly applied metadata format, which should lend it added appeal. The research takes the following approach:

- (1) *Literature survey.* We survey related research in the areas of identifier encoding, metadata and resolution policies as applied to Internet resource management and searching.
- (2) *System analysis and design.* Functional modules in the < LIPS-DOI > system are analysed. These modules are: enumeration, description and resolution. We investigate what they do and how they do it.
- (3) *System implementation.* We deploy the < LIPS-DOI > system based on a sample set from a library collection containing mixed resources and other online data storage. We also use a time-qualified shared protocol so that our system can synchronise with a globally-standard clock so that the timestamp of a digital document can be properly recorded at its time of registration. This measure also assists in implementing a level of protection for intellectual property.

We implement three modules: the enumeration module, the description module and the resolution module. The enumeration module is responsible for enumerating digital documents. There will be one unique identifier for each resource. The user employs such an identifier to search for the resource no matter where it is located. The description module is responsible for managing the document's metadata. This paper adopts Simple Dublin Core (Simple DC) as the metadata format of choice. The resolution module is responsible for mapping identifiers to Internet addresses. A digital document may contain multiple copies or exist in multiple locations. The resolution module makes the

physical location of a digital object transparent to the user. Its corresponding identifier remains fixed, independent of changes in physical location.

The rest of this paper is organised as follows: the next section describes related works for integrated search, such as: Z39.50, OpenURL, OAI, DOI and their applications; this is followed by a section investigating the DOI system and its usability; a section discussing the < LIPS-DOI > system, the features and its implementation; a section discussing the < LIPS-DOI > application; and a short conclusion.

Related works

Libraries provide information to users, not only information stored on the premises but also in other library locations as well. This is realised by an integrated directory. As the Internet has become an important means of holding and communicating information, providers, including libraries, strive to provide an integrated search service to their users. Integrated directories are the basis for accomplishing this.

We describe Z39.50, OpenURL, OAI and DOI in this section to pave the way for the subsequent discussion. Z39.50 is a communication protocol on top of which an integrated directory can be built. OpenURL uses integrated searching to make possible the concept of integrated directories. OAI uses the concept of a service provider to centralise all scattered directories in one location. DOI also uses a centralised scheme, but it further integrates resource management, resource search and intellectual property protection features.

Z39.50 is an older technology and consequently carries a higher overhead. Its premise was to compensate for the stateless aspect of the HTTP protocol. However, it was too ambitious and eventually produced an overly complex specification with high development costs. In many ways it collapsed under its own weight. However, Z39.50 was useful in proving that it was possible to implement integrated searching using a shared protocol stack. The merit in this alone justifies its place in history. As an improvement over Z39.50, OAI was established. OAI makes the following modifications: it defines six verb commands, simplifies the search process, allows independent applications to run on top of it and provides a framework to allow separate program-modules to communicate with each other. Its design goals are to lessen communication overheads and make searching and storing of digital objects easy (Van de Sompel and Lagoze, 2000). Figure 1 shows the flow diagram for OAI. The latest OAI version is OAI-PMH v2.0. Details concerning its protocol specifications are not the focus of this paper, and interested users can refer to the original source – Van de Sompel and Lagoze (2004).

OpenURL is another approach. It uses a search technology to achieve integrated searching. It is similar to traditional CGI (Common Gateway Interface) in that it allows static web content to become dynamic. In other words it allows the user to send commands such as “search” to effect dynamic interaction. However, OpenURL uses a Link Resolver to resolve various URL searches. OpenURL started in April 1999 when Van de Sompel and Hochstenbach (1999a, b, c) published a series of research papers in *D-Lib Magazine*. Link Resolver plays a key role in OpenURL (refer to Figure 2). Link Resolver stores the metadata of document contents in a centralised location. The target users of the OpenURL mechanism are both library end-users and staff. It also maintains statistics on user searches.

