RDF: Back to the Graph

Peter F. Patel-Schneider
Bell Labs Research
Alcatel-Lucent
http://ect.bell-labs.com/who/pfps

April 8, 2010

Abstract

The two natures of RDF—data structuring via RDF graphs versus knowledge representation via the RDF semantics—are in conflict, particularly when considering same-syntax extensions to RDF in the higher levels of the Semantic Web. This conflict can only be resolved by abandoning either the use of RDF to structure all syntax or the use of RDF to represent all semantics. If RDF is to serve as firm foundation for the Semantic Web then it is best to abandon both these ideas, and form the Semantic Web in a way that is common to other representation families.

Since the respecification of RDF in 2004 [Manola and Miller, 2004], RDF can be viewed in two ways. One view of RDF treats it as a data structuring language. In this, the graph view, what counts is the graph encoded by a particular RDF document [Klyne and Carroll, 2004] (in any of the surface syntaxes for RDF). The other view of RDF treats it as a logic for representing knowledge. In this, the semantic view, the RDF graph is just a means to an end. The meaning of a particular RDF document is only mediated by the RDF graph encoded in the document. Instead what counts is the set of RDF interpretations that satisfy this graph [Hayes, 2004].

This is not to say that the RDF graph is unimportant in the semantic view of RDF. The semantic view uses RDF graphs to state "facts," after all, so RDF graphs delimit what can be said, and what can be entailed.

The problem is that the two views of RDF produce a split personality for RDF, resulting in a divergence in what users of RDF documents conclude from an RDF document. The two views are not so divergent at the RDF level. RDF is so inexpressive that very little can be inferred beyond graph matching. Nonetheless, even at the RDF level there is the issue of how to treat blank nodes, where the semantically correct treatment results in more inferences than would be performed by a simple implementation. The two views are somewhat more divergent when the RDFS extensions to RDF [Brinkley and Guha, 2004] are considered, as there are considerably more inferences in RDFS, including inferences triggered by augmenting the ontology definition part of RDFS, such as by creating subproperties of `rdfs:subClass`.

Because of problems like these, quite a few implementations of RDFS are incomplete, particularly ones that are designed to rapidly handle large amounts of information in RDFS [Weaver and Hendler, 2009; Urbani *et al.*, 2009]. When handling large amounts of information, then, it is convenient to treat RDFS as something less than the semantic specification states, so that the use of RDFS diverges from the RDF semantic specification.

This divergence comes to the fore in OWL [OWL, 2009], where there are two semantics, one for OWL 2 Full [Schneider, 2009], which is an extension to the RDFS semantics, and one for OWL 2 DL and other OWL profiles [Motik *et al.*, 2009], which, although quite compatible with the RDF semantics, has a totally different basis. The major reason for the divergence is that there are effective reasoners for the DL semantics whereas there are none for the RDF-compatible semantics. In the DL view, RDF is a data structuring language for OWL 2 syntactic constructs, albeit one where most RDF triples encode facts. just as they do in the RDF semantics. Again, here the use of RDF diverges from the RDF semantic specification.

I propose that these divergences be resolved by going "back to the graph", i.e., treating RDF simply as a data structuring language, much as was done in the initial specification of RDF [Lassila and Swick, 1999].

It may seem that this is a step backward, as the respecification of RDF was done precisely to firm up just what should be done with RDF graphs. The problem is that it is not really possible to treat RDF graphs as both a syntactic and semantic foundation for the Semantic Web. Even at the RDFS level, there is a desire to ignore parts of the RDF semantics. At the OWL level, effective reasoning requires a different semantic treatment. At higher levels, such as a Semantic Web language that covers all of first-order logic, a different semantic treatment is required to avoid paradoxes [Patel-Schneider, 2005].

Therefore something needs to be done, and it seems to me that the simplest thing to do is to turn RDF graphs into a neutral data structure so they can be used as the higher levels of the stack desires.

This would make is quite palatable for higher levels to not cover all RDF graphs; their syntax can be defined as a subset of all RDF graphs, perhaps avoiding some constructs that appear to be higher-order (like creating sub-properties of rdfs:subClass).

Independently, or synergistically, as desired, the higher levels may choose to interpret what have in the past been considered to be the RDF logical relations (rdf:type, rdfs:subClassOf, ...) as simply non-logical relations, no different from other domain relations. Of course, doing this may cause interoperability issues, but this may not be considered to be a problem.

Further, higher levels could choose to treat parts of RDF graph not as facts, but only as some complex construct. This would be useful, for example, to encode disjunction or negation in RDF graphs, so that the pieces of the disjunction or negation (reified triples?) are not facts. In this way the difficulties [Patel-Schneider, 2005] of building OWL [OWL, 2009] on top of RDF would be eliminated, and more expressive layers could be built without any problems with paradoxes.

Higher levels could even create their own syntax for their constructs, and not use RDF graphs at all. So, for example, a future version of OWL could use an XML syntax for its class axioms and not have any RDF graph syntax for them at all.

