DRTC Workshop on
Semantic Web
8th – 10th December, 2003
DRTC, Bangalore

Paper: I

Identification Schemes for Digital Resources

Dimple Patel
Documentation Research and Training Centre
Indian Statistical Institute
Bangalore-560 059
dimple@drtc.isibang.ac.in

Abstract

This paper discusses the various naming and addressing systems used to identify and locate resources in the digital environment. There are various schemes that have been developed for this purpose, like, URL, URN, URC schemes developed by the IETF (Internet Engineering Task Force), PURL developed at OCLC. The publishing industry also has developed the Digital Object Identifier (DOI), which is being used for rights management of intellectual property. The specifications and the working of URLs, URNs, PURLs, Handles and DOIs are discussed in detail in this paper.

1. Introduction

Identification of a resource is the first step to gain access to any resource. In the print world legacy systems like the International Standard Book Number (ISBN), the International Standard Serial Number (ISSN), and the Universal Product Code (UPC) are being used for past many years. Similar identification systems are also required for electronic resources. Especially, on the Internet where resources are available in various formats. To this end the IETF (Internet Engineering Task Force) - an open international community that manages the standards development for the Internet has developed various schemes such as the URI, URL, URN, etc. Notable initiatives taken up by other organizations are OCLC's PURL System, CNRI's Handle System, the Digital Object Identifier (DOI) initiative, and the CrossRef system (which uses DOIs). This tutorial deals with these identification systems for electronic resources.

2. URI (Uniform Resource Identifier)

"A Uniform Resource Identifier (URI) is a compact string of characters for identifying an abstract or physical resource." (1)

The URI (Uniform Resource Identifier) specifies a generic syntax. It is a generic set of schemes that identify any document/resource on the Internet. The schemes like URL, URN, etc form the subset of the URI scheme. The specifications given for URIs are applicable to all the subsets under it. The generic syntax for a URI scheme given in the specification RFC2396 (*URI Generic Syntax*) is as follows:

<scheme>:<scheme-specific-part>

where, scheme can be any protocol that is used to access the resource like http, ftp, mailto etc. And the scheme-specific-part is rendered according to the conventions specified by the given scheme. The following are the examples of URI:

http://www.isibang.ac.in/DRTC/srr/index.htm (URL)
ftp://ftp.rfc-editor.org/in-notes/rfc2034.txt (FTP)

URI is characterized by the following definitions (1):

Uniform

Uniformity provides several benefits: it allows different types of resource identifiers to be used in the same context, even when the mechanisms used to access those resources may differ; it allows uniform semantic interpretation of common syntactic conventions across different types of resource identifiers; it allows introduction of new types of resource identifiers without interfering with the way that existing identifiers are used; and, it allows the identifiers to be reused in many different contexts, thus permitting new applications or protocols to leverage a pre-existing, large, and widely-used set of resource identifiers.

Resource

A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Bangalore"), and a

collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources. The resource is the conceptual mapping to an entity or set of entities, not necessarily the entity which corresponds to that mapping at any particular instance in time. Thus, a resource can remain constant even when its content - the entities to which it currently corresponds - changes over time, provided that the conceptual mapping is not changed in the process.

Identifier
An identifier is an object that can act as a reference to something that has identity. In the case of URI, the object is a sequence of characters with a restricted syntax.

Having identified a resource, a system may perform a variety of operations on the resource, such as *'access', 'update', 'replace',* or *'find attributes'*.

3. URL (Uniform Resource Locator)

The most familiar scheme is the URL (Uniform Resource Locator). URL (Uniform Resource Locator) is the address of the file accessible on the Internet. As its name suggests, it helps us in physically locating resources on the Internet. It is an abstract identification given to a resource available on the Internet, accessible through various schemes and protocols like *ftp (file transfer protocol), http (hypertext transfer protocol),* etc. The general syntax and the use of URLs is described in the RFC1738 (*Uniform Resource Locators*). (2)

3.1. Components of a URL

The syntax for a URL is defined as follows:

<scheme>:<scheme-specific-part>

it consists of two main parts, the scheme being used, followed by a 'colon' then a string of characters (scheme-specific-part) which can be interpreted depending on the scheme being used. The second part is divided into two parts i.e. the domain name specifying the specific computer on the Internet and a pathname (hierarchical description of the file i.e. directory, filename) where the resource is stored on the computer. Optionally, it may also contain the port number.

