Workshop on
Digital Libraries: Theory and Practice
March, 2003
DRTC, Bangalore

Paper: L

Naming and Addressing Conventions for Digital Resources

Dimple Patel
Documentation and Research Training Centre
Indian Statistical Institute
Bangalore-560 059
email: *dimple@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, URCs, PURLs, and DOIs are discussed in detail in this paper.

1. INTRODUCTION

In this world, to gain access to any entity (object or human) two things are very essential to know about the entity i.e. identification and location of the entity. By identity here I mean, the entity should have a name. Like if I am searching for a person then I should know the name of that person so that I can distinctly identify him/her from others. Secondly, if I want to contact/meet that person, I should know where I can find him/her i.e. I should have a house address/email id/phone number of the person. In the same way, the Internet is a space where resources (like computer files or images or documents in the form of HTML, PDF, etc), are stored. To gain access to these resources some kind of naming and addressing systems are necessary to identify and locate them. There exist different naming and addressing schemes, to identify and locate resources on the Internet. This tutorial will discuss about these various naming/addressing schemes, how they work, etc.

The IETF (Internet Engineering Task Force) is developing the schemes such as the URI, URL, URN. The Internet Engineering Task Force is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and it's smooth operation. It is open to any interested individual. It develops the schemes and syntax rules for these schemes and publishes them in the Internet Drafts or the RFCs (Request For Comments) (*See* Appendix I).

2. UNIFORM RESOURCE IDENTIFIER

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, URC, 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 is given in the RFC2396 (*URI Generic Syntax*).

2.1. Examples

URLs are the most widely used scheme under URI. 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. 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 NAMES)

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. Requirements for Functional Capabilities

The URN Working Group had come up with the following requirements (5):

- **Global scope:** A URN is a name with global scope which does not imply a location. It has the same meaning everywhere.
- **Global uniqueness:** The same URN will never be assigned to two different resources.
- **Persistence:** It is intended that the lifetime of a URN be permanent. That is, the URN will be globally unique forever, and may well be used as a reference to a resource well beyond the lifetime of the resource it identifies or of any naming authority involved in the assignment of its name.
- **Scalability:** URNs can be assigned to any resource that might conceivably be available on the network, for hundreds of years.
- **Legacy support:** The scheme must permit the support of existing legacy naming systems, as long as they satisfy the other requirements described here. For example, ISBN (International Standard Book Number) scheme seems to satisfy the functional requirements, and allows an embedding that satisfies the syntactic requirements described here.
- **Extensibility:** Any scheme for URNs must permit future extensions to the scheme.
- **Independence:** It is solely the responsibility of a name issuing authority to determine the conditions under which it will issue a name.
- **Resolution:** A URN will not impede resolution (translation into a URL). To be more specific, for URNs that have corresponding URLs, there must be some feasible mechanism to translate a URN to a URL.

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 Implementation

A URN can be associated with many URLs. A resource should have a single URN, even if it has multiple URLs. A URL may change, but URNs are persistent over time. For a URN to operate, it should be linked to actual Internet location by an intermediate *'resolution service'*. Resolution service is a network-accesible service that can map the URN onto the corresponding resource that it identifies. A URN can be very compact as it does not have to specify the path to the location.

Several European national libraries use NBN (National Bibliography Number)-based URNs for identification of electronic resources in their digital archives. One 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-1039428424571

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 1998, in the CENL (Conference of European National Librarians), it was decided that the participating countries will implement URNs. In Sweden and Finland the generation of URN started in 1998, with the commencement of the Nordic Metadata Project. But this has not been let known outside the library community as there is lack of proper resolution services available globally. 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) The software is capable of delivering URNs for all Nordic countries.

4.5. Embedding URNS into HTML File

A URN only identifies the document. To get access to the document the location i.e. URL of the document is also required. Currently, the URNs cannot be used *'as is'* i.e. they cannot be typed in the Address box like URLs, because an international URN resolution infrastructure is not available yet. The URNs have to be embedded in the HTML file of the resource. This will ensure that the search services will be able to retrieve the document. It has to be placed in the META tag of the HTML document. 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">

For example:

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

5. UNIFORM RESOURCE CHARACTERISTIC (URC)

As per the Internet Working Group on URCs, *"The purpose or function of a URC is to provide a vehicle or structure for the representation of URIs and their associated meta-information."*(8).

