The Linked Data Strategy for Global Identity

Hugh Glaser • *Seme4*

Harry Halpin • *World Wide Web Consortium*

The Web's promise for planet-scale data integration depends on solving the thorny problem of identity: given one or more possible identifiers, how can we determine whether they refer to the same or different things? Here, the authors discuss various ways to deal with the identity problem in the context of linked data.

Identity is easily one of the most difficult research areas on the Web and Semantic Web, and one that needs both practical solutions and multidisciplinary research. Identity is how to refer reliably to anything, abstract or more concrete, over time and space, and in different contexts. We're used to identity being quite simple, as your name easily refers to you when another person is speaking to you. Yet on closer inspection, and at a Web scale, identity is quite tricky, as when you type your name into a search engine and see that it can refer to many other people in different contexts. It might even refer to you in a context that you didn't intend! Classical systems often envision a world of discrete objects with given absolute identities, organized in a single, monolithic way. *Linked data*, on the other hand, consists of decentralized fragments of data linked together in possibly conflicting ways and with no singular set of referents.

The linked data project, which aims to expand the Web beyond documents to data, was kick-started by Tim Berners-Lee, the Web's widely acclaimed inventor. So, it's no surprise that Berners-Lee proposed solving the identity problem using HTTP URIs to identify not just webpages but everything — including real-world entities and intangible concepts. To illustrate, Berners-Lee's webpage is `http://www.w3.org/People/Berners-Lee/`, whereas the URI for the flesh-and-blood Berners-Lee is `http://www.w3.org/People/Berners-Lee/card#i`. Using URIs rather than natural language labels is precisely what separates linked data from earlier attempts to build knowledge representation languages in artificial intelligence. Just as the Web consists of URIs where anyone can post a webpage and create a link, anyone can create a URI to identify anything and make a statement of equivalence between URIs.

In his introductory installment of this department, Tom Heath emphasized that links between datasets are the "very essence of linked data,"[1] just as hyperlinks are the essence of the current Web. For example, we might want to state that both Berners-Lee's Wikipedia entry and his homepage are about the same real-world individual. Creating a URI for yet another webpage and hyperlinking it to a related page is straightforward; determining whether two distinct linked data URIs are about the same or different things is much more difficult.

Here, we look at the issues and challenges inherent to using URIs to create identities. We examine how we might answer the questions of identity and equivalence, and discuss the problems of managing this information in the context of linked data.

HTTP URIs as Identifiers

Using URIs in linked data is nothing short of audacious and hints at a certain technological hubris: identity has long been a crucial problem

1089-7801/12/$31.00 © 2012 IEEE

for both computer science and related fields ranging from library science to linguistics. Although learning a shared repository of identifiable names is second nature to any speaking human being, precisely how we attach "names" to concrete actions and parts of the world still bedevils cognitive scientists and philosophers. If we can learn anything from the past few hundred years of debate, it's that we should consider identity to be more a social construct than a technical problem.

One way to deal with identity is to establish a common social convention that identifies particular things in a uniform manner that's easily re-used in diverse contexts. Government identity cards are an excellent example of an identity system established by fiat: imagine how hard it was to determine a person's identity before the invention of photographs and other identity documents. Perhaps the most astonishing example, dating back more than 35 years, is the Universal Product Code and its successors (that is, the familiar barcode on almost all products). All these were motivated by powerful use cases involving delivering messages, paying taxes, or tracking goods, so it's unsurprising that as the information revolution reaches maturity, identifying data itself is increasingly a priority.

Why not just establish a new kind of centralized registry like ICANN to maintain identifiers for real-world entities such as people? This super-registry could ensure that everything in the world has a unique identifier. Many an identity scheme with such cosmic pretensions has already been tried: DOIs (such as 10.1109/ICDM.2004.10104) have found success only in the realm of printed materials. A registry for Uniform Resource Names (URNs) aimed to identify those things outside the Web, with URLs meant only for webpages and other network-accessible objects. Yet few people ever registered a URN, so both URNs and URLs re-merged and were relabeled URIs.

The Web is a space of resources (any item of interest) where any resource can be identified by at least one distinct URI. Despite assertions that the Web is decentralized, a critical point of centralization comes from ICANN's control over the DNS — that is, ICANN has power via fiat to license out to registrars the ability to mint domain names such as w3.org. Yet the domain name registry doesn't know that the URIs `http://www.w3.org/People` and `http://www.w3.org/People/Berners-Lee` identify different resources. Whereas domain registrars are centralized, URIs are themselves decentralized: once someone buys or has access to a domain name, he or she can use a theoretically infinite number of hierarchical components (given by slashes) to mint as many new URIs as they want.

The key advantage of HTTP URIs over any other identification scheme is that linked data principles say these URIs should be dereferenceable and so return some useful description of what the URI identifies when accessed in a Web browser or computer application using HTTP GET. Yet how do we avoid confusing URIs for webpages about things with the things themselves? Thanks to little HTTP tricks called *303 redirection* and *content negotiation*, when a linked data application dereferences a linked data URI, that URI automatically redirects to another URI to return data in the Resource Description Framework (RDF, the W3C standard for Semantic Web data). When accessed by a browser, the same URI redirects to yet another URI and returns a hypertext webpage with a human-readable description of the document. This hack creates three URIs from one: `http://dbpedia.org/resource/Engelbert_Humperdinck` for the thing itself, which then redirects to `http://dbpedia.org/data/Engelbert_Humperdinck` for data, and `http://dbpedia.org/html/Engelbert_Humperdinck` for the webpage.