The different schemes recognized are (3):

ftp	File Transfer protocol
http	Hypertext Transfer Protocol
gopher	The Gopher protocol
mailto	Electronic mail address
news	USENET news
nntp	USENET news using NNTP access
telnet	Reference to interactive sessions
wais	Wide Area Information Servers
file	Host-specific file names
prospero	Prospero Directory Service

New schemes can be added by future specifications.

3.2. Examples of URLs

http://www.isibang.ac.in/DRTC/srr/index.htm

Here, *http* (Hypertext Transfer Protocol) is the protocol being used to access the resource; *www.isibang.ac.in* is the name of the computer where the resource is located; *DRTC/srr/index.htm* is the pathname to locate the specific file

ftp://ftp.rfc-editor.org/in-notes/rfc2034.txt

Here, *ftp* (File Transfer Protocol) is the protocol of Internet used for transferring files between remote systems. This is a protocol being used in the above example to access the resource on the Internet. *ftp.rfc-editor.org* is the computer where the resource is stored**.** *in-notes/rfc2034.txt* is the pathname of the location of the resource.

3.3 Problems with URLs

Though URLs, are meant for locating resources they are also being used to identify resources on Internet. This presents some problems while using URLs. Because the URLs keep changing and result in broken links. The main causes for the broken links are (3):

- the server moves to a new computer
- the server moves to a new port on the same computer
- the name of the computer the server runs on changes
- the resource's name changes
- the resource moves to a new server

Some other problems identified are (4):

- long URLs are difficult to type;
- changing hostnames, port numbers, directories, filenames; and

4. URN (Uniform Resource Name)

To resolve the problem of ending up with invalid URLs the IETF formed the URN Working Group. URNs are still in the developing stage. They are intended to be more persistent and unique identifiers than URLs. The URNs are meant only to identify the resources and *not* to specify the location. The URN Working Group came up with the requirements for URNs.

4.1. Functional Capabilities of URN

A URN should have the following functional capabilities(5):

- **Global:** A URN should be able to reference a resource globally irrespective of its location.
- **Unique:** The same URN will never be assigned to two different resources.

- **Persistence:** It is intended that the lifetime of a URN be permanent or atleast over a long period of time.
- **Scalability:** URNs can be assigned to any resource available on the Internet.
- **Legacy support:** The scheme must permit the support of existing legacy naming systems (like, ISBN), as long as they satisfy the other requirements described in the URN specification.
- **Extensibility:** Any scheme for URNs must permit future extensions to the scheme.
- **Resolution:** A URN should have a feasible mechanism to translate a URN to a URL, for URNs that have corresponding URLs.

4.2. Syntax of URNs

The fundamental requirement for a URN was that unlike the URL, its syntax should be independent of technicalities and also it should support the existing legacy systems, for instance the International Standard Book Number (ISBN), International Standard Serial Number (ISSN), etc. The specifications for the syntax of URNs are given in RFC 2141 (*URN Syntax*)

Syntax of URN

URN:*NID:NSS*

where NID is the *Namespace Identifier* and NSS is the *Namespace Specific String*. The *NID* can be any of the currently existing naming schemes. Whereas, the *NSS* is dependent on the rules of NID.

4.3. Examples of URNs

urn:isbn:123456789X

This example URN shows support for the ISBN scheme used by the publishing industry.

urn:inet:drtc.isibang.ac.in

This example is used by Internet servers. The NSS in this case being a hostname and a string to resolve at that host.

urn:bsnl:910808483975

This URN shows a (hypothetical) example of a national telecommunications carrier and a particular individual's telephone number.

