

When owl:sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web

Harry Halpin and Ivan Herman
World Wide Web Consortium
ERCIM
Sophia, France
hhalpin@w3.org/ivan@w3.org

Patrick J. Hayes
Institute for Human and Machine Cognition
40 South Alcaniz St.
Pensacola, USA
phayes@ihmc.us

ABSTRACT

In Linked Data, the use of *owl:sameAs* is ubiquitous in ‘inter-linking’ data-sets. However, there is a lurking suspicion within the Linked Data community that this use of *owl:sameAs* may be somehow incorrect, in particular with regards to its interactions with inference. In fact, *owl:sameAs* can be considered just one type of ‘identity link,’ a link that declares two items to be identical in some fashion. We outline four alternative readings of *owl:sameAs*, showing with examples how it is being (ab)used on the Web of data. Then we present possible solutions to this problem by introducing alternative identity links that rely on named graphs.

Categories and Subject Descriptors

H.3.d [Information Technology and Systems]: Meta-data

General Terms

Knowledge Representation

Keywords

Linked Data, ontology, resource, Web architecture

1. INTRODUCTION

As large numbers of independently developed data-sets have been introduced to the Web as Linked Data, the vexing problem of identity has returned with a vengeance to the Semantic Web. As the ubiquitous *owl:sameAs* property is used as the RDF property to connect these data-sets, it has been dubbed the *owl:sameAs* problem. However, the problem of identity lies not within Linked Data or within the Semantic Web languages, but is an outstanding and well-known – if sometimes not precisely labeled – issue in pre-Semantic Web knowledge representation languages in artificial intelligence. What precisely is new in its latest guise of this problem on the Web of Linked Data is that this is the first time the problem is being encountered by different individuals attempting to *independently* knit their knowledge representations together using the same standardized language. Much of the supposed “crisis” over the proliferation of *owl:sameAs* in Linked Data can be traced to the fact that these uses of *owl:sameAs* tend to be mutually incompatible, and almost always violate the rather strict logical

Copyright is held by the author/owner(s).

RDF Next Steps Workshop, June 26-27, 2010, Palo Alto, USA.

semantics of identity demanded by *owl:sameAs*. However, the exact types of distinctions made by these individuals are important, even if they contradict the relevant specification of *owl:sameAs*. First, these uses and abuses of *owl:sameAs* demonstrate for the first time in the history of knowledge representation how precisely these problems play out in the wild. Second, as the Semantic Web is a project in development, it is always possible to specify anew different and new kinds of language constructs and more clearly specified best practices to align the specifications with the actual empirical use of the Semantic Web in the wild.

First, we will give an overview of the problem of identity and its somewhat dusty lineage in artificial intelligence, if only to show how what was already a known issue for knowledge representation becomes even more exacerbated when knowledge representation goes global for the Semantic Web. Then, four distinct uses of *owl:sameAs* are discussed in addition to the precise idea of “same thing as,” namely:

- Same Thing As But Different Context
- Same Thing As But Referentially Opaque
- Represents
- Very Similar To

Finally, a number of suggestions for how the current situation can be improved are sketched. The necessity of both semantic and theoretical work is given as well.

2. THE IDENTITY CRISIS OF LINKED DATA

Contrary to popular belief in some circles, formal semantics are not a silver bullet. Just because a construct in a knowledge representation language is prescribed a behavior using formal semantics does not necessarily mean that people will follow those semantics when actually using that language “in the wild.” This can be laid down to a wide variety of reasons. In particular, the language may not provide the facilities needed by people as they actually try to encode knowledge, so they may use a construct that *seems* close enough to their desired one. A combination of not reading specifications - especially formal semantics, which even most software developers and engineers lack training in - and the labeling of constructs with “English-like” mnemonics naturally will lead to the use of a knowledge representation language by actual users that varies from what its designers intended. In decentralized systems like the Semantic Web, this problem is naturally exacerbated. However, far from

being a sign of abuse, it is a sign of success, as it means that the Semantic Web is actually being deployed outside academia and research labs.

