RDF Needs Annotations

Nuno Lopes* Antoine Zimmermann* Aidan Hogan* Gergely Lukácsy†
Axel Polleres* Umberto Straccia‡ Stefan Decker*

Abstract

While the current mechanism of reification in RDF is without semantics and widely considered inappropriate and cumbersome, some form of reification – speaking about triples themselves – is needed in RDF for many reasonable applications: in particular, reification allows for enhancing triples with annotations relating to provenance, spatio-temporal validity, degrees of trust, fuzzy values and/or other contextual information. In this position paper, we argue that – besides resolving the issue of how to syntactically represent reification in the future (i.e., whether to stick with the current reification mechanism or standardise a different mechanism such as Named Graphs) – it is time to agree on certain core annotations that are widely needed. We summarise existing work and provide a possible direction towards handling reification by means of a general annotation framework that can be instantiated for those major use cases we currently see arising.

1 A Need for RDF Annotations

In this paper, we motivate and discuss the syntactic representation and semantic interpretation of generic RDF annotations. In particular, we propose the introduction of a generic semantic framework to represent annotations in RDF. Although these annotations can be used to represent arbitrary domains, we suggest a standard representation for some common domains such as time, space and provenance.

Syntactically, RDF Annotations can be considered an extension of RDF by attaching to each triple, or to sets of triples, an annotation value. Thus, instead of the RDF triple composed of subject, predicate and object, we may refer to an RDF quad composed of the previous elements plus an annotation value.

Currently, reification is the only standardised mechanism to add meta-statements about RDF triples; however, it remains without a semantic specification meaning that, e.g., reified statements are not affected by RDFS or OWL inferences. Despite the lack of standard semantics, several current activities in W3C's Semantic Web activities would warrant an extension that enables one to annotate statements:

- Web data represented in RDF will – in parts – be dynamic, be it the affiliation of a person changing (represented, e.g., by a `foaf:workplaceHomepage` triple), or more rapidly changing sensor data such as current temperature readings published online[1]. These use cases warrant annotations that talk about the *temporal validity* of triples.
- Similarly, relying on the sensor example from before, a temperature reading may be valid for a certain location or area only. Likewise, semantically annotated observations made in a blog may be local. These examples warrant annotation that talk about *geo-spatial validity* of triples.
- As discussed in the report [8] of the W3C's Incubator group on "Uncertainty Reasoning for the World Wide Web"[2] there are use cases where agents may want to represent and express vagueness of state-

*Digital Enterprise Research Institute, National University of Ireland, Galway. firstname.lastname@deri.org
†Cisco Systems Inc. glukacsy@cisco.com
‡Istituto di Scienza e Tecnologie dell'Informazione (ISTI - CNR), Pisa, Italy umberto.straccia@isti.cnr.it

[1] W3C's Semantic Sensor Network Incubator Group is currently taking first steps towards common formats for publishing sensor data, cf. `http://www.w3.org/2005/Incubator/ssn/charter`

[2] `http://www.w3.org/2005/Incubator/urw3/`

ments: e.g., in the form of *fuzzy values*. Such use cases warrant annotations that attach fuzzy values to triples.

Other domains of interest requiring annotations for RDF triples include modelling *provenance or trust* – for example, many systems operating over heterogeneous Web data sensibly track provenance of triples. Also, as opposed to the aforementioned examples, triple annotations may not necessarily be published online: instead, consumer or publishing systems may store annotations internally for internal processes.

However, despite the clear motivation for annotations in RDF, there are some open questions relating to (i) syntax: e.g., "how can annotations be concisely represented in a manner compatible with legacy RDF systems?"; and (ii) semantics: e.g., "how should annotations be handled during RDFS/OWL inferencing, and what annotations should be attached to inferred triples?". Having reinforced the need for some form of reification in RDF – particularly in light of various W3C activities – we continue by discussing syntactic representations of annotations, and then present a concrete proposal to handle the semantics of annotations in a generic way.