4.4. URN Usage

Several European national libraries use NBN (National Bibliography Number)-based URNs for identification of electronic resources in their digital archives. One such example is the E-depot system, DIAS (Digital Information Archiving System), recently implemented (December 2002) in the Dutch national library. Their URNs look like this (6)

URN:NBN:nl:kb:eDepot-103942842457

This syntax, using NBN, is as defined in the RFC3188 (*Using National Bibliography Numbers as Uniform Resource Names*). Accordingly, the namespace specific string (in the above example, *eDepot-1039428424571*) can be basically anything as long as the string is unique; the DIAS uses UNIX time of the moment the URN was generated

In Sweden and Finland the generation of URN started in 1998, with the commencement of the Nordic Metadata Project. These libraries use a software called *URN generator*. It was developed in Perl language and hence is portable on different platforms. It is available for use free of cost. (7) It generates sequential numeric identifiers. For example:

URN:NBN:fi-fe977564
URN:NBN:fi-fe977565

These URNs cannot be used 'as is' i.e. they cannot be typed in the Address box of a Web browser, like URLs, because an international URN resolution infrastructure is not available yet. These URNs only identify the digital documents in the repository. Therefore, URNs are embedded in the META tag of the HTML file of the resource. The Identifier element of the Dublin Core Metadata Set can be used for this purpose. The general syntax would be:

<META NAME="DC.Identifier" SCHEME="URN" CONTENT="URN code">

Example:

<META NAME="DC.Identifier" SCHEME="URN" CONTENT="URN:NBN:fi-fe977564">

This ensures that the search services will be able to retrieve the document.

4.5. ISSN-URN Portal

The International Standard Serial Number is a widely recognized identification system for uniquely identifying serials. ISSN is being used by publishers, librarians, etc for various purposes since many years. E-journals also are identified by ISSNs.

One of the functional requirement of URNs is support for legacy systems already in use like ISBN, ISSN, etc. The ISSN International Centre has undertaken an URN implementation taking advantage of the functional compatibility between the ISSN and the URN. The ISSN can be supported within the Uniform Resource Name (URN) framework as a specific *URN namespace identifier*. In February 2001 the ISSN URN has been registered by IETF and IANA. (8)

"An ISSN URN resolution system using the ISSN identifier as Uniform Resource Name within an ISN URN Namespace has been developed by the ISSN International Centre (ISSN-IC) and is operating as a demonstrator to evaluate all requirements to deploy it in an operational environment." (9)

An ISSN URN resolution system has been developed at the ISSN International Centre, France, which is able to translate a given ISSN-URN into electronic locations :

- location of the bibliographic description or metadata
- location of the periodical (when it is in electronic form).

The following are the different tools developed during this project (8):

A resolution system: a global centralised system located at the ISSN IC. It ensures effective mapping between a given ISSN and one or several URL(s) that implies that *'multiple resolution'*, must be actually performed.

An URL Watcher: is a robot which scans all URLs stored in the ISSN-URN database for checking the persistence and the accessibility to ISSN-URN electronic resources. Broken or redirected URLs are logged and stored to permit efficient resolution.

An ISSN-URN browser plug-in : Existing browsers do not support the URN *'protocol'*. To *'simulate'* such protocol a plug-in for Netscape communicator and MS-Explorer has been developed and is available on the ISSN Website. (10) With this plug-in it is possible to enter directly the ISSN preceded by the string *'urn:ISSN:'* in the browsers' address box instead of typing the complete URL string. As a result, the bibliographic record or/and the online resource are displayed.

5. PURL (Persistent Uniform Resource Locator)

PURL was a project of OCLC. Now the project has ended but the PURL resolution services are being continued. PURLs, are nothing but URLs. A PURL consists of 3 parts (11):

(1) Protocol – the protocol used to access the PURL resolver.
(2) Resolver Address – the IP Address or Domain Name of the resolver.
(3) Name – User Assigned name (here, user means the person who created the PURL)

A PURL looks like this (very similar to a URL as you can see):

http://purl.oclc.org/NET/Ranga

Here, *http* is the hypertext transfer protocol; *purl.oclc.org* is the resolver address; *NET/Ranga* is a user-assigned name

A *'PURL Resolver'* program redirects the PURL to the associated URL. The working of a PURL is very simple. When an end-user types in the PURL in the web browser, it directs to an intermediate resolution service (represented by the resolver address in the above example). The resolution service associates the PURL to the related URL and sends it to the client. At the client end, the rest of the transaction takes place in the normal fashion.