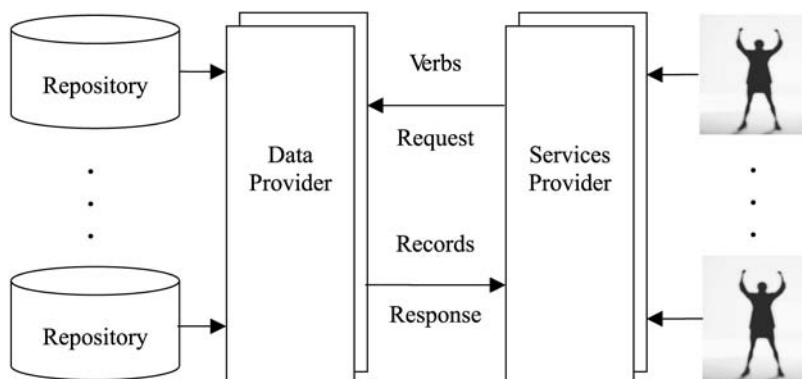


Figure 1. OAI

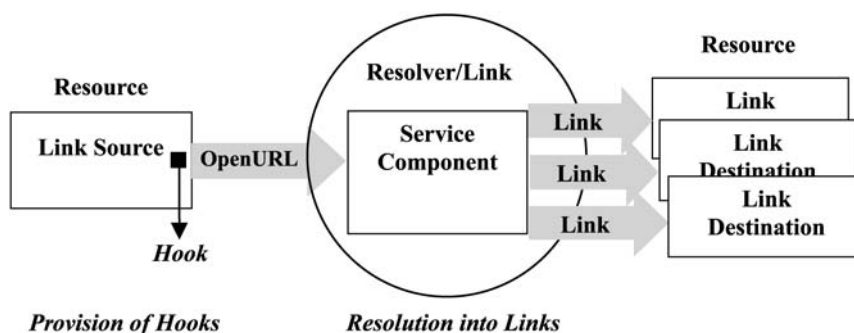


Figure 2. OpenURL

DOI is a type of persistent identifier. Once such an identifier is assigned to a digital object it will not change again. DOI is currently used in electronic publishing and provides a service for digital rights. DOI may eventually become the basis for e-commerce.

The governing body which advocates on behalf of DOI is the International DOI Foundation (IDF), which was founded in 1998 (IDF, 2006a). It serves the intellectual property community in the digital environment by promoting the use of Digital Object Identifier as a common framework for developing content management systems. IDF is responsible for setting policies and addressing the various needs of content holders. Operationally, IDF is a non-profit and member-controlled organisation. Policies are set in a democratic fashion by its participating members, and DOI systems are uniformly supported by all members.

In an effort to advocate and advance DOI major publishers of technical, scientific and medical information around the world came together to establish the CrossRef organisation. CrossRef's mission is to promote the development and cooperative use of new and innovative technologies to speed and facilitate scholarly research. CrossRef allows researchers to link references in their papers to various publishers without needing to worry about when those links should become outdated. This is a real gain for adopting DOI. Currently, there are 1,651 publishers and academics who participate in CrossRef. Their contents are linked through the DOI system. CrossRef currently

contains links with 14,581 journals, 8,321 conference titles and 22,318 book titles. This system contains 21,146,699 records (CrossRef, 2006).

In addition to CrossRef being the biggest organisation promoting and initiating DOI applications, almost all of the applications of DOI are based on the services provided by CrossRef. Few other implementations of DOI-like services are found. Paskin (2005) has discussed the application of DOI for scientific data, but this is still being developed by IDF.

It should be noted that OAI, OpenURL, and DOI began with different sets of ideas, but they all share the same goal in wanting to have an integrated search environment. Furthermore, they could achieve this on the basis of mutual inclusion. For example, DOI does not limit which underlying protocols it must work with, and OAI does not impose its protocols with regard to the identification of digital objects.

DOI overview

The DOI system assigns identifiers that are qualified Uniform Resource Names (URNs) in compliance with RFC1737. Although the DOI system's primary task is to distribute identifiers, it also has other responsibilities. According to IDF's vision, a complete DOI system should be equipped with the following functions:

- enumeration;
- description;
- resolution; and
- policies.

Enumeration

Enumeration deals with the syntax of DOI identifiers. To be accurate, DOI identifiers are alpha-numeric strings. Each DOI is a unique identifier referring to a digital object. Although the DOI system will ensure that no identifier can ever be assigned twice, it is incumbent upon DOI registrants (companies or individuals who have registered with IDF to obtain DOI prefixes) to also support uniqueness in their local assignments. In other words on a policy level the DOI system imposes strict rules on identifier uniqueness.

A digital object may already have another identification system associated with it – for example, a book may have an International Standard Book Number (ISBN) or a code from the library's own system. Nevertheless, DOI can complement these existing systems by appending its codes in the DOI suffixes or tagging them in a metadata. Moreover, DOIs can be used as an extension to existing systems. For example, one can assign DOIs to specific chapters or sections in a book or periodical which already contains an ISBN or International Standard Serial Number (ISSN) without breaking unity. Given any collection of disparate data objects (such as texts, images, audio/video items, software, etc), the DOI system can easily categorise and annotate the objects through the metadata database. The DOI system enables any data object to become uniquely identified across global boundaries in a fast and easy manner (IDF, 2006b).

DOI identifiers are made up of a prefix and a suffix separated by a slash. Its syntax is: < DIR > . < REG > / < DSS > , where < DIR > . < REG > makes up the prefix and < DSS > the suffix. < DIR > in the prefix denotes Directory Code; it is a required value. In the DOI's version of the Handle system, the only valid value for

< DIR > is 10. (10 is a reserved code for the DOI system.) < REG > denotes Registrant's Code; it is a required value assigned by IDF to the registrant. There is a "." character separating < DIR > and < REG > . < DSS > in the suffix is a required value. It is a code freely assigned by registrants to reference their digital objects.

IDF encourages organisations to register multiple DOI prefixes for the purposes of distributed management. In addition, because DOI suffixes are to be freely assigned by the registrant organisation, the DOI system allows existing identifying conventions to be incorporated into its own system. In addition the metadata system in DOI can also be updated to accommodate other conventions.