The role of URCs in the Internet is being envisioned in three areas:

- To resolve URNs to the list of URL(s) of a resource
- To help in bibliographic searches on Internet
- To validate the authenticity of the information/resource

URC, simply put, is metadata about the URN. They are being developed along with the URN scheme. They are like document surrogates. They contain the name and the location(s) of the resource in the form of URLs, as well as other meta-information like author, title, access restrictions cost, etc.

There is no RFC available on URCs. Only an Internet Draft entitled *'URC Scenarios and Requirements'* (8) is available which has expired in December, 1995. URCs have also been mentioned in RFC 1737 and RFC 2168, but have not been dealt in detail.

5.1. Compone nts of a URC

Each line in an URC consists of a simple attribute/value pair as shown below:

<attribute_name>:<value>

where ***attribute_name*** is one of the specified set of well known attribute_names that should be recognized. Experimental attribute_names should begin with the string ``X-". And, ***value*** is a text string in which contents and encoding is decided by each attribute_name. The string may be of several lines. In such a case, a space should be the as first character on the following lines.

For example (9):
URN:dns:drtc.isibang.ac.in:12345
URL:http://drtc.isibang.ac.in/SRR.html
TTL:1209600
Author:Akku
Content-Type:text/html
Content-Length:23167

URN:dns:drtc.isibang.ac.in:12345
Uniform resource name

URL:http://drtc.isibang.ac.in/SRR.html
Uniform resource locator

Author:Akku
The name of the author of the resource as first, middle and last name. If there are several authors they are separated by a comma.

TTL:1209600
Time to live. This denotes for how long a value is valid and references the attribute/value pair denoted directly preceding it in the URC. The value is given in seconds and infinity is denoted by a single plus sign.

Abstract:This document describes the development of the Internet Protocol and it's advantages versus disadvantages.
A short abstract describing the document referred to.

Other possible attribute/value pairs

- *Collection:* This pair gives the encoder the possibility to specify a list of URNs or URLs that are considered to be in a collection with the given URN.
  Example:
  URN:dns:luth.se:98765
  Collection:
  URN:dns:luth.se:74532
  URN:dns:luth.se:12567
  URN:dns:luth.se:23476
- *Cost*: A field describing the cost of retrieving the document.
  Example:
  Cost:$5.00
- *Version*: A field telling the version of the document. Smaller changes in the document should not mean that a new URN should be allocated. It is up to the owner to decide when a new URN allocation is needed.
  Example:
  Version:1.2
- *Copyright*: A field regarding current copyright issues of this document.
  Example:
  Copyright:This document can be freely distributed as long as it is not altered in any way.

5.2. URN to URL Resolution (9)

The fundamental purpose of the URC service is to map a URN to URL(s) so that a resource can be retrieved if its name is known i.e to take a URN and return a (possibly empty) list of URLs where the resource named by the URN can be found.

- User provides a URN to the browser by clicking on an anchor or by entering text into a dialog box.
- Browser connects to the URC service, possibly through a caching intermediary, and gives it the URN.
- Service returns a (possibly empty) list of locations to the browser. The means by which the URC service determines this list is outside the scope of this usage scenario. Each location must contain a URL. It may also contain information on Content-Type, Price, Signatures, Version, etc. The list of locations is unordered. Note that if a location contains information in addition to the URL, ordering may be used to associate the additional information with a particular URL, but no importance should be placed on one URL appearing before another in the list of locations sent back to the browser.
- The browser uses user-configurable preferences to order the list. For example, a user might prefer HTML to PostScript to text. One user might prefer locations that carried signature information, another might not care. Most would prefer the cheapest version of a resource, and the most recent version. If multiple locations tie, the browser randomizes them in the list to prevent overload of any one server.
- Once the list of locations has been sorted, the browser attempts to retrieve the resource from the first location. If that fails, the next location is tried. This continues until one of the following is true:
  - The browser successfully retrieves the resource
  - The list is exhausted
  - The user tells the browser to cancel the retrieval

- The browser displays the resource to the user

5.3. Authenticity Issues

As the users have to sometimes pay for information, their main concern is about the authenticity of the information. In such a case, the URCs can carry a digital signature of the resource so that the user can verify its authenticity. When the user retrieves a resource where the URC contains a digital signature the browser should retrieve the resource and display it to the user and in the background it should verify the signature. And the user should be alerted if the signature doesn't match. (9)

There are no URC resolution services available as of now. Though, the URI Working Group has worked out the requirements of a URC resolution service.