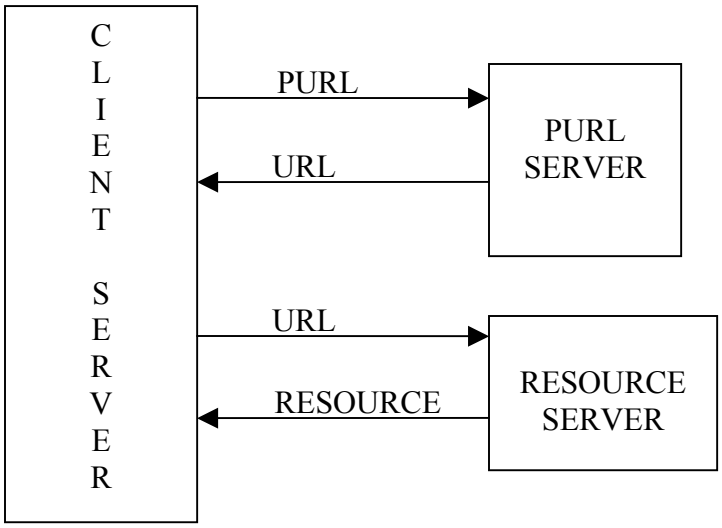

Fig. 1: Purl Resolution Service (11)

PURLs have to be created by the maintainer of the URL using a resolution service. One thing has to be noted that the changes in the URL also are to be entered by the maintainer of the URL. There is no way that the resolution service will automatically come to know of the changes made to the URL. It is entirely the responsibility of the maintainer of the URL to update/modify the PURL as and when required.

6. Handle System

The Handle System was developed at the CNRI (Corporation for National Research Initiatives) as part of the Computer Science Technical Reports (CSTR) project, funded by the Defense Advanced Projects Agency (DARPA). It is a distributed system designed to support global interoperability.

"The Handle System is a comprehensive system for assigning, managing, and resolving persistent identifiers, known as 'handles' for digital objects and other resources on the Internet." (12)

6.1 Handle System Syntax

A handle consists of two parts: its naming authority, also known as *prefix*, and a unique local name under the naming authority, also known as *suffix.* It is rendered as follows (13):

<Handle> ::= <Handle Naming Authority> "/" <Handle Local Name>

The naming authority and local name are separated by "/". A naming authority is a unique string, assigned by CNRI, which identifies a local handle service - LHS (and handles) to the '*Global Handle Registry'*. This will be used as the first part, or *'prefix'*, of each of the handles created under that naming authority. The naming authority string, combined by a slash with another unique string called the *'suffix'*, is called a *'handle'*.

For example,

hdl: 1849/185

The above handle identifies a document on the DRTC Digital Library for Library and Information Science website (*https://drtc.isibang.ac.in*). Here, the local naming authority is DRTC which has been given the unique identifier prefix *1849* by the CNRI Global Handle Registry. *185* is the suffix unique to that particular local naming authority prefix, in this case, *1849*. Together, *hdl: 1849/185* identify a particular digital resource available on the DRTC Website.

6.2 Handles and URNs

The Handle System can be said to be an implementation of URNs. Some of the features of a Handle System that satisfy the functional requirements intended of URNs are: (13):

- **Global:** Every handle is able to reference a resource globally irrespective of its location.
- **Uniqueness:** Every handle is globally unique within the Handle System.
- **Persistence:** Handles may be used as persistent identifiers for Internet resources.
- **Legacy support:** Handles accommodate legacy identifier systems.
- **Scalability:** Handles can be assigned to any resource available on the Internet.
- **Extensible Namespace:** Existing local namespaces may join the handle namespace by acquiring a unique handle naming authority. This allows local namespaces to be introduced into a global context while avoiding conflict with existing namespaces. Use of naming authorities also allows delegation of service, both resolution and administration, to a local handle service.
- **Efficient Resolution Service:** The handle protocol is designed to allow highly efficient name resolution performance.

Other features of Handles are:

- **Multiple Instances:** A single handle can refer to multiple instances of a resource, at different and possibly changing locations in a network.
- **Multiple Attributes:** A single handle can refer to multiple attributes of a resource, including associated services, available through any method at different and possibly changing network locations.