Description

Description deals with metadata which provides descriptive annotations for corresponding digital objects. Since DOI is considered an opaque string – nothing can be inferred from the number with respect to its use in the DOI system, the use of metadata is necessary for describing the object in question. In the DOI system the DOI identifier itself is recorded in metadata.

The description module in the DOI system is based on the < indecs > framework which requires that all DOI registered objects be described by a well-formed metadata scheme. This does not mean that all registered DOI objects should use the same metadata format, but that they are based on the same kernel of metadata. Additional metadata elements needed for a particular transaction are built on top of this kernel. Failure to use the common kernel can result in the appearance of gaps when multiple metadata formats are involved (IDF, 2006c).

In light of the preceding issue DOI requires that registrants must specify the metadata kernel for every DOI item they register. Intellectual properties of different types or classes require different metadata elements. For example, articles, music recordings and photo images require different descriptions and therefore different metadata elements. The DOI system suggests that registrants (or resource holders) use the DOI Application Profile (DOI-AP) to describe their objects. (DOI-AP was known as Genre in its earlier incarnation.) Each DOI is assigned to at least one DOI-AP. Base-AP is the most common example of a DOI-AP: it encapsulates the kernel metadata as mentioned above.

In the early days of DOI development it was not compulsory to indicate a metadata format to which a DOI object is assigned. To accommodate objects having been created under these conditions, a Zero-AP was created as their metadata format. Subsequently, additional APs, as appropriate to their context, can be assigned to them to enable added features.

Resolution

DOIs are persistent identifiers linked to digital objects. By comparison URLs are identifiers representing addresses of internet resources. Although objects which DOIs identify can sometimes be internet addresses, just like URLs, DOIs and URLs are quite different. DOIs identify all sorts of digital objects, while URLs identify only internet addresses. DOI resolution systems are built to resolve identifiers to their associated objects based on the application context. DOI is said to be actionable precisely because of the resolution feature. At present DOI uses the Handle system to provide resolution services. The Handle system works as follows: given a document object (e.g. the PDF

version of the DOI Handbook) which one hopes to locate through DOI, there are a couple of options one can consider:

- (1) install a Handle system browser plug-in (CNRI Handle System Resolver) and type doi:10.1000/182 in the browser to bring up the document in question;
- (2) or type <http://dx.doi.org/10.1000/182> into the browser.

The latter requires a server behind <http://dx.doi.org> to provide an HTTP Handle resolution service to return the corresponding digital object, which in this case is the URL <http://dx.doi.org/hb.html>. In the event that this handbook has been moved, the resource holder needs only to update the registration server to reflect the new URL address. The broken link problem as such is eliminated. This is effectively what the Handle system does in providing resolution services.

It is the responsibility of the DOI registrant to maintain the correct links in the DOI system. The types of documents the Handle system is able to support can be extended to include all kinds of internet documents such as Java applets CGI scripts and other dynamic web documents.

Policy

The DOI system is governed by a set of policies. These policies cover the development, cooperation and integration of the three function modules: enumeration, description, and resolution. The staff of IDF are responsible for setting policy for the entire DOI system. Registration responsibilities are delegated to the Registration Agency Working Group (RAWG). Resolution services are delegated to the Handle system. The policy body ensures that the DOI system develops into a rich and complete system (IDF, 2006d).

< LIPS-DOI > system

Given that the cost for international participation in the DOI system is too high for many organisations in Taiwan to consider, an initiative was set up in 2004 to develop an alternative system called < LIPS-DOI > , which allows for an affordable adoption of DOI in Taiwan. The basic tenets behind the establishment of < LIPS-DOI > are low cost and compatibility with international DOI, although there will be differences in certain encoding details and manners of operation. In the first version of < LIPS-DOI > application-level searching and storage functions were implemented using built-in features of database management systems (Yeh, 2003; Yeh and Chen, 2004). The results produced less than desirable performance in the search function. A decision was made to re-implement the system using a full-text search technology and Java in order to improve search performance and promote platform compatibility. The following describes the < LIPS-DOI > system covering individual components and their interfaces.

System components

Enumeration module. Although the prefix and suffix components of DOI enumeration are considered opaque strings, we can nevertheless derive from them information about the registrants and their registered objects. In other words DOI strings use the prefix to denote the registrant, and the suffix to denote an identifier as defined by the registrant. The design of the < LIPS-DOI > system follows the same principle that

the registrant can freely define the suffix, but at the same time the registrant can choose not to explicitly define it and allow the system to place an automatic time string in its place. < LIPS-DOI > syntax imposes that the time string be a required value positioned between the prefix and the suffix. The suffix argument is optional. The prefix, time string and suffix are separated by “-” (unlike “/” as in DOI), to make provision for a possible future merge with DOI. In other words, if in the future < LIPS-DOI > becomes an official DOI registrant agent, all its derivative identifiers can very easily be upgraded into DOI identifiers. For example, when using the < LIPS-DOI > system’s search service, object identifiers will have this structure: < Registrant > - < RegistrationTime > - < LocalName > . When using < DOI > , the same object identifiers are now < DIR > . < REG > / < Registrant > - < RegistrationTime > - < LocalName > . < REG > will denote the < LIPS-DOI > system, as officially registered under the international DOI registration body. The rationale behind embedding time string in < LIPS-DOI > is as follows:

