Persistent Identifiers and URLs

Martin Fenner, Gobbledygook

June 3, 2015

Just like the rest of the internet, much of our scholarly infrastructure is built around the Hypertext Transfer Protocol (HTTP), increasingly HTTPS for security, and soon HTTP/2 for better performance. In this infrastructure Universal Resource Locators (URLs) are essential to locate resources (sic) such as scholarly articles, datasets, researchers, organizations, or grants. Read this recent Thomson Reuters report for a good recent perspective on this topic. While this works for the most part, there are some issues with URLs - not specific to scholarly content, but particularly import here:

1. multiple URLs can point to the same resource
2. URLs can be long and look ugly
3. URLs can change or break, making it hard or impossible to locate the resource
4. we are used to central indexes (or databases) describing these resources, allowing us to do sophisticated queries not possible in a generic web search, e.g. find all publications by author John Doe, published since 2012.

No. 1 is a problem relevant to all URLs, e.g. web searches or liking/commenting a particular web page. Originally suggested by Google, Canonical URLs are essential for services such as Facebook or Hypothes.is. They have been formalized in rfc6596 and are commonly used.

No. 2 can be a problem, in particular if we are not careful in designing appropriate URLs for landing pages (see next paragraph), but rather use something long and unreadable that also includes query parameters, etc. If we have no control over how the URL looks like, we can use URL shortener services such as bit.ly, which of course have become a common sight on the web. ShortDOIs are an URL shortener for DOIs, but they don't seem to have gained much traction.

No. 3 is a particularly important issue, commonly referred to as **link rot** and described extensively for the scholarly literature, e.g. by (Klein et al., 2014). There are several technical solutions to this problem, a common approach is to use a landing page for the resource that will never change (and follows the recommendations by Tim Berners-Lee for Cool URIs, and then use redirection to point to the current location of the resource. This is easily for changes of the URL path using web server redirect rules. It gets more complicated if the

server name also changes, in particular if it is the server holding the landing page. Thinking this through you realize that the only way this can be done on a larger scale is via one or more centralized services that not only provide the technical infrastructure for a central redirection (or resolver) service, but also come with a social contract of rules that everyone submitting URLs to the service has to follow - a major difference to URL shorteners, which don't solve the link rot problem.

The above is of course a description of the DOI service provided by CrossRef, DataCite, and others, as well as similar persistent identifier services. Unfortunately some persistent identifier services don't do the above: they create and use persistent identifiers, but there is no central resolver service that maps these identifiers back to URLs. This breaks the integration with the bigger scholarly infrastructure based on URLs. One common example are nucleotide sequences such as U65091 (Shioda, Fenner, & Isselbacher, 1997), there is no single corresponding URL because the sequence can be found in all three main nucleotide databases: http://www.ncbi.nlm.nih.gov/nuccore/U65091, http://www.ebi.ac.uk/ena/data/view/U65091, and http://getentry.ddbj.nig.ac.jp/getentry/na/U65091. It would help to have a central resolver, e.g. http://nucleotide.org/U65091 that then redirects to one of the three databases based on geographical location or user preference.

There are also problems with DOIs. They use the Handle system to resolve the identifier to a location, and this system was built in the 1990s as infrastructure independent of URLs or DNS (Domain Name Service), at a time when it wasn't clear yet that URLs and associated standards would become ubiquitous. I don't have numbers, but practically all DOIs are of course now resolved to URLs using the DOI proxy server at http://doi.org (preferred) or http://dx.doi.org. One main consequence of this is that DOIs are frequently not written as URLs - e.g. doi:10.5555/12345678 instead of http://doi.org/10.5555/12345678 - again breaking the integration with the bigger scholarly infrastructure. The CrossRef DOI display guidelines clearly state that DOIs should be written as URLs in *the online environment*, which basically is whenever DOIs are used, as PDFs and even Word documents know how to handle URLs. Unfortunately this guideline is still frequently ignored. The above is of course also true for other persistent identifiers using the Handle system, e.g. EPIC.