6. 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 (10):

(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

During the PURL project, a software called the *'PURL Resolver'* program was developed, which redirects the PURL to the associated URL. This software can be downloaded free of cost from the PURL homepage, and anyone can create his/her own resolution service on a server. The working of the PURLs 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.

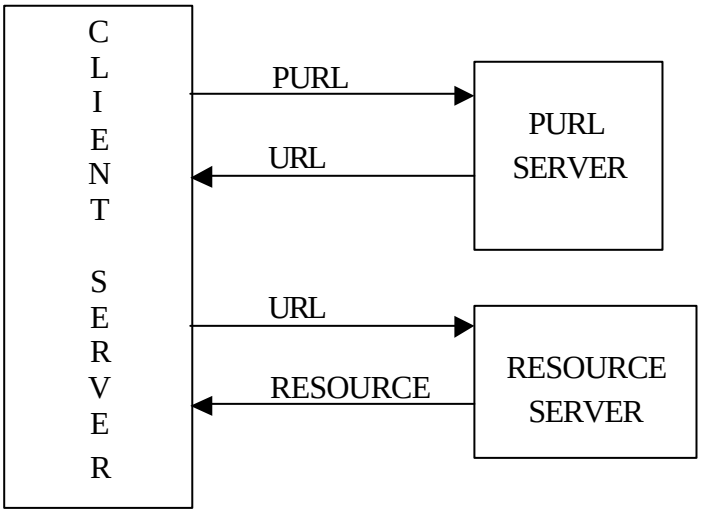

Fig. 1: Purl Resolution Service

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.

7. DIGITAL OBJECT IDENTIFIER (DOI)

The Digital Object Identifier (DOI) is a system for **(11)**:

- 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.

DOI was a project begun by the American Association of Publishers 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. In short, it can be thought of as a kind of catalog. The DOI system is still under development.

7.1. What is DOI?

When publishers created the DOI in 1996, they had two goals in mind (11):

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

The DOI provides a stable and persistent link to the resource/content on the Internet. 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 a permanent number that is independent and that never changes once it is assigned to the content, very similar to the ISBN. Instead of pointing to the location of the specified content *(e.g.* Web site address or URL), 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 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 called the Handle System, and when it receives a DOI request from a user's browser, it translates or "resolves" that DOI to the specific location which the publisher has specified for it, and then automatically re-routes the user's browser to that location.(11)

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 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. (11)

7.2. The Structure of a DOI (11)

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

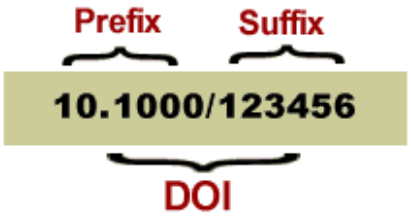

Fig. 2: DOI Structure (11)

DOIs may incorporate any printable characters from the Universal Character Set (UCS-2), of ISO/IEC 10646, which is the character set defined by Unicode v2.0. The UCS-2 character set encompasses most characters used in every major language written today. 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.

7.2.1. The DOI Prefix

The prefix itself 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.

There is no limitation placed on the number of DOI prefixes that any organization may choose to apply for. For example, a publishing company might have a single DOI prefix, or might have a different one for each of its journals, or one for each of its imprints.

The DOI is an opaque string (dumb number). No definitive information can or should be interpreted from the number in use. In particular, the fact that the DOI has a prefix issued by a particular organization should not be used to identify the owner of any given intellectual property -- the DOI remains persistent through ownership changes, and the prefix is unaltered. It identifies uniquely and persistently the content only.

7.2.2. 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 (11):

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

7.3. 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.

7.3.1. Simple Resolution

A DOI persistently identifies a specific intellectual property entity, which may or may not be an Internet-accessible file. The URL identifies a specific address on the Internet. These applications of identification are completely different. One identifies an entity; the other identifies a location (where a specific entity may or may not be found). 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.

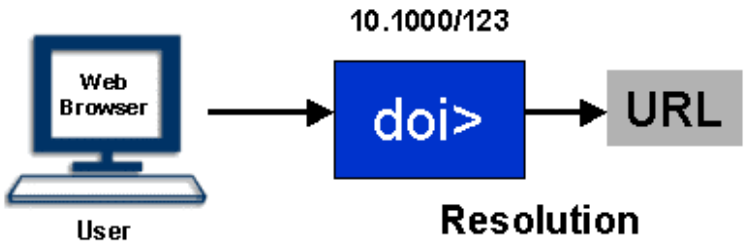

Fig. 3: Single Resolution (11)

7.3.2. 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'*.