2 Annotations – Syntactic Representation

In this paper, we wish to focus on a semantic framework for annotations in RDF and motivation thereof; however, in this section we provide an important – albeit brief – overview of possible approaches for syntactically representing RDF annotations.

The first and only standard way to represent annotations of triples in RDF is reification; for example, taking the use case of stating one's affiliation homepage alluded to in Section 1, a person could use RDF reification and the time-interval vocabulary introduced in [3] to specify a temporal validity for their affiliation as follows:

```
_:stmt rdf:subject :me ; rdf:predicate foaf:workplaceHomepage ;
       rdf:object <http://coolCompanyA.example.org/> ;
       tmp:interval [ tmp:initial "2008-04-01T09:00:00Z"^^xsd:dateTimeStamp ;
                      tmp:final "2009-11-11T17:00:00Z"^^xsd:dateTimeStamp ] .
```

Alternatively, annotations can be attached to named graphs [1], which can hold sets of statements. Graphs can be named using a non-standard "quad-centric" syntax, such as N-Quads[3], TriG[4] or TriX[5]. The following is an equivalent N-Quads representation for the above example:

```
:me foaf:workplaceHomepage <http://coolCompanyA.example.org/> _:c .
_:c tmp:interval [ tmp:initial "2008-04-01T09:00:00Z"^^xsd:dateTimeStamp ;
                   tmp:final "2009-11-11T17:00:00Z"^^xsd:dateTimeStamp ] .
```

However, the different representations of named graphs share similar problems: (i) none of those syntaxes is "rubber-stamped" by a W3C standard at the moment, and (ii) many legacy RDF systems would not be forwards-compatible with such quad-centric syntax and data representation. Whilst named graph representations would probably allow for more concise annotation encoding than reification, the latter remains the only standard syntax and the only syntax with which all RDF systems are compatible.

Thus, the core question remains: how can a concise representation of annotations be sensibly integrated into the core of RDF, and how will such annotated RDF be integrated with the billions of triples of legacy non-annotated data?

3 Annotations – Towards A General Semantic Framework

Acknowledging that the problem of an agreed syntactic representation has to be solved first, in the rest of this paper, we will step aside from such considerations focusing instead on the semantic implications

[3] http://sw.deri.org/2008/07/n-quads/

[4] http://www4.wiwiss.fu-berlin.de/bizer/TriG/Spec/

[5] http://sw.nokia.com/trix/TriX.html

of annotations. Particularly, whether named graphs become standardised or reification is finally given a semantics, RDF with annotations needs to play with its neighbour standards, namely RDFS, OWL and SPARQL.

For instance, from the statement:

```
foaf:workplaceHomepage rdfs:range foaf:Document
```

in the FOAF ontology, we would expect in our above example (using reification syntax here) that something like:

```
_:stmt2 rdf:subject <http://coolCompanyA.example.org/> ;
        rdf:predicate rdf:type ; rdf:object foaf:Document ;
        tmp:interval [ tmp:initial "2008-04-01T09:00:00Z"^^xsd:dateTimeStamp ;
                       tmp:final "2009-11-11T17:00:00Z"^^xsd:dateTimeStamp ] .
```

also holds. This means that `http://coolCompanyA.example.org/` is necessarily a document as long as it is the workplace homepage of `:me`.

The addition of annotations to triples may lead to the introduction of inconsistencies in RDF, something previously non-existent. However, and as presented in [15], by defining the annotation domain as a lattice, consistency is guaranteed.

Different approaches for annotating triples with values and giving a semantics to such annotations have been proposed for specific domains, such as time [4, 9, 14], degree of truth [12], trust [5] and provenance [2]. In [13], we have presented an approach that is generic enough to encompass all of these approaches. This is achieved by allowing the use of generic annotation labels and by the definition of operations that relate annotation elements belonging to a specific domain. The annotation domain is represented as a lattice $\langle L, \preceq \rangle$, where L specifies the annotation terms and $\preceq$ is a partial order relating elements of L. For each defined domain, the annotation labels are elements of L and the partial order is used to represent redundant (entailed) information.