The problem has definitely arisen on the Semantic Web in terms of the use of *owl:sameAs* in Linked Data. In Linked Data, each item of interest is given a URI, that in turn redirects to either human-readable HTML or machine-readable RDF depending on content negotiation. The URI for the item itself, which is called rather confusingly a “non-information resource” in Linked Data circles, as a web-page or RDF graph would be an information resource, as the “distinguishing characteristic of these resources is that all of their essential characteristics can be conveyed in a message” [3]. Usually, this data is released in some sort of automated or semi-automated manner, often by mapping relational data to RDF. Somehow, a URI is chosen for each identifier in the data-set that is exported in Linked Data. Although the general thinking in RDF (and thus, the main idea behind the ability of RDF graph merge) was that URIs would be re-used, in practice URIs are simply minted anew for each identifier in a Linked Data set. As opposed to the simple exporting of data-sets into RDF, what puts the *links* in Linked Data is the use of what we term *identity links* - links that define two things to be identical or otherwise closely related - to link between diverse and heterogeneous data-sets. While there has been some research that deals with this problem [4], the scope of the problem is just beginning to be understood.

The most typical link used is *owl:sameAs*, which is in general used to say “that two URI references actually refer to the same thing” [1]. For example, the city of Paris is referenced in a number of different Linked Data-sets: ranging from OpenCyc to the New York Times. In DBpedia, a Linked Data export of Wikipedia, these data-sets are connected by *owl:sameAs*. In particular, *dbpedia:Paris* is *owl:sameAs* as both the *opencyc:CityOfParisFrance* and *opencyc:Paris_DepartmentFrance*, as OpenCyc distinguishes that “the department of Paris. Paris_DepartmentFrance is a distinct geopolitical entity from CityOfParisFrance, despite the fact that both share the same territory, while Wikipedia does not make this distinction.

3. THE SEMANTICS OF OWL:SAMEAS AND ALTERNATIVES

At first, this use of *owl:sameAs* seems to be harmless. Its actual definition is that “the built-in OWL property *owl:sameAs* links an individual to an individual” and “Such an *owl:sameAs* statement indicates that two URI references actually refer to the same thing: the individuals have the same identity” [6]. Furthermore, OWL states that “It is unrealistic to assume everybody will use the same name to refer to individuals. That would require some grand design, which is contrary to the spirit of the web” [6].

However, *owl:sameAs* does have a particular semantics of individual identity, namely that the two individuals are *exactly* the same and so share all the same *properties*. Given that OWL has no unique name assumption, once there is an application of *owl:sameAs* to two different URIs, then any statement that is given to a single URI is true for every other URI that has an *owl:sameAs* link. Furthermore, while in OWL Full *owl:sameAs* can be considered to be the same as between *any* URIs as classes can be considered “individual”

instances of other classes and properties can be considered individuals, in OWL DL in order to preserve decidability individuals are strictly separated from classes, and so one should use OWL DL *equivalentClass* and *equivalentProperty* instead. At least in OWL 1.0 DL, quick-and-dirty use of *owl:sameAs* will almost always lead to OWL Full, which very little work has been done on in terms of efficient implementations of inference. Interestingly enough, in OWL 2 it is possible to use the same URI to denote classes as well as individuals, except that certain inferences cannot be drawn (and thus leading to the open question of whether or not *owl:sameAs* inference falls under OWL 2 RL). Regardless, the real trick with *owl:sameAs* is that it works both ways: as it is both symmetric and transitive, so that anyone can link to your data-set with *owl:sameAs* from anywhere else on the Web without your permission, and any statement they make about their own URI will immediately apply to yours. As imaginable, such transitive closures can immediately get very large. There have been considerable rumors in the Linked Data community that such use of *owl:sameAs* is somehow “wrong” with regards to the formal semantics of OWL 1.0. It does seem intuitively that the use of *owl:sameAs* may be the logical equivalent of a bulldozer. Since inference is rarely used on the Linked Data, these possible side-effects have not been noticed. Does this really matter? Is the use of *owl:sameAs* an exploding time-bomb for Linked Data, or a harmless convention? What exactly is the *point* of linking data if nobody is going to draw any conclusions which use the links?