6.3 Users of Handle System

Many organizations have implemented the Handle System for identification of their digital resources. Some of them have implemented the Digital Object Identifier (DOI) which is based on the Handle System technology. (DOI has been dealt in detail in the later section). A list of organizations which have already implemented the Handle System is listed below:

- IDF (International DOI Foundation)
    - CrossRef (scholarly journal consortium)
    - Enpia (Korean content management technology firm)
    - Content Directions, Inc. (US content management technology firm)

- Library of Congress
- DTIC (Defense Technical Information Center)
- Other Interested Parties:
    - cIDF (Content ID Forum, based in Japan)
    - NTIS (National Technical Information Service)
    - NMPA (National Music Publishers' Association)
    - Numerous digital library research projects (eg. DSpace)

6.4 Implementation of Handle System in DSpace

Each site running DSpace needs to obtain a Handle *'prefix'* from CNRI. The Handle System software (written Java) needs to be downloaded from the CNRI website and installed. Then the *'site bundle'* containing the IP address of the website and administrator information is sent to the Global Handle Registry (GHR) administrator, who then allots an identifier for the LHS (Local Handle Service) i.e. the *'prefix'*. In case of the DRTC DL Repository was allotted the identifier *1849*. Then, DSpace has an internal mechanism to allot sequential numeric identifiers to each electronic document added to it. This identifier becomes the *'suffix'*. Hence, any document in the DRTC DL will be identified with handles as:

hdl:1849/158

The GHR stores the identity and location of all local handle services (LHS), and tells a handle client which service to query to resolve a handle. All handle clients (for resolution or administration) know how to contact and query the GHR.

6.5 Resolution of Handles

Handles can be written in two forms:

hdl:1849/158
http://hdl.handle.net/1849/158

The above represent the same Handle. But the current Web browsers are not capable of supporting the first form the handle. The second form of the handle becomes resolvable by the current browsers by using proxy servers. (The proxy servers act as a gateway between the HTTP protocol used by all web browsers and the handle resolution protocol.) CNRI runs a proxy server at *http://hdl.handle.net/* and the Library of Congress runs one at *http://hdl.loc.gov/*. Each proxy server can resolve any handle that corresponds to a URL.

To enable browsers to resolve the first form of Handle CNRI's *Handle Resolver plug-in* needs to be installed (available for download on the Handle System Website). Once this plug-in is installed, the handle (for example, *hdl:1849/158*) can be directly typed in the Location/Address bar of the Web browser. This plug-in works on Windows 95/98/2000/NT/XP Versions. Strangely it does not work on Windows ME version.

7. DOI (Digital Object Identifier)

DOI was a project begun by the American Association of Publishers (AAP) in 1996. Since 1998, the DOI has been managed by the International DOI Foundation. The DOI can be used to identify any of the various physical objects that are "manifestations" of intellectual property for example, printed books, CDs, videotapes, journal articles. A DOI can also be used to identify the digital files in the network environment.

The Digital Object Identifier (DOI) is a system for (14):

- Identifying and exchanging intellectual property in an interoperable digital environment;
- Providing an extensible framework for managing intellectual content in any form at any level of granularity and linking customers with content suppliers;
- Facilitating electronic commerce and;
- Enabling automated copyright management for all types of media.

The DOI was developed by the AAP with two goals in mind (14):

1. Facilitating the creation of an e-commerce market for digital content.
2. Enabling copyright protection and anti-piracy in the digital environment.

Unlike a URL, which points to the *location* of a resource on a computer connected to the Internet, the DOI identifies a piece of content by an identifier that is independent and never changes once it is assigned to the content, very similar to the ISBN. The DOI points to a directory on the Internet, which in turn redirects the user's browser to the current location of the specified content. As long as the URL to which the DOI points is maintained in the central DOI directory, a DOI link survives even when content is moved to a different server or ownership of the work is transferred from one party to another. The underlying technology for this central DOI Directory is the Handle System (discussed in the previous section).