- Timestamp records the act of registering an intellectual property. The timestamp format is year, month, day, hour, minute, and second in sequence (or YYYYMMDDhhmmss). The length is 14 characters, which is relatively long but necessary to precisely describe the time of an object’s registration.
- Users who do not have an organisation-wide standard numbering convention can rely on < LIPS-DOI > to auto-assign unique identifiers.
- One can tell which objects are registered at the same time by their identifiers.
- System-assigned time strings are opaque strings in keeping with < LIPS-DOI >’s characteristics. Users can also decide whether they want to explicitly assign suffixes in batch or single assignment as they see fit.

The enumeration module in the < LIPS-DOI > system uses standard time protocols to acquire globally standard time. The two commonly used protocols are NTP (Network Time Protocol, RFC-1305) and SNTP (Simple Network Time Protocol, RFC-2030). The use of a time protocol allows the < LIPS-DOI > system to be compatible in case of future integration with a global network. This is part of its long-term strategy. Currently, the clock in the < LIPS-DOI > server is synchronised with the NTPClock software version 1.2.1 (NSTFL, n.d.).

Description module. < LIPS-DOI > was planned from the early stages to handle various digital object types in the same manner as DOI. During the early course of its development the description module would first implement a set of the basic metadata elements, which were grouped together into Kernel AP. Later, additional application-specific metadata elements were created and grouped into corresponding Application Profiles. Relationships or mappings between different metadata elements were also taken into account in the description module. This mapping is the so-called Metadata Crosswalk (St. Pierre and LaPlant, 1998).

In the < LIPS-DOI > system the kernel metadata are based on the 15 elements (or vocabularies) of the Simple DC, instead of the elements in Kernel AP as defined by DOI. This gives one the advantages of having a more descriptive vocabulary set and a model that has been more widely adopted by the international community. In fact DOI itself could have been based on DC as its kernel metadata format. However, it chose

Kernel AP, which imposes stricter rules on the resources being described and provides basic mapping functions for any pair of metadata elements (IDF, 2006e).

The < LIPS-DOI > and DOI systems both take the approach of using a flat data structure – XML files – to describe resources. The < LIPS-DOI > system uses the XML schema from Simple DC to verify the syntax of the metadata enclosed in an XML document (Van der Vlist, 2000).

Resolution module. In the DOI framework, when the user enters a DOI string into the browser, the Handle system activates and sets the resolution. In order for this to work, either a Handle plug-in must be installed in the client's browser, or one may use a proxy server which knows the underlying protocol and thus can handle the resolution on behalf of the browser. A Handle system consists of many service sites. Each service contains at least one primary site and multiple secondary sites. Within each site there are multiple Handle servers. Globally, there is the Global Handle Registry Service which oversees the location and namespace registrations of all the regional public local services. Some resolutions require cross-regional hops. Such hops will go through the Global Handle Registry Service before arriving at another public local service. Therefore, one can enter into the Handle system from any point in a global network and be assured that the resolution will be carried out either locally or remotely. Each Handle can be linked to one or many types of resources. That is, each DOI can be resolved to one or more instances, such as one or more URLs. This is the concept of one-to-many resolution. The < LIPS-DOI > system looks to the DOI system as a reference. It first implements the one-to-one resolution function, then one-to-many. In terms of deployment the < LIPS-DOI > system is not as complicated as the Handle system: each service involves the work of one server, and the resolution of a < LIPS-DOI > identifier is processed by a proxy server.

System interface

Management interface. The management interface of the < LIPS-DOI > system prompts the user to enter the prefix, username and password to login. Management functions include: creating a new < LIPS-DOI > identifier, editing an already registered digital object, deleting a digital object and creating/deleting/editing digital objects in batch mode. Once the user has logged in by successful verification of the prefix, username and password, the system displays a list of digital objects the user owns. Beside them the functions to add, edit and delete a digital object are provided. The metadata concerning user registration information are stored in a database, while the metadata for the digital object are stored in a flat file.

Search interface. According to the DOI system policy, the Kernel AP (kernel metadata) of a digital object should administer public access – available for public search. Accordingly, the < LIPS-DOI > system provides an open search interface for all to access, and furthermore a rich interface to enable more effective search results. < LIPS-DOI > can use XSL to transform XML into a user preferred format. Since metadata in the < LIPS-DOI > system are expressed in XML, the task of displaying them is quite straightforward. One can simply use an XSLT to allow the XML document to show up as HTML in a browser. Presently, the presentation of metadata is an HTML table of 15 items with their corresponding values. One can also use XSL to display certain key metadata elements such as title, creator, etc. alongside a search result.

Data maintenance. The < LIPS-DOI > system needs to solve the following two issues in data maintenance:

- (1) input/output of data; and
- (2) batch processing of data.

