

Why URLs are good URIs, and why they are not

Pierre-Antoine Champin, Jérôme Euzenat, Alain Mille
mailto:champin@lisi.univ-lyon1.fr
mailto:Jerome.Euzenat@inria.fr
mailto:amille@lisi.univ-lyon1.fr

05/05/2001

Note to the reader (06/06/2001)

This note has been much controverted and discussed, when submitted to the RDF interest group. Hence the reader may be interested in reading the following discussions in the mailing list archive (see <http://lists.w3.org/Archives/Public/www-rdf-interest/2001Apr/0020.html>). We are currently working on version 2, in order to take include all these remarks and comments.

1 Introduction

Uniform Resource Identifiers or URIs [1, 3] have first been designed to offer a global and uniform mechanism to identify network accessible resources. More recently, the will to achieve the *Semantic Web* [2], and more particularly the *Resource Description Framework* (RDF) [11] made it a base vocabulary to describe not only network accessible resources, but *any* resource.

As a matter of fact, people are used to handle URIs, but mostly one kind of them: Uniform Resource Locators or URLs [4]. Hence, whenever a resource needs to be identified, a URL is used which corresponds more or less to that resource. This is the *less* part which is concerning, has become a problem for RDF, and may become, in our opinion, a serious obstacle to the Semantic Web.

We will first present our understanding of the notion of resource, which is the ground of the following discussions. Then we will explain why we think that URLs are often misused when employed as URIs, while they nevertheless have some advantages. Finally we discuss straightforward solutions which could be used to keep those advantages, without the drawbacks, based on Uniform Resource Names (URNs) [12].

2 About resources

As far as we know, no specific definition of the term “resource” has ever been given in the literature about the Web or URIs¹. We hence take it in its common meaning; for example from Wordnet <http://cogsci.princeton.edu/~wn/>):

2: a source of aid or support that may be drawn upon when needed:
“the local library is a valuable resource”

As we pointed out in introduction, the web initially handled computer retrievable resources, *i.e.* resources being deliverable online. However the definition above is very general, and depending on the task, any identifiable thing can be considered a resource: a file, a web page, a person, a company, *etc.* In this section, we identify some properties of resources, which should be kept in mind when trying to locate or identify them.

A thing may be several resources, depending on the task From the definition above, the notion of resource is related to the notion of utility: hence the same thing could be considered as different resources by different people. For example, someone’s homepage can be used by anyone else to get information about her, but also by herself because she put in it useful links. Another example is one of the authors, being a resource both as a computer scientist and as a tennis player.

A resource may be several things, depending on the context Conversely, the thing being a given resource may vary with the context, mostly with the moment in time: web pages have their typos corrected, their contents updated, their stylesheets enhanced. People change their haircuts, their opinions or their jobs — tennis players get less challenging... Moreover, a weather report on the web changes daily, the president of the United States is not always the same person.

More generally, different elements of the context can make resources vary. For example, the resource at http://www.w3.org/lcons/w3c_main is an image (namely the W3C logo) formatted as a GIF or a PNG file, depending on the content negotiations between the HTTP server and the web browser.

Some resources are more specific than others The last example does not mean that the specific PNG file is not a resource. Actually, it is, and it has its own URL: http://www.w3.org/lcons/w3c_main.png. Similarly, the weather report of 03/01/2001 or the president of the US at the same date are also resources, more specific than those enumerated above, because their description is more constrained. We say that a specific resource (like the PNG file) is

¹[3] mentions that a URI identifies “a mapping to an entity or a set of entities”. We do not understand this to be a definition of resources, but rather a description of the relation between URIs and resources, insisting on the fact that resources are not always elementary entities.

an *instance*² of a more generic one (like the W3C logo); we also consider any resource to be an instance of itself.

3 URLs as URIs

3.1 Drawbacks

The main problem with URLs when used as URIs, *i.e.* as *identifiers*, is that they usually do not identify what one would expect them to. This is due to a number of properties that URLs have or may have, as specified in [10]; a (probably not exhaustive) list of those problematic properties follows.

URLs are locators That may seem obvious, but despite this obviousness, URLs are often used to identify resources which they do not (and cannot) locate. Common examples are XML namespaces: [5] recommend that namespaces are identified by a URI, which is not required to be usable for retrieval, and they suggest the use of URNs. However, other W3C recommendations use URLs to identify namespaces [6, 8, 11]. Those URLs do not locate the corresponding namespaces (which are abstract things and hence not locatable with the HTTP protocol), and furthermore *are* usable for retrieval (the retrieved data being either a page explaining that the URL identifies a namespace, or an RDF description of the namespace).