Determining Equivalence between Identities

Managing identity is the first issue that arises when we want to add new information to the Web of linked data. Let's say you wanted to add some data about the musician Engelbert Humperdinck. You might want to re-use another URI in your dataset or mint a new one. To discover a URI for Engelbert Humperdinck, you could use a linked data search engine such as Sindice (http://sindice.com) to retrieve other linked data URIs that might be about your item of interest. In a top-down approach, you could also look at large existing datasets such as DBPedia (a linked data export of Wikipedia at http://dbpedia.org) that have linked data URIs for all Wikipedia pages and categories. Government open data sites such as http://data.gov and http://data.gov.uk also provide naturally authoritative URIs for schools, roads, hospitals, and the like. Re-using well-known URIs facilitates discovery, allows for instant linking with other datasets, and exploits RDF's ability to instantly "mash up" data based on URI matching. However, what if the resource in your new dataset isn't exactly the same as a URI in an existing one? Worse, what if the owner of the other URI changes the meaning by altering the RDF it returns? Or consumers of the new dataset assume that you somehow endorse the remote URIs? Consider choosing a URI from somewhere such as DBPedia to make your own statements about a controversial topic such as global warming. Such a scenario could present many of these problems, especially as the associated Wikipedia page changes over time.

An alternative bottom-up approach is to let a thousand URIs bloom and accept that many identifiers will exist for any given resource. Creating yet another identifier seems to be a counterintuitive solution to the problem of too many identifiers, but this unstructured approach is attractive and in some sense natural for the Web. It also means that linked data publishers must be able to identify when two URIs are about the same thing, and that applications must be able to process sets of equivalent URIs. We can represent URI equivalence in various ways depending on the URI type and the equivalence statement's strength. In linked data, the current practice for linking two individuals that are about the exact same thing is to use a `sameAs` link from the Web Ontology Language (OWL) vocabulary. In RDF, someone stating that `http://www.w3.org/People/Berners-Lee/card#i owl:sameAs http://dbpedia.org/resource/Tim_Berners-Lee` makes a strong statement that the URI for Tim Berners-Lee in DBPedia is the exact same as the one hosted by the W3C (`owl:sameAs` is shorthand for the URI `http://www.w3.org/2002/07/owl#sameAs`).

Although `sameAs` is by far the most popular kind of link, it has a precise semantics — namely, that both things share all the same properties. This is fine for data about Tim Berners-Lee in Wikipedia obviously referring to the same Tim Berners-Lee as his personal URI, even noting that the data in Wikipedia includes many properties, such as birth date, that are missing from his personal site at the W3C. Because the different URIs purport to identify the same person, the information at both URIs can be merged without much worry. However, given that anyone can add equivalence links, these links might not always be correct in linked data. For example, a study over real-world linked data[2] shows that while roughly half the equivalence links examined were used correctly, `sameAs` sometimes links things that are merely related or even just incorrect. When the OpenCyc dataset defines a chemical element as the set of all pieces of the pure element, then sodium has exactly 23 neutrons. The OpenCyc sodium URI has a `sameAs` link to sodium in DBPedia, but as defined there, sodium includes all isotopes regardless of whether they have a different number of neutrons than "standard" sodium. Worse, a `sameAs` link that (incorrectly) connects Tim Berners-Lee to the concept of a person would result in all people having Tim's birthday. The threat then is that inappropriate use of `sameAs` combined with overambitious inference engines could lead linked data to become a semantic soup in which all things are equivalent.

The concept of equivalence, and therefore identity over space, time, and context raises some complex questions. If I replace one rotten plank on my boat, it appears to be the same boat, but what if I replace all the planks? The prevalent view in the linked data community tends to pragmatically lean toward "buyer beware." That is, if a data producer finds it useful to consider two or more URIs equivalent, then asserting their equivalence is sensible; an application consuming this data, however, should check to make sure it trusts these relationships before mashing up properties or running an inference engine. Luckily, linked data doesn't force us to declare that things are exactly the same. Strictly speaking, we can label properties and classes with `equivalentProperty` and `equivalentClass`, respectively (from OWL). Concepts should use other properties — such as `exactMatch` and `closeMatch` from the Simple Knowledge Organization System (SKOS) — that have no semantic ramifications and are thus weaker notions of equivalence, as well as RDF's more vague `seeAlso` predicate.

The open world assumption states that just because something isn't explicitly stated, doesn't mean that it might not be stated elsewhere. Unless two URIs are stated to be nonequivalent, they could be the same thing, so nonequivalence can be just as important as equivalence. Knowing that two URIs refer to things that you might have thought were the same but are, in fact, different is valuable information. How many Engelbert Humperdincks from the world of music do you know about?