4. FOUR VARIATIONS OF IDENTITY IN LINKED DATA

4.1 Same Thing As But Referentially Opaque

The first case is when the two URIs do refer to the same thing, but all the properties ascribed to one URI are not necessarily accepted by the other. This means that the use of the URI is referentially opaque, which means that one URI cannot be substituted for another (the Principle of Substitution is violated), i.e. the context is intensional. A classic example of this would be the the concept of sodium in DBpedia, which has an *owl:sameAs* link to the concept of sodium in OpenCyc. The OpenCyc ontology says that an element is the set (class) of all pieces of the pure element, so that for example sodium in Cyc has a member which is the lump of pure metallic sodium. On the other hand, sodium as defined by DBpedia is used to also include isotopes, which have different number of neutrons than “standard” sodium. So, one should not state the number of neutrons in DBpedia’s use of sodium, but one *can* with OpenCyc. Therefore, *owl:sameAs* here is in error, as it does not allow mutual substitutivity. Indeed, this use of URIs in an opaque referential context is likely what most uses of *owl:sameAs* actually are for, as it is unlikely that most deployers of Linked Data actually check whether or not *all* the properties and their associated inferences are shared amongst linked data-sets. This property is exceedingly important for Linked Data, as contrary to popular doctrine, it is possible that the Web is full of referentially opaque contexts. The problem is there is no way to get a handle on contexts informally without descending into non-logical reasoning currently.

4.2 Same Thing As But Different Context

In this case, two URIs do refer to the same thing and all properties do hold of both URIs, but that we cannot re-use the URI in a different context. The central intuition here is there are 'forms of reference' appropriate to a context, especially in social contexts. To use an example from Lynn Stein, when at a meeting of the PTA (Parent-Teacher Association) she is Ms. Stein, Rachel's mum, not Professor Stein of MIT. This does not mean that in the PTA meeting Ms. Stein is somehow *not* a professor, but that within that context those properties do not matter. At first, this distinction may not seem directly relevant to linked data, provided we keep 'name' in the social sense distinct from 'identifier' in the Web sense. However, this distinction raises other issues about what kind of 'names' URIs really are and precisely *why* certain properties for linked data are given in the RDF description of a certain URI and others are not.

4.3 Represents

Often identity is conflated with representation. While the term "representation" is often very contentious, its intuitive definition is that, just as a picture of something depicts something, a URI can be for a representation of a thing rather than the thing itself. Intuitively, there seems to be a clear-cut line between that which represents something (the representation) and that which is represented (the referent), sometimes called the relationship between a "sign" and a "signifier." However, the relationship is often not as clear-cut as we would lead ourselves to believe. In fact, in human natural language use-mention confusions are ubiquitous and often useful. For example, often a web-page or an e-mail address are used to refer to a person. Rather than yell at the world to get an education in philosophical logic, it may be better to clarify this relationship. It also might be worth distinguishing between using a representation to refer to the represented, such as using a picture of Berners-Lee to refer to Tim Berners-Lee himself, using something accidentally or contextually to refer to something, a phenomenon called *displaced reference*. The example of using an e-mail box to refer to a person is not an error but rather more a displaced reference.

