

## Position Statement: The RDF\* and SPARQL\* Approach to Annotate Statements in RDF and to Reconcile RDF and Property Graphs

Posted on January 10, 2019 by Olaf Hartig

1 people like this post.

[Rapportera inlägget / Report this post](#)

*With this post I am providing a position statement for the upcoming [W3C Workshop on Web Standardization for Graph Data](#).*

The lack of a convenient way to annotate RDF triples and to query such annotations has been a long standing issue for RDF. Such annotations are a native feature in other contemporary graph data models (e.g., edge properties in the Property Graph model) and there exist a number of popular use cases, including the annotation of statements with certainty scores, weights, temporal restrictions, and provenance information. To mitigate the inherent lack of a native support for such annotations in the purely triple-based data model of RDF, there exist several proposals to capture such annotations in the RDF context (e.g., RDF reification as proposed in the RDF specifications, singleton properties, single-triple named graphs). However, these proposals have a number of [shortcomings](#) and none of them has yet been adopted as a (de facto) standard.

We are proposing an alternative approach that is based on nesting of RDF triples and of query patterns. This approach has already attracted interest not only in the RDF and Semantic Web research community (as indicated by [some blog posts](#) and by winning the [People's Choice Best Poster Award at ISWC 2017](#)) but also among RDF system vendors. In fact, the approach is already supported in two commercial RDF graph database systems ([Blazegraph](#) and [AnzoGraph](#)) and in an [extension of the popular Open Source framework Apache Jena](#). Important properties of the approach are that



it allows for a compact representation of data and queries,

it is backwards-compatible with the aforementioned existing approaches,

it can serve naturally as a foundation for achieving interoperability between the RDF and the Property Graphs world, and

it can be employed as a common conceptual framework to capture more specific annotation-related extensions of RDF and SPARQL (such as temporal or probabilistic extensions).

The goal of this position statement is to bring the approach to the attention of the workshop attendees and to put on the workshop agenda a discussion regarding standardization opportunities for this approach.

In the remainder of this position statement we outline the approach and elaborate more on its properties.

## Overview of the Approach

The basis of the proposed approach is to extend RDF with a notion of nested triples. More precisely, with this extension, called **RDF\***, any triple that represents metadata about another triple may directly contain this other triple as its subject or its object. For instance, suppose we want to capture a statement indicating the age of Bob together with the metadata fact that we are 90% certain about this statement. RDF\* allows us to represent both the data and the metadata by using a nested triple as follows.

```
<<:bob foaf:age 23>> ex:certainty 0.9 .
```

Notice that we write the nested triple using an extension of the [RDF Turtle syntax](#) that captures the notion of nested triples by enclosing any embedded triple using the strings '<<' and '>>'. This extended syntax is called **Turtle\*** and it is specified in Section 3.3 of our [technical report](#).

Given the outlined notion of RDF\* which supports (arbitrarily deep) nesting of triples, the crux of the proposed approach is to extend the RDF query language SPARQL accordingly. That is, in the extended query language, called **SPARQL\***, triple patterns may also be nested, which gives users a query syntax in which accessing specific metadata about a triple is just a matter of mentioning the triple in the subject (or object) position of a metadata-related triple pattern. For instance, by adopting the aforementioned syntax for nesting, we may query for all age-statements and their respective certainty as follows (prefix declarations omitted).

```
SELECT ?p ?a ?c WHERE {  
  <<?p foaf:age ?a>> ex:certainty ?c .  
}
```

Notice that the query is represented in a very compact form; in particular, in contrast to the corresponding queries for other proposals (e.g., RDF reification, singleton properties), this compact syntax does not require users to write verbose patterns or other constructs whose only purpose is to match artifacts that these proposals introduce to establish the relationship between a triple and the metadata about it.

