URN to URC resolution scenario

Abstract

This document is intended to address the issue of URN to URC resolution at a level between the IIIR Vision document [clw/peterd:1] and the various standards documents such as the URL specification. [timb]. This document is also intended to act as some pointers for people who might want to implement URNs in information systems they are building.

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months. Internet-Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

To learn the current status of any Internet-Draft, please check the lid-abstracts.txt listing contained in the Internet-Drafts Shadow Directories on ds.internic.net, nic.nordu.net, ftp.isi.edu, or munnari.oz.au.

Sample URC

Title:          URN to URC resolution scenario
Version:        0.3
URN:            Unknown - how about <urn:ietf/pandora/mitra:urn2urc>
URL:            ftp://pandora.sf.ca.us/pub/mitra/urn2urc-04.txt
Format:         Text/plain
Cost:           0
Date:           19 Feb 93
Comment:        This version has had minor changes since 0.3

Note

Throughout the document, I’ve tried to identify places where discussion is still going on that may affect this proposal, and indicate with ‘{’ and ‘}’).

The first version was written prior to IETF Houston - the discussion has moved along a bit since then, but it isn’t worth updating this document until more choices are clarified.

Acronym Soup

I’ve avoided defining any new acronyms in this document. The following acronyms will be used.

URI     Uniform Resource Identifier: Any of URN, URL, URC etc
URN     Uniform Resource Name: A persistent, location independent identifier for an object.
URL     Uniform Resource Location: The address of an object, contains enough information to identify a protocol and retrieve the object.
URC     Uniform Resource Characteristics: Any combination of one or more URN’s or URLs with meta information. The set of information in a URC is not defined. In some documents this is referred to as
URT or URM. [mm]

Id Authority    That part of a URN which identifies the authority
that issued the URN. In documents written prior to it becoming
a hierarchical entity, [clw/peterd:2] it is usually split into
two parts "Naming Authority" and "Publisher Id"

IIIIR Integration of Internet Information Retrieval.

General scenario
================

The general scenario described here might proceed as follows, although
this document is not constrained by the scenario it is useful to articulate
it so that we can check that what we are proposing makes sense.

1) A client program, running on a user's workstation, receives a
hypertext document, menu, or search result etc containing a number of
URNs.

2) The user selects one of the URNs

3) The client locates a URN->URC resolution service.

4) The client contacts the URN->URC resolution service, and
retrieves a number of URLs for this document, along with meta
information about those URLs (e.g. cost and format) and about the URN
itself.

5) The user, or the client, pick the "best" URL

6) The client either retrieves the URL itself (e.g. via its access
library) or if the URL is for a access method it doesn't speak, via a
gateway service.

7) The client either displays the object, or if its in a format it doesn't
handle, launches an appropriate viewer.

Technical detail for each step
==============================

1) The client program receives the URNs

The URNs are embedded in an object, or other data structure, the client
needs a way to locate and extract those URNs.

The current proposal is that in
text they are always of the form <urn:xxx/yyy:ABC123456> where:

xxx/yyy is a hierarchical identifier (read left to right) for the ID
authority. (Note other proposals are to use one or more ":" as separators
for the hierarchy, or to reverse it to look like a FQDN)

ABC123456 is an opaque string assigned by the ID authority.

For more information on URNs see the current version of the URN
document [clw/peterd:2]

2) The user selects one of those URNs.

For this to be meaningful their must be enough meta-information with
the URN for the user to make a selection. Depending on the protocol,
this might be outside the standardisation process, (e.g. a
textual description in a mail message. Or it might be part of another
protocol (e.g. the title in gopher or html). Or it might be that the
URN is received embedded in a URC

3) The client locates a URN->URC resolution service.

The URN needs to contain enough information within it to locate the
resolution service. The client can extract the ID Authority from the
URN in the example above this would be xxx/yyy, this is reversed and
".urn" appended to form a FQDN of yyy.xxx.urn
If other punctuation schemes are adopted, then the process will change, but the principle remain the same. Ditto if we adopt a different top-level domain. This does have implications for the character set allowed in the ID Authority part of a URN, it either has to be those characters allowed in a FQDN, or an escaping scheme chosen.

This FQDN, can be passed to the DNS which can resolve it to an address, while this might involve several network accesses to traverse the hierarchy, the standard caching and UDP parts of DNS make this an efficient process. Typically this requires just a call to "gethostbyname" which returns a IP address.