In terms of data output, since metadata are already in the format of XML, it is by definition structured data and therefore can directly output the data as XML. In terms of data input and updating a metadata value, the < LIPS-DOI > system will parse the incoming XML-formatted data to extract the < LIPS-DOI > identifier and use that to update the value of the metadata element in question.

In terms of batch processing, the key tasks are building data stores in batch and backing up. Data stores are built usually from existing user data coming from spreadsheets and database programs. These existing data will be formatted into XML to match the format of Kernel AP of Simple DC. The XML will contain an identifier element which is assigned a value equal to the old data's primary key, ISBN, ISSN or a newly created serial number. The < LIPS-DOI > system then parses out the identifier from the XML data and pre-appends a prefix and timestamp to the front of the current value and then stores the whole string in the description module.

< LIPS-DOI > applications

Although most applications of the < LIPS-DOI > system have focused on Chinese-language resources, the system is capable of integrating disparate digital objects and documents in multiple languages. The < LIPS-DOI > system is located at <http://lipsdoi.lis.ntu.edu.tw/> (or <http://doi.lips.tw/>). The web site display is shown in Figure 3. The default language is set to Chinese. If one wishes to see it in English, click the link on top labelled “English”. The following descriptions are based on the English display, but the resources in < LIPS-DOI > may be in Chinese.

The screenshot shows the web interface of the < LIPS-DOI > system. At the top, there is a navigation bar with the system name and a 'login' button. Below the navigation bar, there is a sidebar with a menu containing 'Introduction', 'News', 'Resource', and 'Contact'. The main content area contains the following text:

<LIPS-DOI> is constructed by [Language & Information Processing System \(LIPS\)](#) Lab under the direction of Prof. Kuang-hua Chen. The current system is version 2.0.

The basic tenets behind the establishment of <LIPS-DOI> are low-cost and compatibility with international DOI - although there will be differences in certain encoding details and manners of operation.

The first version of <LIPS-DOI> system is implemented by Kuan-Chih Yeh and [Kuang-hua Chen](#). Due to inefficiency of search module, we apply full-text search technique to re-implement <LIPS-DOI>. In addition, we use Java to code all programs for system compatibility. The second version is implemented by Kuang-hua Chen and Yu-ting Chiang.

Currently, the registered resources in <LIPS-DOI> are shown as follows.

- National Taiwan University Library:
<http://www.lib.ntu.edu.tw/pub/univ1/menu.html>
- The resources of LIS Library of National Taiwan University
- The registered resources by third parties or persons.
- A few online resources like online free journals, papers, etc.

A callout box with a speech bubble points to a link labeled 'Switch to Chinese Version'.

Figure 3.
< LIPS-DOI > system
display

The “Introduction” link on the sidebar provides a brief history of how the < LIPS-DOI > system started and discusses the capacity ranges for resource integration. The “Reference Resources” link provides topics related to current studies. The “Metadata Search” link provides a basic and advanced search to allow users to find what digital objects they have registered in the system, as shown in Figure 4.

The “Element Search” link allows the user to search the < LIPS-DOI > system based on the 15 elements of Simple DC. The user can key partial or complete strings into the search prompt to begin searching. For example, the user can enter the keyword “library” to retrieve an enumeration of all metadata belonging to all registered objects. “Basic search” allows users to run a search against all element names. Figure 5 shows a search result example. The user can click on an identifier to link to a more detailed web page. For example, in Figure 5, when the user clicks on the first item in the table, the new page appears as shown in Figure 6.

Figure 6 shows a screenshot of NTU Library’s WebPAC system displaying an item previously selected from the < LIPS-DOI > search interface. This is a demonstration of the < LIPS-DOI > system’s integrated search feature. One can register the metadata of a digital object, be it a physical item such as a book or video cassette, from a different system into the < LIPS-DOI > system and thereafter use the < LIPS-DOI > system’s search feature to locate that item and be redirected back to its native interface.

The web page’s upper right-hand corner provides a link for the user to login to the < LIPS-DOI > system. When the user clicks on the “login” link, the login screen will pop up. Users have to enter account, password and prefix. Once logged in, the authorised user can perform various system administration tasks such as importing a new metadata format, registering a new object, importing a group of objects in batch, registering a single object or managing rights (user or group rights). Once a registered

The screenshot displays the LIPS-DOI system interface. At the top, the header includes the LIPS-DOI logo, the text “國立臺灣大學圖書館學系數位物件辨識碼系統” (National Taiwan University Library School of Library Studies Digital Object Identifier System), and a “login” button. Below the header, the main content area is divided into a sidebar on the left and a main search area on the right. The sidebar contains links for “Introduction”, “News”, “Resource”, “Contact”, “Search”, “Simple Search”, “Advanced Search”, “Prefix Registration”, and “DOI Naming Proxy”. The main search area features a “Basic Search” section with a text input field and a “Search” button. Below this is an “Element Search” section with a “Boolean” dropdown menu set to “AND”, and a “Search” button. A list of 15 DC elements is provided for selection: title, creator, subject, description, publisher, contributor, date, type, format, identifier, source, language, relation, coverage, and rights. A callout box labeled “DC elements” points to this list. At the bottom of the page, a footer reads “2005 © 國立臺灣大學圖書館學系 版權所有 LIS, NTU”.