4.4 Very Similar To

Sometimes its clear that two things are not identical but simply closely related in some manner. This, for example, is the relationship between the district of Paris and the Department of Paris in Cyc. Furthermore, there are often complex, structured, yet hard-to-specify relationships between things, such as the relationship between isotopes and elements, the quantity and a measurement of a quantity, and an image and a facsimile of that image. In web architecture, it is clear there is a close relationship These relationships that are 'very similar to' seem to deserve their own property, but are often currently lumped together in Linked Data under the all-encompassing use of *owl:sameAs*. Most of the more noticeable errors of *owl:sameAs* seem to come from this category, and it is likely that examples such as the relationship of sodium within DBPedia to sodium in OpenCyc are of this kind as well.

5. MOVING FORWARD

Obviously, this list of possible variations of the use of *owl:sameAs* in Linked Data may not be complete. How-

ever, it already illustrates a number of important distinctions for Linked Data. In general, the real problem with the use of URIs as identifiers and *owl:sameAs* is a problem of context and the implicit import of properties. These can all be remedied, and we walk through a case-by-case basis.

5.1 Same Thing As But Referentially Opaque

Surprisingly, most of the time people use *owl:sameAs* they are *accidentally* doing what is sort of an *implicit import* of statements of the subject of the *owl:sameAs* statement. Obviously, to address the weaker identity implied by the referentially opaque use of identity, a weaker version of *owl:sameAs* should be specified that does not import all the properties in a full transitive closure. Somewhat similar predicates already exist in SKOS as *skos:exactMatch* and *skos:closeMatch*, but their use seems rare in Linked Data [5] and they require domains and ranges of SKOS concepts. As most Linked Data does not actually do much inference, one in reality only imports what statements are actually used. So could continue using *owl:sameAs* with a kind of 'importer beware' principle. Informally, it is one thing to link to your URI, but its another thing to believe what you say about it as though you were talking about my URI. Put another way, one should be wary of accepting conclusions *over here* that could have been drawn *over there*, so to speak.

5.2 Same Thing As But Different Context

There is already a notion of context built into RDF, namely *named graphs* [2]. Even though it is not part of the official standard (albeit, snuck into RDF through SPARQL and implemented in almost every tool-set), it is clear that part of the problem with *owl:sameAs* usage on the Semantic Web is that *sameAs* should not always be a statement between two URIs in an unqualified manner, but may be qualified as holding only within a certain named graph. Furthermore, noting the that the use of *owl:sameAs* is somewhat equivalent of an accidental usage of *owl:imports*, although the exact behavior of this construct has only been intuitively (although not formally) specified in OWL Full, although its semantics have been precisely defined in OWL 2.0. These implicit imports should probably either be separated, so that one states at first that two items are identical using the weaker form of identity given above, and then *independently* if one feels strongly about that the two URIs are not referentially opaque, one imports all (or even some of) the associated properties of the "identical" resource. One way this could be done would be to state that if one URI is declared identical to another URI, this relationship is bound to a particular named graph, so that all properties given hold only within that named graph.

5.3 Represents

The use of *owl:sameAs* is already a sort of statement of this kind in the FOAF vocabulary, the *foaf:isPrimaryTopicOf* statement. One possible solution to this problem would be to wrap such a property into some core W3C approved standard. However, the problem is that it is unclear if a strict separation between mention and use is necessary or even desirable. In many contexts, as relevant experience in OpenID deployment shows, using an e-mail as an identifier for a person is often more natural than the URI of a home-page, or even a "non-information resource." What is needed however, is a way to make the distinctions that either conflate

or separate mention and use or on the fly. The use of weak identity statements - and in this case, a “represents” statement - and explicit importing and de-importing of properties within the context of particular named graphs would allow us to do state things like “Within this named graph and only within this named graph, the e-mail address URI is identical to the person and shares their properties” and “Within this other named graph, the e-mail address represents the person, but does not have all the properties of that person.” Although this would complicate the *httpRange-14* finding that demands a strict separation of these two cases (information resources and the implicit “non-information” resources), this approach would probably be more useful for defining heuristic-driven programs in a rigorous manner, who could then define on a case-by-case basis what they meant the URI to mean, and then export those cases to other users and programs in a standard manner.