Perhaps the biggest advantage, though, in going "back to the graph" is that it allows easy extensions to the data-structuring capabilities of RDF itself. Several such extensions have already been proposed, or are in use, such as named graphs [Carroll *et al.*, 2005] and quads [MacGregor and Ko, 2003]. When RDF is treated as a data structuring language, these extensions can be done without having to worry about devising a model-theoretic semantics for them. For example, embedded graphs could be added to RDF simply by allowing graphs as the objects of triples. Named graphs and quads, or even arbitrary triples, could be similarly added. Of course, there would have to be consensus or standardization of the surface syntax for these extensions to support interoperability of these data structures.

Applications would treat these extensions as data structures, imparting their own meaning on them, whether for provenance, disjunction, or modality. When adequate consensus as to the meaning of particular structures has arisen, a higher, representational level of the Semantic Web could be standardized. Existing higher levels, such as RDFS, OWL and N3, could quickly exploit these new data structures, providing a better syntactic basis, at least for OWL.

If the current way of using RDF in the Semantic Web is to coexist with this new way, there would need to be a way of distinguishing between new- and old-style RDF documents. MIME types could be used to make this distinction, with new-style RDF documents getting new MIME types. It would probably be possible to use old-style RDF document in the new style, by ignoring the logical consequences of certain multi-triple constructs in these documents and just massaging the data structures that the triples encode into a new-style data structure.

Once the triple straitjacket is removed from formalisms such as OWL and N3 it would also be possible to move completely away from RDF for parts of these formalisms. Even a tuple version of RDF can be problematic for encoding complex syntax, because of the inability in RDF to mandate that certain constructs look like tree structures. It might be possible to add this facility to RDF, but it also might be better to use XML directly to encode, for example, complex OWL classes or N3 disjunctions. This would form the Semantic Web in a way very similar to compatible logic extensions are often formed, with extensions (such as first-order logic, higher-order logics, and modal logics) extending and modifying the syntax of the basic logic (propositional logic) and providing extended semantics that remain compatible with the base.

It is important to note that even under this proposal, RDF itself could remain a way of encoding simple facts, just as it is used for now. This move back to RDF as solely a data structuring language, as embodied in the RDF graph, possibly extended, and removing RDF's stranglehold on both the syntax and semantics of the Semantic Web would provide a firm foundation for the Semantic Web, on which well-specified extensions could be constructed without having to worry (as much) about the entire edifice collapsing.

References

[Brinkley and Guha, 2004] Dan Brinkley and R. V. Guha. RDF vocabulary description language 1.0: RDF schema. W3C Recommendation, `http://www.w3.org/TR/rdf-schema`, 10 February 2004.

[Carroll *et al.*, 2005] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs, provenance and trust. In *Proceedings of the 14th World Wide Web Conference*, Japan, May 2005.

[Hayes, 2004] Patrick Hayes. RDF semantics. W3C Recommendation, `http://www.w3.org/TR/rdf-mt/`, 10 February 2004.

[ISWC 2009, 2009] *Proceedings of the Eighth International Semantic Web Conference*. Springer, November 2009.

[Klyne and Carroll, 2004] Resource Description Framework (RDF): Concepts and abstract syntax. W3C Recommendation, `http://www.w3.org/TR/rdf-schema`, 10 February 2004.

[Lassila and Swick, 1999] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF): Model and syntax specification. W3C Recommendation, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222, 22 February 1999.

[MacGregor and Ko, 2003] Robert MacGregor and I.-Y. Ko. Representing contextualized data using semantic web tools. In *Practical and Scalable Semantic Systems (ISWC 2003 workshop)*, 2003.

[Manola and Miller, 2004] Frank Manola and Eric Miller. RDF primer. W3C Recommendation, http://www.w3.org/TR/rdf-primer, 10 February 2004.

[Motik *et al.*, 2009] Boris Motik, Peter F. Patel-Schneider, and Bernardo Cuenca Grau. OWL 2 Web Ontology Language: Direct semantics. W3C Recommendation, http://www.w3.org/TR/owl2-direct-semantics, 27 October 2009.

[OWL, 2009] OWL 2 Web Ontology Language: Document overview. W3C Recommendation, http://www.w3.org/TR/owl2-overview, 27 October 2009.

[Patel-Schneider, 2005] Peter F. Patel-Schneider. Building the semantic web tower from RDF straw. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. International Joint Committee on Artificial Intelligence, August 2005.

[Schneider, 2009] Michael Schneider. OWL 2 Web Ontology Language: RDF-based semantics. W3C Recommendation, 27 October 2009, http://www.w3.org/TR/owl2-rdf-based-semantics, 2009.

[Urbani *et al.*, 2009] Jacopo Urbani, Spyros Kotoulas, Eyal Oren, and Frank van Harmelen. Scalable distributed reasoning using mapreduce. In ISWC 2009 [2009].

[Weaver and Hendler, 2009] Jesse Weaver and James Hendler. Parallel materialization of the finite RDFS closure for hundreds of millions of triples. In ISWC 2009 [2009].