Moreover, [6] allow RDF schemas to assign URIs to abstract things (such as classes or properties), but those URIs are built from the URL of the schema as fragments of it³. For example, the URI of the type property of RDF is <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>. As a matter of fact, the property itself is not located by that URL: its description is.

URLs are transient That means they may become invalid after a certain period of time [9]. The section 5 of [10] even implies that an invalid URL may become valid again, but locate another resource, and it is not a formal requirement that URL producers prevent that behavior.

For example, the homepage of the first author is located at <http://www710.univ-lyon1.fr/~champin/>. If he quits that university today, that URL will become invalid. If tomorrow somebody named Champin joins the university, she may get the same URL for her homepage.

3.2 What URLs do identify

In the *immediate interpretation*, a URL identifies the resource retrieved through it. But, as discussed above, there can be several such resources, depending on the context. Hence this interpretation is not appropriate.

²The word *instance* here is employed in the meaning of *example*, rather than in the object oriented programming meaning of “instance of a class”.

³Fragments of resources are defined in [3] as “properties of the data resulting from a retrieval action”; fragment identifiers are appended to a URI with the “#” character.

The *wholistic interpretation* of a URL is the set of all possible resources that can be retrieved via this URL in any possible context. This interpretation is classic in model-theory, and is a natural extension of the previous one. It has however the drawback of not being as intuitive as it seems: in the example of the homepage (section 3.1) the intension would be something like “the homepage of the university member with login `champin`”. Another problem is linked with URLs having a side-effect beyond the retrieved data: a registration CGI script may always return the same “thank you” page, whatever parameters it is given; moreover, a `mailto:` URL is not designed to retrieve anything.

The *process wholistic interpretation* of a URL is the set of all possible processes triggered by the URL used in conformance with its scheme, in any possible context. This interpretation allows to tackle the latter drawback of the previous interpretation: even side-effects are taken into account.

We are aware that the latter interpretation is quite different from the more intuitive two others (though we showed that the second one is not as intuitive as it seems). We agree that these interpretations suit many applications – we see them as a kind of *metonymy*⁴. But still we believe that several efforts related to the Semantic Web, implying formal logic and ontologies, require robust identifiers to deal with.

We also want to insist on the fact that none of these interpretations justifies using URLs to identify resources not located by them.

4 Use URNs instead - a discussion

Contrarily to URLs, URNs (Uniform Resource Names) are designed to persistently identify a given resource. Hence we suggest that they should be more extensively used than they are currently. One problem is how to build these URNs. Another one is how to get the URN corresponding to a given URL. In this section we make some suggestions to solve these problems.

4.1 URNs inheriting the advantages of URLs

Besides the fact that they are commonly used, we see two main reasons which make URLs popular URIs: unicity and retrievability. Hence, a useful URN scheme should have the same advantages, which can be somehow inherited from URLs.

Unicity This is, to our opinion, the most important and often overlooked advantage of using URLs as identifiers. Unicity is guaranteed in a very decentralized way: most people “own” a part of the URL space, which they received from their companies or organizations, which themselves received it from the ICANN (Internet Corporation for Assigned Names and Numbers). This is not the only identifier scheme to use such delegation to guarantee unicity (ISBN

⁴Using the name of a feature of the object instead of the name of the object itself. For example: “you are currently reading *Champin*, Euzenat and Mille”.

numbers do too) but this is probably the most locally distributed. Of course, domain names are not designed to identify all kinds of resources, but only internet domains; they have already been used to other purpose, though, for example to name packages in the Java language, and even in some URN schemes (the most used, though not normalized, being inet:). As a matter of fact, reusing this well working name space to build robust identifiers (including URNs) looks convenient and pragmatical. For example:

```
urn:new-urn-scheme:www710.univ-lyon1.fr/~champin/03-2001/myIdentifier
```

Such an URN scheme nevertheless requires precautions: name subspaces should be *permanently* assigned, which is not currently the case with something like /~champin, as pointed out in section 3.1. This is why a date has been included in the example above⁵ — Note that a similar method is already used for URLs in the W3C to differentiate different versions of the same document.

Retrievability This “advantage” is often pointed out by people identifying abstract resource with URLs: the *retrieved* resource can be a description of the *identified* resource. This is all the more useful that abstract resources can only be described, not retrieved.