The other problem with the DOI system is that it doesn't address issue No. 4, i.e. provide a central metadata index for the resources that use the system. This job is left to the DOI registration agencies such as CrossRef and DataCite, who have implemented a central metadata store (e.g. CrossRef, DataCite) in different ways (e.g. using different metadata schemata), or not at all. This means that we have to look in several places to find all DOIs associated with author John Doe, published since 2012. Obviously we are used to looking up information in multiple places, but not being able to look up the metadata for a DOI without some extra work (finding out the registration agency for the DOI and then going to the respective metadata store) is a problem. One way around these problems

is to use the DOI Content Negotiation Service.

Another problem with the DOI system is more a social than a technical issue. Neither CrossRef nor DataCite seem to enfource that DOIs should alsways resolve to URLs when using a computer program. DOI resolution for humans works fine, but computers, e.g. command line tools such as cURL, can run into issues such as requiring cookies, javascript or user input, or permission problems getting to the journal landing page (see this earlier blog post for some numbers). People seem to forget that a DOI that is not actionable is not really useful, and that scholarly infrastructure is not only used by people, but of course also by automated tools.

The persistent identifiers used in our scholarly infrastructure would benefit from a clearer focus on the problems they should solve, startin with No. 1-4 above. One problem is that we probably focus too much on the persistence problem, implied also by the term **persistent identifier** or **PID**. What we have neglected is the resolvable problem, i.e. making as easy as possible to get from the persistent identifier to the resource and/or its metadata. Based on the Den Haag Manifesto and suggested by Todd Vision, we therefore proposed the term **trusted identifier** with the following characteristics in the conceptual model of interoperability for the ODIN Project (ODIN Project, Fenner, Thorisson, Ruiz, & Brase, 2013):

- are unique on a global scale, allowing large numbers of unique identifiers
- resolve as HTTP URI's with support for content negotiation, and these HTTP URI's should be persistent.
- come with metadata that describe their most relevant properties, including a minimum set of common metadata elements. A search of metadata elements across all trusted identifiers of that service should be possible.
- are interoperable with other identifiers through metadata elements that describe their relationship.
- are issued and managed by an organization that focuses on that goal as its primary mission, has a sustainable business model and a critical mass of member organizations that have agreed to common procedures and policies, has a governing body, and is committed to using open technologies.

While not directly relevant for resolving persistent identifiers as URLs, the last point is really important for any persistent identifier infrastructure, described in detail recently by (Bilder, Lin, & Neylon, 2015).

If I would design a persistent identifier service today (as if we would need yet another persistent identifier service), I would build the system around an URL shortening service that I control. The URLs could look very similar to what we have with DOIs now, e.g. http://doi.org/10.5555/12345678, but it would be clear that persistent identifiers are URLs, not something separate. Plus we could take adavantage of all the lessons learned - and possibly even reuse open source code - with URL shorteners, which are much more widely used than scholarly persistent identifiers.

Update 6/4/15: added link to Thomson Reuters report on identifiers and open data.

References

Bilder, G., Lin, J., & Neylon, C. (2015). Principles for open scholarly infrastructures-v1. https://doi.org/10.6084/m9.figshare.1314859

Klein, M., Van de Sompel, H., Sanderson, R., Shankar, H., Balakireva, L., Zhou, K., & Tobin, R. (2014). Scholarly context not found: one in five articles suffers from reference rot. *PLoS ONE*, *9*(12), e115253. https://doi.org/10.1371/journal.pone.0115253

ODIN Project, Fenner, M., Thorisson, G., Ruiz, S., & Brase, J. (2013). D4.1 conceptual model of interoperability. https://doi.org/10.6084/m9.figshare.824314

Shioda, T., Fenner, M. H., & Isselbacher, K. J. (1997). Mus musculus melanocyte-specific gene 1 (msg1) mRNA, complete cds. ENA. Retrieved from http://www.ebi.ac.uk/ena/data/view/U65091