In addition to nested triple patterns, SPARQL\* introduces a new type of BIND clauses that allows us to express the example query in the following, semantically equivalent form.

```
SELECT ?p ?a ?c WHERE {  
  BIND (<<?p foaf:age ?a>> AS ?t)  
  ?t ex:certainty ?c .  
}
```

The latter example also highlights the fact that in SPARQL\*, variables in query results may be bound not only to IRIs, literals, or blank nodes, but also to full RDF\* triples. For a detailed formalization of SPARQL\*, including the complete extension of the full W3C specification of SPARQL, refer to Sections 4-5 of the [technical report](#).

## Properties of the Approach

We emphasize three orthogonal perspectives on the proposed approach:

1. On one hand, RDF\* and SPARQL\* may be understood—and used—simply as **syntactic sugar on top of RDF and SPARQL**. That is, any RDF\*-specific syntax such as Turtle\* may be parsed directly into plain RDF data that uses RDF reification or any of the other approaches to annotate statements in RDF. Likewise, SPARQL\* queries may be rewritten into ordinary SPARQL queries. Based on such conversions, RDF\* and SPARQL\* may be supported easily by implementing wrappers on top of existing RDF triple stores. Then, users can query either RDF\* data or RDF data with other forms of statement annotations, both by using SPARQL\*. The formal mappings necessary as a foundation of such wrapper-based implementations have already been [defined and studied](#), and there exists an initial set of [conversion tools](#).
2. On the other hand, the proposal may also be conceived of as a **new abstract data model** in its own right. As such, it may be implemented by developing techniques to execute SPARQL\* queries directly on a physical storage model that is designed to support RDF\* natively. The formal foundations of this perspective exist; that is, we have defined [the RDF\\* data model and a formal semantics of SPARQL\\*](#). Moreover, the RDF graph database systems [Blazegraph](#) and [AnzoGraph](#) provide native support for RDF\* and SPARQL\*, and so does the aforementioned [extension of Apache Jena](#).
3. A third perspective on the approach is that it presents a step towards closing the gap between the RDF and the Property Graphs world. That is, by extending RDF and SPARQL with a feature that is similar to the notion of edge properties in Property Graphs, the approach may serve as an **abstraction for integrating RDF data and Property Graphs**. In fact, in addition to the aforementioned RDF\*-to-RDF mappings, there already exist formal definitions of [direct mappings from RDF\\* to Property Graphs and vice versa](#), and these mappings have been implemented in [conversion tools](#).

---

← [Lightweight summary of our paper “Semantics and Complexity of GraphQL”](#)

## Leave a Reply

Your email address will not be published. Required fields are marked \*

Comment

Name \*

Email \*

Website

**Post Comment**



## Sök

## Senaste inläggen

[Position Statement: The RDF\\* and SPARQL\\* Approach to Annotate Statements in RDF and to Reconcile RDF and Property Graphs](#)

[Lightweight summary of our paper “Semantics and Complexity of GraphQL”](#)

[A Foundation for Comparing Linked Data Fragment Interfaces](#)

[Dagstuhl seminar on Federated Semantic Data Management](#)

## Senaste kommentarerna

## Arkiv

[January 2019](#)

[August 2018](#)

[October 2017](#)

## Kategorier

[Allmän](#)

[Event Report](#)

[Proposals](#)

[Research Results](#)

## Meta

[Log in](#)

[Entries RSS](#)

[Comments RSS](#)

[WordPress.org](#)

## Linköpings universitet

581 83 LINKÖPING

Tel: 013-28 10 00

Fax: 013-14 94 03

[Kontakta LiU](#) | [Kartor](#)

[Om webbplatsen](#)

Sidansvarig: [webmaster@liu.se](mailto:webmaster@liu.se)

## Organisation

[Om LiU](#)

[Filosofiska fakulteten](#)

[Medicinska fakulteten](#)

[Tekniska högskolan](#)

[Utbildningsvetenskap](#)

[Institutioner](#)

[Universitetsförvaltningen](#)

## Linköpings universitet

Ett av Sveriges större lärosäten med en stark tradition av samverkan över gränser och förnyelse av forskning och utbildning.