5.4 Very Similar To

Again, the tempting easy solution is simply to introduce a new predicate for “very similar to.” The SKOS vocabulary has a number of “matching” predicates that are close in meaning to this, ranging from hierarchically structured *skos:broadMatch* and *skos:narrowMatch* to the more suitable *skos:closeMatch*. However, the main issue with these predicates is that again, their use may be a matter of opinion, as someone’s close match may be another person’s identical match. One is also tempted to engage with some sort of “fuzzy” or numerically weighted uncertainty measure between one and zero of identity, but then the real hard questions of where precisely will these real values come from and their relationship to actual probability theory muddies these conceptual waters quickly. It seems that beneath this apparently simple property is likely a whole family of heterogeneous and semi-structured identity relationships that should be studied more carefully and empirically observed before any hasty judgments are made. We may have to accept that some terms cannot be very formally defined. It would likely be far too complex to pursue this path using fuzzy logic, although it would be interesting to see how machine-learning techniques can help determine similarity.

6. CONCLUSION

Obviously, the issue of how to express relationships of identity on Linked Data is more complex than just applying *owl:sameAs*. At the same time, a more nuanced approach that fulfills the current four additional possible uses of identity beyond *owl:sameAs* would be a useful step for the Linked Data community. However, what becomes clear even after a cursory glance at possible solutions is that solving the issue of identity in Linked Data may require a certain refactoring of some core constructs of RDF, including relating identity to named graphs and to the use of imports on the Semantic Web.

It is possible to do empirical studies of exactly *how* people use *owl:sameAs* in the wild. Examples of *owl:sameAs* can be taken from the Linked Data Web in the wild in order to determine how experimentally robust these distinctions are would be, i.e. do people actually use *owl:sameAs* in the four ways that are outlined above, and are there more possible ways that we are not aware of? In fact, even the ability to recognize these kinds of distinctions may vary quite wildly by background and training. Lastly, if a number of em-

pirical distinctions between identity links that are currently conflated by *owl:sameAs* can be made in a robust manner, then there is considerable formal semantic work to be done. Giving the Linked Data community well-defined (both formally and informally) predicates should be done even when one does think of the properties given to URIs as absolute truths given by Linked Data publishers or W3C specifications, but as functions of their actual use. The (ab)use of *owl:sameAs* in Linked Data is not a threat, it’s an opportunity.

One way forward would be for the W3C to define a core vocabulary for identity management, either as part of the next version of RDF or as a separate vocabulary. If it was defined as part of the core RDF(S) vocabulary using the suggestions listed above, some sort of formal semantics that incorporated named graphs would be expected to be part of the whole, even if not all identity relationships could be formalized semantically.

7. REFERENCES

- [1] C. Bizer, R. Cygniak, and T. Heath. How to publish Linked Data on the Web, 2007. <http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/> (Last accessed on May 28th 2008).
- [2] J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs. *Journal of Web Semantics*, 4(3):247–267, 2005.
- [3] I. Jacobs and N. Walsh. Architecture of the World Wide Web. Technical report, W3C, 2004. <http://www.w3.org/TR/webarch/> (Last accessed Oct 12th 2008).
- [4] A. Jaffri, H. Glaser, and I. Millard. Managing URI synonymy to enable consistent reference on the Semantic Web. In *Proceedings of the Workshop on Identity, Reference, and the Web (IRSW) at ESWC2008*, 2008.
- [5] A. Miles and S. Bechhofer. SKOS Simple Knowledge Organization System reference. W3c recommendation, W3C, 2008. <http://www.w3.org/TR/skos-reference/>.
- [6] C. Welty, M. Smith, and D. McGuinness. OWL Web Ontology Language Guide. Recommendation, W3C, 2004. <http://www.w3.org/TR/2004/REC-owl-guide-20040210>.