The extension of RDFS rules to incorporate annotations allows for the inference of triples whose annotations are determined by the provided lattice operations over the annotations of the input triples. Tying in with the previous example, the inference rule `rdfs3` (which supports `rdfs:range`) can support annotations as follows (taken from [13, Table 2]):

$$(b) \quad \frac{(A, \mathsf{rdfs{:}range}, B) : \lambda_1, (X, A, Y) : \lambda_2}{(Y, \mathsf{rdf{:}type}, B) : \lambda_1 \otimes \lambda_2}$$

where triples – possibly reified – are represented between brackets, λ_1 and λ_2 are the annotation values for the specified triples, and $\otimes$ represents the *minimum* operation for the given domain. In our example, where the annotations are temporal intervals, the $\otimes$ operation corresponds to the *set intersection* operation. Although the semantics presented in [13] are generic and allow one to define the representation for any domain, implementations should use lattice operations required by the specific domain. RDFS semantics is a particular case of the annotation extension, so our approach is backward compatible with the current standards.

Moreover, in order to complete our annotated RDF/RDFS framework, we consider extensions to the SPARQL query language and to OWL. Considering that annotations are assigned to triples, an extension of the framework to OWL would be tied to the RDF-based semantics of OWL [10].

The SPARQL query language already supports querying named graphs using the `FROM NAMED` construct. Thus depending on the representation for the annotations (as presented in Section 2), the extension could be minimal. For the general case, extending SPARQL relies on considering the extended syntax for triples and – as a safe approach – defining the set of annotation variables as disjoint from the set of triple pattern variables.

Any of the previously discussed annotation domains (temporal, spatial, fuzzy, trust, provenance) can be encoded in the framework of [13], and different annotation domains are arbitrarily combinable. The proposed approach remains backwards compatible, allowing to incorporate non-annotated RDF data in the reasoning process. Our semantic definitions are well-founded, being inspired by earlier ideas such as annotated logic

programs [7] or TRIPLE [11]. However, we consider this only a starting point. The main purpose of the present paper is to argue for agreement on the semantics of some core annotations that are widely used and the absence of which – in our opinion – hampers the usability of RDF.

4 Annotations – Instantiation of Specific Domains

Although unrestricted annotations values allow for the introduction of new annotation domains, suggesting a representation for the most currently used domains, enables a faster adaptation of RDF annotations. Next we present a possible representation for some domains:

4.1 Provenance

Provenance is perhaps the most important information that could be attached to a triple. In particular, systems which operate over data collected from a large number of unvetted sources must track triple provenance in order to justify, rank, isolate, or even negate the contribution of independent – possibly nonsensical – sources.

4.2 Temporal

For the temporal domain, we consider that any triple can be valid over different, non-overlapping periods of time. A possible representation for the temporal annotation uses a set of disjoint time intervals that represent each period of time when the triple was valid. Each time interval is defined by the start and end point, represented using a *xsd:dateTimeStamp* literal, as can be seen in the examples in the previous sections.

4.3 Fuzzy

For the fuzzy domain, the standard is to consider the annotation as a value between 0 and 1, as suggested in [12]. Although the annotation values for these domains are different, conceptually they can be handled similarly, as a single decimal value for the annotation. For instance, the following annotated triple says that `:me` is interested in the Semantic Web to the degree 0.8.

```
_:stmt3 rdf:subject :me ; rdf:predicate foaf:topic_interest ;
        rdf:object dbpedia:Semantic_Web ;
        fuzzy:value  "0.8"^^xsd:decimal .
```

4.4 Trust

The trust and fuzzy domains, although having different meanings, can be represented in a similar manner. For instance, the Hoonoh [6] approach models trust using a custom reification vocabulary together with numerical values:

```
_:stmt4 a hoohnoh:ExpertiseRelationship ; hoonoh:From :you ;
        hoonoh:toTopic dbpedia:Semantic_Web ;
        hoonoh:value  "0.8"^^xsd:decimal .
```

This can be treated as above. Slightly differing, Hartig [5] suggests to use a trust value from the interval $[-1, 1]$, where values tending to -1 represent a higher degree of *disbelief* in the triple and values close to 1 represent a higher degree of *belief* in the triple. A value of, or close to, 0 represents *uncertainty* about the trustworthiness of the triple – here, trustworthiness could relate to the provenance domain previously discussed above.