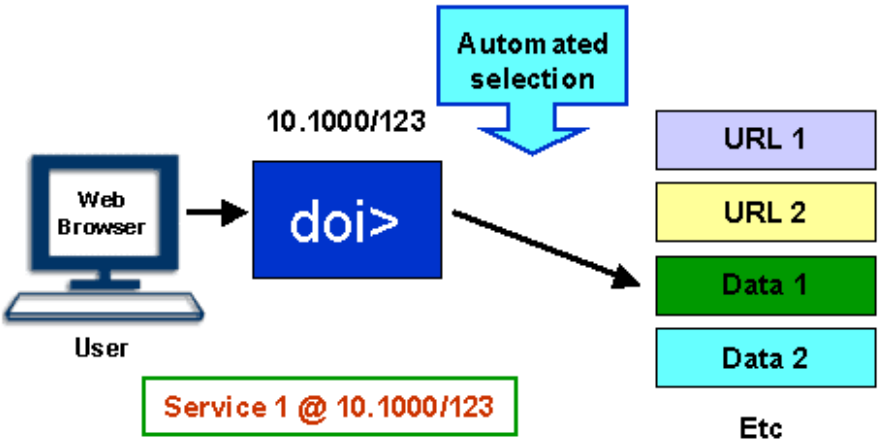

Fig. 4: Multiple Resolution (11)

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.

7.4. DOI as a Universal Resource Identifier (URI)

The DOI, as an identifier, stands on its own. When used in Web context, , it can be put in the form of a URI with a scheme *doi:* In this sense it is no different than a telephone number, for which there is a URI scheme of *tel:*

The specification for DOI as a URI is given in the Internet Draft *'Uniform Resource Identifier (URI) scheme for Digital Object Identifiers (DOIs)'*. (12) This document defines the *'doi'* Uniform Resource Identifier (URI) scheme for DOIs, which allows a DOI to be referenced by a URI for Internet applications.

The *'doi'* URI scheme conforms to the generic URI syntax as specified in RFC2396. UTF-8 encoding is mandated for any DOI transmitted between *'doi'* user agent and any DOI service. Syntax for DOI identifier within the *'doi'* scheme is defined in accordance with ANSI/NISO Z39.84 standard for Digital Object Identifier Syntax. (12)

Some examples of syntactically valid *'doi'* URIs are given below:

doi:10.1000/182
where "10.1000" is the prefix and "182" is the suffix
doi:alpha-beta/182.342-24
where "alpha-beta" is the prefix and "182.342-24" is the suffix
doi:10.abc/ab/cd/ef
where "10.abc" is the prefix and "ab/cd/ef" is the suffix
doi:1.23/2002/january/21/4690
where "1.23" is the prefix and "2002/january/21/4690" is the suffix
<element xmlns="doi:1.23/2002/january/21/4690">

7.5. DOI as a Universal Resource Name (URN)

The DOI can also 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. Daniel, Jr., R., & Michael Mealling (1995, June 27). *URC Scenarios and Requirements*, from http://www.acl.lanl.gov/URI/Scenarios/scenarios_3.asc
9. Parnes, P. *Uniform resource names in the World Wide Web and their resolution*, from http://www.cdt.luth.se/~peppar/master_thesis/node23.html#SECTION00225000000000000000
10. *PURL Home Page.* from http://purl.oclc.org/
11. *What is a Digital Object Identifier?* from http://www.doi.org/handbook_2000/what_is_a_doi.html
12. *The "doi" URI Scheme for Digital Object Identifier (DOI).* (2002, October). from http://www.ietf.org/internet-drafts/draft-paskin-doi-uri-02.txt

Appendix I

Internet Drafts are the working documents of the IETF, its areas and it's working groups. Internet Drafts are valid only for six months and after that they are updated, replaced or may also become obsolete. Hence every Internet Draft carries an instruction at the beginning that these should not be cited as references, else they should be cited as "work in progress". Request For Comments (RFCs) must be first published as Internet Drafts. The RFCs are defined as *"a set of technical and organizational notes about the Internet"*. The RFC series discuss about various aspects of the Internet architecture and functioning like networking protocols, procedures, etc. The official specification documents of the Internet Protocol suite that are defined by the IETF and the Internet Engineering Steering Group (IESG) are recorded and published as RFCs. Hence, Internet Drafts and RFCs play a very important role in the development of Internet standards and protocols and in improving the its functionality.