Applications for Discovering and Managing Identity

Trying to bring new data into the linked data world, such as bringing in a new dataset from a database, presents a cold-start problem. Although re-using URIs is recommended where possible, it's often easier and more reliable to map many identifiers in the original dataset to automatically generated new URIs, using a mapping language such as RDB2RDF (www.w3.org/2001/sw/rdb2rdf). However, this results in many URIs with no links at all to outside datasets. In practice, automated tools such as Silk (http://www4.wiwiss.fu-berlin.de/bizer/silk/) usually detect new links between datasets by identifying possible URI equivalences. To do this, such tools must examine a URI's "meaning" by retrieving associated RDF graphs and then comparing them to the graphs other URIs produce. They can then quantify the closeness of this match and deploy various heuristics that depend on the resource type (such as a book) and the associated properties' values (such as whether their ISBN numbers match). As time goes on, with more links made, the matching process becomes more reliable because it can match using the new URI equivalences, rather than the text and other values from the original dataset.

For example, if we were looking at academic publications, it might be sufficient that the titles of two resources of type "publication" match exactly and that the strings that are the authors match within some Levenstein distance (although this heuristic could confuse workshop publications with publications in journals). To find a good proportion of the correct equivalences, we must accept some false positives, but to avoid too many false positives, we must accept some false negatives. Because linked data applications aim to exploit Web-driven network effects, to be completely conservative about making equivalence assertions would reduce the links in the Web of linked data and limit its value.

Another question is whether to put equivalence relationships directly inside the RDF data a URI returns or somewhere else. Both approaches have their advantages: If we can retrieve equivalent URIs directly from the URI, then publishing and finding this data is straightforward. If this information is elsewhere, then we can manage sets of equivalent URIs (such as sets of `sameAs` links) separately, which makes sense because such data might have different provenance and licensing. Also, the dataset owner probably wants the equivalence information put in search indices to aid discovery, but wouldn't necessarily permit republication of the dataset. The coreference service (CRS) lets dataset publishers provide linkage information in associated equivalence link-bases as a separate view of the data.[3] The CRS doesn't become another authority with a new URI, but rather is a service that simply takes a URI and returns a set of equivalent URIs (possibly singleton). An example of such equivalence information is available at http://sameas.org. You'll find that the sameAs.org matching is very liberal, with a higher proportion of false positives to avoid false negatives. Other stores, such as that from Freebase (http://sameas.org/store/freebase/), aim to provide more authoritative equivalence link-bases of their own data; they have fewer false positives but are consequently more likely to have false negatives.

We're only just beginning to explore the vast field of identity, and more work is needed before linked data can fulfill its full potential. Techniques from information retrieval, databases, and knowledge representation are all necessary, including coreference resolution, entity reconciliation, and ontology alignment. Following in the footsteps of work such as Yahoo's Semantic Search contest (http://semsearch.yahoo.com), the next step for the community is to create new challenges and a gold-standard around identity to produce better automated systems for linking data together. Also, given that having perfectly correct automatic systems is difficult, we should explore crowdsourcing over linked data as well as read-write linked data interfaces that let ordinary users annotate and modify data in their own browsers. Lastly, empirical work has just begun to explore how people use identity "in the wild" via linked data, and this information should help push new data-driven standardization.

The entire bet of the linked data enterprise critically rests on using URIs to create identities for everything. Whether this succeeds might very well determine whether information integration will be trapped in centralized proprietary databases or integrated globally in a decentralized manner with open standards. Given the tremendous amount of data being created and the Web's ubiquitous nature, URIs and equivalence links might be the best chance we have of solving the identity problem, transforming a profoundly difficult philosophical issue into a concrete engineering project.

References

1. T. Heath, "Linked Data – Welcome to the Data Network," *IEEE Internet Computing*, vol. 15, no. 6, 2011, pp. 70–73; http://dx.doi.org/10.1109/MIC.2011.153.
2. H. Halpin et al., "When owl:sameas Isn't the Same: An Analysis of Identity in Linked Data," *Proc. 9th Int'l Semantic Web Conf.* (ISWC 10), Springer, 2010, pp. 305–320; http://data.semanticweb.org/conference/iswc/2010/paper/261.
3. H. Glaser, A. Jaffri, and I. Millard, "Managing Co-reference on the Semantic Web," *Proc. World Wide Web Workshop: Linked Data on the Web* (LDOW 09), 2009; http://eprints.ecs.soton.ac.uk/17587/.

Hugh Glaser is the chief architect at Seme4 and a visiting research fellow in Electronics and Computer Science at the University of Southampton. His interests involve building systems around Semantic Web technologies that are scalable and fit for purpose. Contact him at hugh.glaser@seme4.com.

Harry Halpin is a postdoctoral researcher for the World Wide Web Consortium at the Massachusetts Institute of Technology and a visiting Marie Curie fellow at l`Institut de recherche et d`innovation, at Centre Pompidou in France. His research interests center on establishing a standardized and secure social layer for the Web and its implications for collective intelligence. Halpin has a PhD in informatics from the University of Edinburgh. Contact him at hhalpin@w3.org.