There have been some concerns raised about not increasing the load on the fragile DNS system and software. Note also that in this scheme the DNS needs no records changed or added, and only ID authorities are registered – not documents. Total increased load on DNS for any transaction is going to be of a similar order as the load for any document retrieval etc.

4) The client contacts the URN->URC service.

If we are to avoid mucking with the DNS, then the URN->URC service is going to have to be on a registered port, talking a known protocol.

Currently the proposal is to use a subset of the whois++ protocol, but sitting on a different port. The simplest query is of the form:

Client->Server: Template=URC;URN="urn:xxx/yyy:ABCD123456"

Server->Client: URN: urn:xxx/yyy:ABCD123456
Author: Mitra <mitra@path.net>
URL: gopher://path.net/00/papers/mitra/urn2urc
Format: Text/plain
URL: ftp://path.net/pub/docs/urn2urc.ps
Format: Application/postscript

Decisions are needed about what the minimum set of queries for URN->URC resolution are, also whether the returned information is in whois++ format, or is in URC format, however we define that.

However this is defined, it is going to be a subset of the evolving whois++, and is going to have to be an extendable protocol, in the sense that we are going to want to add more functionality to URN servers as time goes by. Therefore, URN->URC servers should fail gracefully if a client requests a function they dont support, and clients should behave gracefully if the server doesent support a feature they request. Crashing because you see an unrecognised field, or request is NOT conformance with this standard. See the whois++ document [peterd:3]

(Note: Whois++ needs to add a statement that order can not be arbitrarily rearranged in templates)

5) The user, or client chooses the "best" URL

In some cases the URC will contain enough information to pick the document without further input from the user. For instance, in the above case, if the client doesent support Postscript, then it might automatically select the "Text/plain" version.

In most cases, the client is going to have to present some kind of menu, or dialog box to a user for him or her to make that decision. This means that the meta-information in a template should ideally be interpretable by the computer, and should if possible be human-readable.

It would be usefull to enable experimentation at this time, before agreement on these fields is reached. So for now, template fields shall consist of any registered mime field (e.g. Content type) or any IAFA template field. If these are insufficient then select a field
from the report of the "non-existant" Data Elements Working Group [NEEDWG]. If other fields are needed, prepend them with "X-".

It is hoped that the URI group, or some other WG can gradually standardize the contents of most of these fields. However in the shifting world of information systems this will never be a complete task, so clients should never choke on an element they dont recognize.

6) The client retrieves the URL -

The URL theoretically contains enough information for a client to retrieve it. There are three possibilities for what a well-behaved client might do:

a) The client is clever enough to understand the protocol, and passes the URL to its access library.

b) The client knows about the protocol, but cannot handle it itself, in which case it can pass the URL to a gateway that it knows handles this protocol.

c) The client has never heard of the protocol, in which case it should hand the URL to its default gateway, and hope for the best.

Of course, accessing this URL involves DNS lookup and other network functionality, but this is a well understood problem.

7) The client displays the object

The client now has an object - file, menu etc. By virtue of the earlier steps, it also should have enough information to know what to do with it. Typically this will involve either displaying the object itself, or checking in some configuration table for an appropriate application (e.g. xv) to pass the object to.

Conclusion
-----------

Hopefully, this document has outlined a scenario and some ways to achieve, it - I believe the scenario is generic enough to fit many people's needs, if not then lets outline alternative scenarios and determine if the techniques above are sufficient for handling it.

Other docs
-----------

(I'd appreciate URL's etc where these are missing)

X-Ref: peterd:3
Description: Whois++ protocol spec
Author: ??
Title: ??
URL: ??

X-Ref: timbl
Title: Uniform Resource Locators
Author: Tim Berners-Lee <timbl@info.cern.ch>
Date: March 93

X-Ref: NEEDWG
Description: Report of the "non-existant" Element Set Names Working Group
Title: ??
URL: ??

X-Ref: clw/peterd:1
Author: Chris Weider <clw@merit.edu>
Author: Peter Deutsch <peterd@bunyip.com>
Title: A Vision of an Integrated Information Service
Date: Oct 93
URL: ftp://cnri.reston.va.us/internet-drafts/draft-ietf-iir-vision-00.txt