Presently, the central DOI directory is maintained by the Corporation for National Research Initiatives (CNRI). In the future, multiple DOI registration agencies will be established for the purpose distributing and administering DOIs. Presently, there are two registrations agencies certified by the International DOI Foundation (IDF). These are CrossRef and Content Directions.
In addition to its characteristics of persistence, the DOI has a syntax approved by the American National Standards Institute (ANSI) that can facilitate the creation of a unique, unambiguous ID for each element in a work at any level of granularity. For example, the DOI can identify an entire book or magazine, individual chapters or sections, illustrations, or tables. (14)

7.1. The Structure of a DOI

The DOI has two components (14), known as the *'prefix'* and the *'suffix',* separated by a forward slash. The two components together form the DOI.

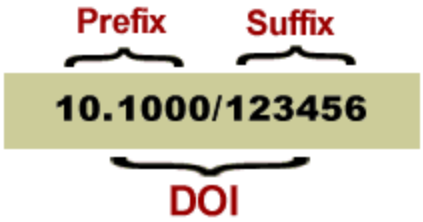

Fig. 2: DOI Structure (14)

There is no technical limitation on the length of either the prefix or the suffix; in theory, at least, there is an infinite number of DOIs available.

7.1.1. The DOI Prefix

The prefix has two components. All DOIs start with "10." This distinguishes a DOI from any other implementation of the Handle System. The next element of the prefix is the number (string) that is assigned to an organization that wishes to register DOIs. The number of DOI prefixes that any organization may choose to apply for is unlimited. For example, a publisher may apply for a single DOI prefix, or for two different ones for journals and books. The prefix is extendible to sub-prefixes. For example:

10.1000.10/123456

The DOI is an opaque string (dumb number). It represents no meaning/information. Because, the DOI remains persistent even though the ownership might change, the prefix remains unaltered once it is registered. In other words it identifies uniquely and persistently the *content only*.

7.1.2. The DOI Suffix

Following the prefix (separated by a forward slash) is a unique suffix (unique to a given prefix) to identify the entity. The combination of a prefix for the registrant and unique suffix provided by the registrant avoids any necessity for the centralized allocation of DOI numbers.

The DOI suffix can be any alphanumeric string that the registrant chooses. This can simply be a sequential number, or it can make use of an existing (legacy) identifier. The latter may often be administratively convenient for the Registrant.

Both of the following would be valid DOIs (14):

10.100X/123456
10.100X/ISBN-900512-44-0

7.2. DOI Resolution

The DOI, uses the Handle System technology for its resolution service. The resolution is from a DOI, e.g., 10.1000/140, to one or more [hence *multiple*] *pieces of typed data*. For instance, URLs representing instances of (manifestations of) the object. Using multiple resolution, a DOI can be resolved to multiple URLs, other DOIs, or other data types representing items of metadata.

7.2.1. Simple Resolution

A DOI persistently identifies a specific intellectual property entity, which may or may not be an Internet-accessible file. The analogy is with the ISBN (which identifies the book) and the shelf-mark (which identifies the place where the book is to be found). When the location changes, the shelf mark changes - but the ISBN does not.

The earliest application of the DOI was for simple, single point resolution. Each DOI had a single URL to which it could be resolved. Thus making a DOI a persistent identifier.

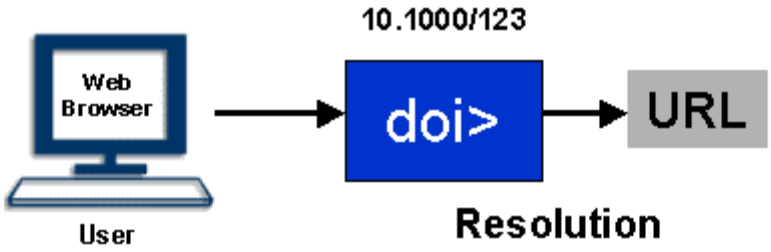

Fig. 3: Single Resolution (14)

7.2.2. Multiple Resolution

A DOI is a *name* for an entity; in the network environment, there may be many identical copies ("instances") of the same piece of content. A DOI can be resolved to an arbitrary number of different points on the Internet: multiple URLs, other DOIs, other data types. This is called *'multiple resolution'*.