Figure 4.
Simple Search provides
both basic search and
simple DC element search



Figure 5. Simple Search results



Figure 6. Redirect to one search result

user has logged into the system, one will see a more selective sidebar on the left as shown in Figure 7. This indicates that a registered user has additional rights to manage resources on top of the basic rights to search.

When clicking on “DTD Management”, one enters a screen as shown in Figure 8. This screen displays all known metadata formats in the system. The user can create a new metadata format or update/edit or delete an existing format. As Figure 8 shows, the currently logged-in user is khchen, who has registered many metadata formats –Discovery of Tamsui River, Lahodobo, Ming and Qing Archives, etc. The screenshot indicates the first format, Dublin Core, which is a default item. This is the so-called Kernel AP or Base AP.



Figure 7.
First screen after a registered user has logged in



Figure 8.
Metadata format management

To import a new object into the system, one clicks “Metadata Management” and proceeds to fill out the new object’s metadata elements. As Figure 9 shows, one can choose either to add a new item or batch import. Figure 10 shows “Add New” item. This example shows a Dublin Core metadata format and its constituent elements as input fields. After the user has filled out the input fields and clicked “OK” the new object will be registered into the system.

If the user chooses batch import from Figure 9, the following screen as shown in Figure 11 appears. On the screen the user must select the metadata format (in this case Dublin Core), enter the location of the import file and click “OK” to begin importing. The file must be a well-formed XML file. Figure 12 shows an example of such a file. To save download time on a large XML file, the user should zip the file first before importing. The system will unzip files accordingly.

Under “Metadata Maintenance” the registered user can also use the search function to bring up an existing object for maintenance. The user can run a search for an existing object based on the < LIPS-DOI > identifier, the metadata element or resolution data to locate then bring up the object for updating. The user can also locate an existing object from a list of all registered items. The newly updated object will keep the persistent < LIPS-DOI > identifier and propagate the respective changes immediately. In other words, if the location of an object has been changed, the user would not know the difference and would still be able to locate that object based on the same identifier.

The “DOI Naming Proxy” link on the left sidebar links the user to < LIPS-DOI > ’s proxy resolution service. The < LIPS-DOI > system provides a built-in proxy resolution service. Going forward, the resolution server will be made stand-alone. The user can enter a < LIPS-DOI > identifier into the web page located at <http://lipsdoi.lis>.

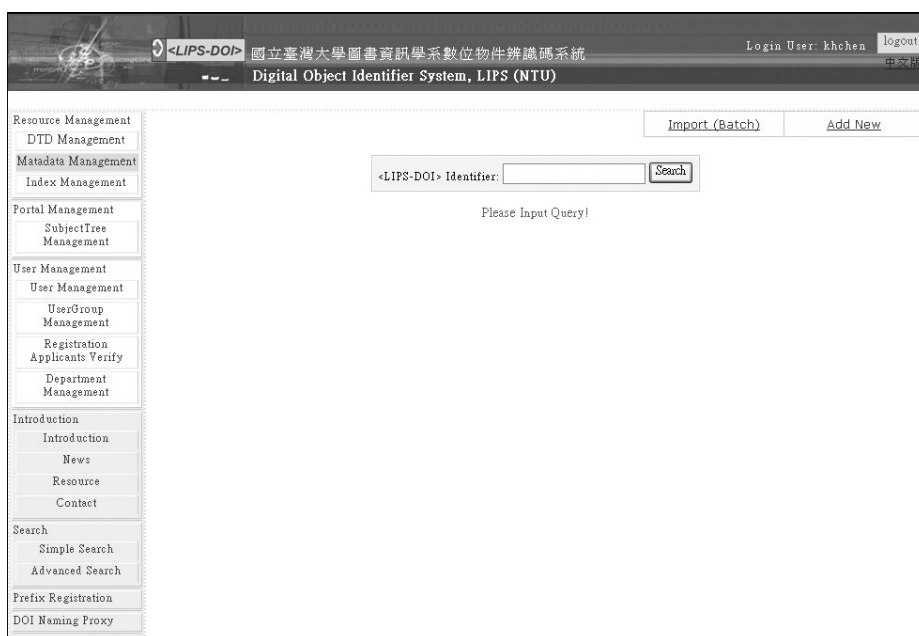


Figure 9.
Metadata management

Figure 10.
Registered user
performing a single import

The screenshot shows the 'Add New Metadata' form in the LIPS (NTU) system. The interface includes a top navigation bar with the system name and user information. A left sidebar contains a menu with categories like Resource Management, Portal Management, User Management, Introduction, Search, and DOI Naming Proxy. The main content area is titled 'Add New Metadata' and contains the following fields:

Department	NTU Library
DTD	DublinCore
Prefix	0100001
Metadata	<input type="text"/> title <input type="text"/> creator <input type="text"/> subject <input type="text"/> description <input type="text"/> publisher <input type="text"/> contributor <input type="text"/> date <input type="text"/> type <input type="text"/> format <input type="text"/> identifier <input type="text"/> source <input type="text"/> language <input type="text"/> relation <input type="text"/> coverage