We take it as a drawback as well, because it is inconsistent with all the interpretations presented in section 3.2, including the more intuitive ones. Actually, it leads to a confusion between the resource and its description, which are indeed two distinct resources. Furthermore, there is no way to distinguish an “identifying” URL (where the retrieved resource is an instance of the identified resource) from a “descriptive” URL (where the retrieved resource describes the identified resource).

In the URN example above, the prefix could be changed to make it a URL, but then any new-urn-scheme should define precisely what protocol may be used with the given path, and if any, what is the meaning of the retrieved resource (instance, description, something else...). Depending on that meaning, computer retrievable as well as abstract resources could be described without ambiguity by those kinds of URNs.

4.2 What is the URN of this web page?

That question may come to any user of the Semantic Web when retrieving a web page (or any other resource) through a URL. If we want URNs to be extensively used, the answer to this question must be as easy to get as possible.

Hence, we suggest that there is a need for the service described in [7] as L2Ns, returning a list of URNs corresponding to a given URL. We additionally suggest that such a service return the URN list in increasing order of generality: the first one being the identifier of the data retrieved in the current context (may be using the URL scheme proposed in [7]), and the last one being the URN of

⁵Assuming that such a “dated” path is used to create URNs only at the corresponding date, noone will ever create an homonym of that URN.

the named most generic resource included in the wholistic interpretation of the URL.

A solution solution would be to include this service in the retrieval process, either as an RDF description, an XHTML meta-element (`<META type="urn" value="...">`) or an HTTP header field (`Content-Urn=...`).

5 Conclusion

In this note, we discussed the issue of what exactly is identified by URLs, when they are employed as Resource Identifiers (URIs). As a matter of fact, we think that they are often misinterpreted when used as such. The interpretation we proposed, though not the most intuitive, seems to be more robust than more intuitive ones.

We then discussed a way of building URNs inheriting the good properties of URLs (unicity and retrievability) and allowing to identify any (network retrievable or not) resource. We believe that such URN schemes are necessary to achieve the goals of the Semantic Web, since they provide cleaner identifiers than URLs. We finally discussed the necessity of implementing L2Ns services so as to encourage the use of URNs.

References

- [1] Tim Berners-Lee. Universal Resource Identifiers in WWW. urn:ietf:rfc:1630, jun 1994.
<http://www.ietf.org/rfc/rfc1630.txt>.
- [2] Tim Berners-Lee. Semantic web road map, oct 1998.
<http://www.w3.org/DesignIssues/Semantic.html>.
- [3] Tim Berners-Lee, R. Fielding, U.C. Irvine, and L. Masinter. Uniform Resource Identifiers (URI): Generic Syntax. urn:ietf:rfc:2396, aug 1998.
<http://www.ietf.org/rfc/rfc2396.txt>.
- [4] Tim Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). urn:ietf:rfc:1738, dec 1994.
<http://www.ietf.org/rfc/rfc1738.txt>.
- [5] Tim Bray, Dave Hollander, and Andrew Layman. Namespaces in XML. W3C recommendation, jan 1999.
<http://www.w3.org/TR/1999/REC-xml-names-19990114>.
- [6] Dan Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification. W3C proposed recommendation, mar 1999.
<http://www.w3.org/TR/1999/PR-rdf-schema-19990303>.

- [7] R. Daniel. A Trivial Convention for using HTTP in URN Resolution. urn:ietf:rfc:2169, jun 1997.
<http://www.ietf.org/rfc/rfc2169.txt>.
- [8] Steve DeRose, Eve Maler, David Orchard, and Ben Trafford. XML Linking Language (XLink) Version 1.0. W3C candidate recommendation, jul 2000.
<http://www.w3.org/TR/2000/CR-xlink-20000703>.
- [9] Martin Hamilton. Uniform Resource Identifiers and the Simple Discovery Protocol. Technical report, Department of Computer Studies, Loughborough University of Technology, Ashby Road, Loughborough, Leics. LE11 3TU, UK, 1995.
<http://martinh.net/uris/uris.html>.
- [10] J. Kunze. Functional recommendation for internet resource locators. urn:ietf:rfc:1736, feb 1995.
<http://www.ietf.org/rfc/rfc1736.txt>.
- [11] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C recommendation, feb 1999.
<http://www.w3.org/TR/1999/REC-rdf-syntax-19990222>.
- [12] R. Moats. URN Syntax. urn:ietf:rfc:2174, may 1997.
<http://www.ietf.org/rfc/rfc2141.txt>.