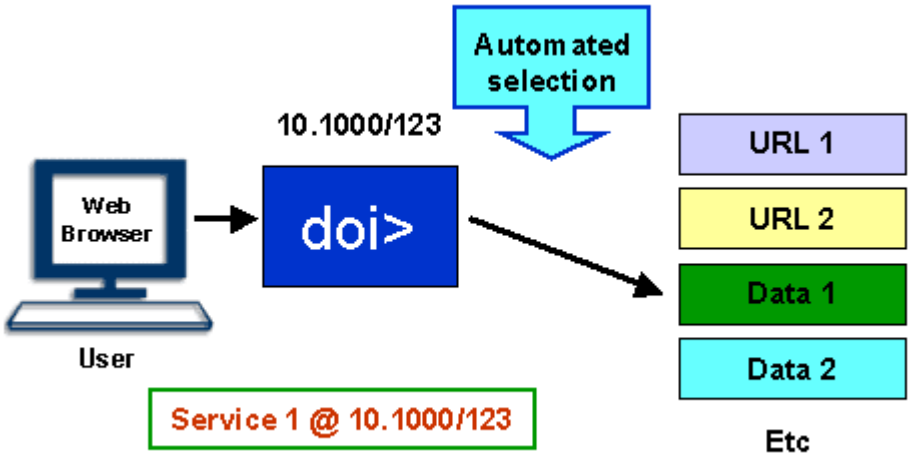

Fig. 4: Multiple Resolution (14)

If the DOI can point to many different possible resolutions, the user may be provided with a list from which to make a manual choice. However, this is not a complete solution

for an increasingly networked and complex environment. The solution is to develop automated systems (users' application software) to handle this.

7.3. DOI as a Universal Resource Name (URN)

The DOI can be considered a Uniform Resource Name, as originally described in W3C and IETF architecture documents. The Technical Working Group of IDF is considering whether a formal specification of DOI as URN is advised or not. At the more general level (URN as persistent name as opposed to URL as current locator) DOIs are, particularly as implemented through the handle system, URNs. At the more formal level of IETF drafts and standards, however, DOIs are not URNs as *doi* is not registered as a *'urn namespace'*, e.g. *urn:doi:10.1000/1*. But it would be considered, if there were a widely deployed and efficient URN resolution system in future.

8. References

1. Berners-Lee, T., Fielding, R., Irvine, U. C., & Masinter., L. (1998, August). *Uniform Resource Identifiers (URI): Generic Syntax*, from http://www.ietf.org/rfc/rfc2396.txt
2. Berners-Lee, T., Masinter, L., & McCahill, M. (1994, December). *Uniform Resource Locators (URL)*, from http://www.w3.org/Addressing/rfc1738.txt
3. *Uniform Resource Identifiers & the simple discovery protocol*. from http://martinh.net/uris/node2.html
4. Hoffman, P. (1995). *The User Interface of URLs*, from http://www.isoc.org/HMP/PAPER/016/html/paper.html
5. Sollins, K., & Masinter, L. (1994, December). *Functional Requirements for Uniform Resource Names*, from http://www.w3.org/Addressing/rfc1737.txt
6. *Metadata working group*. from http://metadata-wg.mannlib.cornell.edu/
7. *URN creation tool, Nordic Metadata Project*. from http://www.lub.lu.se/metadata/URN-help.html
8. Rozenfeld, Slawek. *The ISSN Portal for Accessing Services Related to Serials and Integrating Resources. In* 11th Nordic Conference on Information and Documentation, 2001. from http://www.bokis.is/iod2001/papers/Rozenfeld_paper.doc
9. *Using the ISSN (International Serial Standard Number) as URN (Uniform Resource Names) within an ISSN-URN Namespace.* RFC 3044. January 2001 from http://www.faqs.org/rfcs/rfc3044.html
10. *ISSN-URN Demonstrator.* Version 0.3 beta from http://urn.issn.org/
11. *PURL Home Page*. from http://purl.oclc.org/
12. *Introduction to the Handle System*. from http://www.handle.net/introduction.html
13. Sam Sun, Larry Lannom, Brian Boesch. *Handle System Overview.* RFC 3650, November 2003. from http://www.handle.net/rfc/rfc3650.html
14. *What is a Digital Object Identifier?* from http://www.doi.org/handbook_2000/what_is_a_doi.html