Figure 11.
Registered user
performing a batch import

The screenshot shows the 'Metadata Import' form in the LIPS (NTU) system. The interface is similar to Figure 10, with the same top navigation bar and left sidebar. The main content area is titled 'Metadata Import' and contains the following fields:

Department	NTU Library
DTD	DublinCore
Prefix	0001001
File Import	<input type="text"/> <input type="button" value="瀏覽..."/>
Identifier Generation	<input type="checkbox"/> Keep original filename (suffix) <ul style="list-style-type: none">• Please provide well-form xml document.• Please package multi files by zip file to upload.
<input type="button" value="Confirm"/> <input type="button" value="Cancel"/>	

```
<?xml version="1.0" encoding="UTF-8" ?>
- <dataroot xmlns:od="urn:schemas-microsoft-com:officedata">
- <lipsdoi-dccore>
  <title>市場調查完全手冊</title>
  <creator>赫格、傑克生</creator>
  <publisher>小知堂</publisher>
  <date>2000[民89]</date>
  <type>text</type>
  <format>book</format>
  <identifier>957-03-4975-1</identifier>
  <language>zh</language>
  <rights>國立台灣大學圖書館</rights>
  <resolution>http://140.112.113.1/search*chi/i?SEARCH=957-03-4975-1</resolution>
</lipsdoi-dccore>
- <lipsdoi-dccore>
  <title>執行力組織</title>
  <subject>csh 組織(管理)、csh 決策管理</subject>
  <publisher>麥格羅希爾</publisher>
  <date>2003[民92]</date>
  <type>text</type>
  <format>book</format>
  <identifier>957-493-728-3</identifier>
  <language>zh</language>
  <rights>國立台灣大學圖書館</rights>
  <resolution>http://140.112.113.1/search*chi/i?SEARCH=957-493-728-3</resolution>
</lipsdoi-dccore>
</dataroot>
```

Figure 12.
Sample XML import file
for batch import

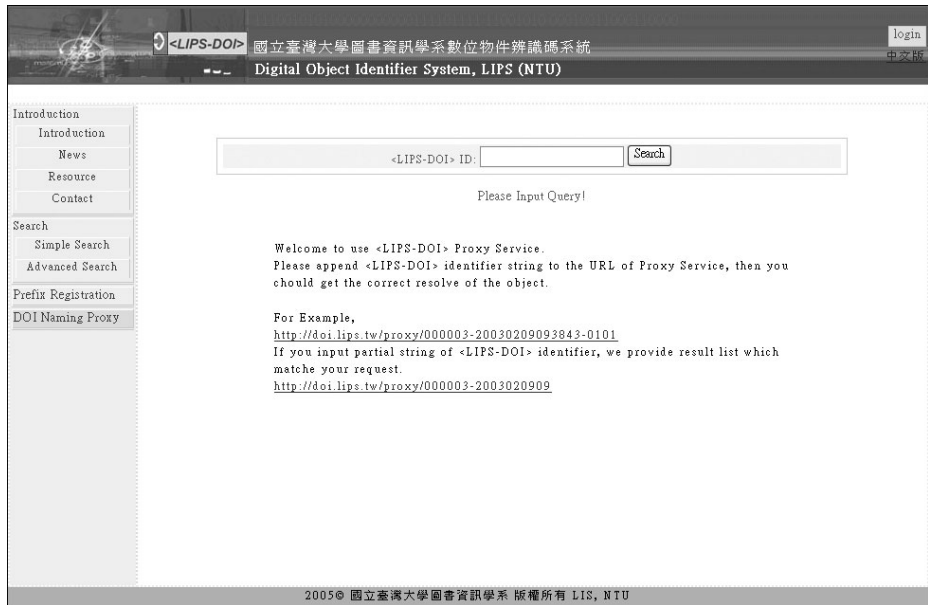
ntu.edu.tw/proxy/ to retrieve its corresponding resolution. Figure 13 shows a screenshot of this web page. Currently the < LIPS-DOI > system is capable of one-to-one resolution. In this case, when the user enters a < LIPS-DOI > identifier into proxy's web page, he will be automatically redirected to the corresponding object.

As these screenshots demonstrate, < LIPS-DOI > can be applied as a practical system of integrated search and management for digital and physical resources. Compared with the DOI system, < LIPS-DOI > provides authorised users an easy and lost-cost solution to attain the management of variant resources and give end-users the ability to search various resources at the same time without worrying about where these resources are deposited. In addition, although < LIPS-DOI > may be implemented in different technologies and be operated on different platforms, < LIPS-DOI > presents the flexibility to work or integrate with < DOI > , provided that < LIPS-DOI > becomes a registered member of CrossRef.