5 Conclusions

We are convinced that there is a need for certain annotations in RDF; time, spatial, provenance, fuzzy, trust and provenance annotations seem to be the most obvious. Thus, we have to agree (i) upon how to syntactically represent them to make these annotations exchangeable – triples alone are not enough – and (ii) which semantics such annotations carry. In [13], we have put a proposal on the table that sketches how annotation domains that form a lattice can be extended to interplay with standards up the layer cake – namely, RDFS and SPARQL. An extension of the framework to OWL tied to the RDF-based semantics of OWL is on our agenda.

References

[1] Jeremy J. Carroll, Christian Bizer, Patrick J. Hayes, and Patrick Stickler. Named graphs. *J. Web Sem.*, 3(4):247–267, 2005.

[2] Renata Queiroz Dividino, Sergej Sizov, Steffen Staab, and Bernhard Schueler. Querying for provenance, trust, uncertainty and other meta knowledge in RDF. *J. Web Sem.*, 7(3):204–219, 2009.

[3] Claudio Gutiérrez, Carlos A. Hurtado, and Alejandro A. Vaisman. Temporal RDF. In *Proc. of ESWC 2005*, volume 3532, pages 93–107. Springer, 2005.

[4] Claudio Gutierrez, Carlos A. Hurtado, and Alejandro A. Vaisman. Introducing Time into RDF. *IEEE Trans. Knowl. Data Eng.*, 19(2):207–218, 2007.

[5] Olaf Hartig. Querying Trust in RDF Data with tSPARQL. In *Proc. of ESWC 2009*, volume 5554 of *LNCS*, pages 5–20. Springer, 2009.

[6] Tom Heath and Enrico Motta. The hoonoh ontology for describing trust relationships in information seeking. In *Proc. of PICKME 2008*, 2008.

[7] Michael Kifer and V.S. Subrahmanian. Theory of Generalized Annotated Logic Programming and its Applications. *Journal of Logic Programming*, 12:335–367, 1992.

[8] Kenneth J. Laskey, Kathryn B. Laskey, Paulo C. G. Costa, Mieczyslaw M. Kokar, Trevor Martin, and Thomas Lukasiewicz. Uncertainty Reasoning for the World Wide Web. Technical report, W3C, March 2008. W3C Incubator Group Report, `http://www.w3.org/2005/Incubator/urw3/XGR-urw3/`.

[9] Andrea Pugliese, Octavian Udrea, and V. S. Subrahmanian. Scaling RDF with Time. In *Proc. of 17th WWW Conference*, pages 605–614, New York, NY, USA, 2008. ACM.

[10] Michael Schneider. OWL 2 Web Ontology Language RDF-Based Semantics W3C Recommendation 27 October 2009. Technical report, 2009.

[11] Michael Sintek and Stefan Decker. Triple - a query, inference, and transformation language for the semantic web. In *Proc. of ISWC 2002*, pages 364–378, 2002.

[12] Umberto Straccia. A Minimal Deductive System for General Fuzzy RDF. In *Proc. of RR 2009*, number 5837 in LNCS, pages 166–181. Springer, 2009.

[13] Umberto Straccia, Nuno Lopes, Gergely Lukácsy, and Axel Polleres. A General Framework for Representing and Reasoning with Annotated Semantic Web Data. In *Proc. of 24th AAAI Conference*, 2010. Accepted for publication, tech. report available at `http://www.deri.ie/fileadmin/documents/DERI-TR-2010-03-29.pdf`.

[14] Jonas Tappolet and Abraham Bernstein. Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. In *Proc. of ESWC 2009*, volume 5554, pages 308–322. Springer, 2009.

[15] Octavian Udrea, Diego Reforgiato Recupero, and V. S. Subrahmanian. Annotated RDF. In *Proc. of ESWC 2006*, number 4011 in LNCS, pages 487–501. Springer, 2006.