Conclusion

The downside of the DOI system for many users in Taiwan is that it is not a free service. Consequently, the < LIPS-DOI > system was designed from the beginning as a system that targets public users, with initial applications focused on Chinese resources. In contrast to the DOI system it targets mostly large electronic publishers. Nevertheless, the DOI system has gained momentum in becoming an international standard. In light of this the < LIPS-DOI > system was designed with a view to integrating with DOI in the future. What this means is that, given the availability of certain indicators (e.g. the quantity of resources registered under the < LIPS-DOI > system reaching certain critical levels or the < LIPS-DOI > system becoming a major

Figure 13.
< LIPS-DOI > system's
proxy service



registration authority in Taiwan), one could then nominate that < LIPS-DOI > be merged with the international community. At that point it is a simple act of appending existing < LIPS-DOI > identifiers to the end of the newly DOI registered prefix to produce DOI identifiers.

The < LIPS-DOI > system has now evolved into Version 2.0 after having gone through many stages of testing and upgrading. It is currently able to conduct a full-text search against registered resources, and, looking ahead, further improvements are expected in the area of search functionality. The integrated resources in the system are currently dominated by Chinese resources such as the electronic version of the *University Library Journal* and NTU LIS Library's books and periodicals. In addition a few English-language public resources such as *D-Lib Magazine* are also searchable through < LIPS-DOI >. This system provides a platform for resource integration through metadata description and search integration through a single interface. The < LIPS-DOI > system is a case in point for a successful working system which provides an integrated search and management environment for varied digital and physical resources across diverse locations.

References

- CrossRef (2006), *CrossRef Newsletter*, June, available at: www.crossref.org/01company/10newsletter.html
- IDF (2006a), *Director's Message*, International DOI Foundation (IDF), Oxford, available at: www.doi.org/welcome.html
- IDF (2006b), "Numbering", *DOI Handbook Version 4.2.0*, International DOI Foundation (IDF), Oxford, available at: www.doi.org/handbook_2000/enumeration.html

-
- IDF (2006c), "DOI data model", *DOI Handbook Version 4.2.0*, International DOI Foundation (IDF), Oxford, available at: www.doi.org/handbook_2000/metadata.html
- IDF (2006d), "Policy", *DOI Handbook Version 4.2.0*, International DOI Foundation (IDF), Oxford, available at: www.doi.org/handbook_2000/policies.html
- IDF (2006e), "DOI Resource Metadata Declaration", *DOI Handbook Version 4.2.0*, International DOI Foundation (IDF), Oxford, available at: www.doi.org/handbook_2000/appendix_5.html
- Lynch, C.A. (1997), "The Z39.50 Information Retrieval Standard, Part I: a strategic view of its past, present and future", *D-Lib Magazine*, Vol. 3 No. 4, available at: www.dlib.org/dlib/april97/04lynch.html
- Marshall, C.C. (1997), "Annotation: from paper books to the digital library", *Proceedings of DL97, Philadelphia, PA*, pp. 131-140.
- NSTFL (n.d.), *Nation Standard Time and Frequency Laboratory*, available at: www.stdtime.gov.tw/
- Paskin, N. (2005), "Digital object identifiers for scientific data", *Data Science Journal*, Vol. 4, pp. 12-20.
- St. Pierre, M. and LaPlant, W. Jr (1998), *Issues in Crosswalking Content Metadata Standards*, National Information Standards Organization, available at: www.niso.org/press/whitepapers/crsswalk.html
- Van de Sompel, H. and Hochstenbach, P. (1999a), "Reference linking in a hybrid library environment, part 1: frameworks for linking", *D-Lib Magazine*, Vol. 5 No. 4, available at: www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt1.html
- Van de Sompel, H. and Hochstenbach, P. (1999b), "Reference linking in a hybrid library environment, part 2: SFX, a generic linking solution", *D-Lib Magazine*, Vol. 5 No. 4, available at: www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt2.html
- Van de Sompel, H. and Hochstenbach, P. (1999c), "Reference linking in a hybrid library environment, part 3: generalizing the SFX solution in the SFX@Ghent & SFX@LANL experiment", *D-Lib Magazine*, Vol. 5 No. 10, available at: www.dlib.org/dlib/october99/van_de_sompel/
- Van de Sompel, H. and Lagoze, C. (2000), "The Santa Fe Convention of the Open Archives Initiative", *D-Lib Magazine*, Vol. 6 No. 2, available at: www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html
- Van de Sompel, H. and Lagoze, C. (2004), *The Open Archives Initiative Protocol for Metadata Harvesting, Protocol Version 2.0*, available at: www.openarchives.org/OAI/openarchivesprotocol.htm
- Van der Vlist, E. (2000), *W3C XML Schema Structures Reference*, available at: www.xml.com/pub/a/2000/11/29/schemas/structuresref.html
- Yeh, K.-C. (2003), "The application of digital object identifier on Chinese resources", Master's Thesis, National Taiwan University, Taipei (In Chinese).
- Yeh, K.-C. and Chen, K.-H. (2004), "The implementation of digital object identifier systems", *University Library Journal*, Vol. 8 No. 1, pp. 107-29 (in Chinese).

Corresponding author

Kuang-hua Chen can be contacted at: khchen@ntu.edu.tw