RDF-star use cases from Amazon Neptune

Contact: Ora Lassila <ora@amazon.com>

The Amazon Neptune graph database [1] supports both RDF and LPGs. The Neptune team is working towards broad interoperability between the two graph models, and sees RDF-star as an important step towards that goal. This document outlines a couple of use cases we believe are necessary towards our interoperability goal.

Issues in graph interoperability

There are several obstacles to full graph interoperability. Many of these are described in a recent paper we have published [2]. The use cases in this document, however, only address the following essential, practical problem: RDF semantics (as well as RDF-star semantics, by extension) stipulate that a triple <s, p, o> is unique (i.e., there is only one, for any combination of s, p, and o). In LPGs, however, no such restriction exists for edges, and in fact it is typical in LPG models to take advantage of this.

Use case: Provenance and lineage

Given a statement (e.g., “Bob is married to Alice”), expressed in Turtle as:

```
:Bob :isMarriedTo :Alice .
```

we may want to annotate it with the source(s) of this information. So let’s say that the New York Times claims that Bob and Alice got married in 2020, but the Washington Post claims this happened in 2019. Given these sources, the RDF-star representation could be:

```
:Bob :isMarriedTo :Alice .
<< :Bob :isMarriedTo :Alice >> :since 2020 ;
                               :source :NYTimes .
<< :Bob :isMarriedTo :Alice >> :since 2021 ;
                               :source :WashingtonPost .
```

On the first line, we have made two statements about the original statement. On the second line we make two different statements about the original statement, but since statements are unique, we effectively make all these statements about the same original statement, and afterwards can no longer distinguish between information that came from the two newspapers.

An alternative representation would be to make two annotation statements about the original statement (concerning the year when Bob and Alice got married), and then annotate those statements with their respective sources.

```
:Bob :isMarriedTo :Alice .
<< << :Bob :isMarriedTo :Alice >> :since 2020 >> :source :NYTimes .
<< << :Bob :isMarriedTo :Alice >> :since 2019 >> :source :WashingtonPost .
```

Now there is no information loss. This is of course a perfectly valid approach, but it does not match the typical approach when using LPGs. Note also that, while the example above captures the correct semantics, it is awkward and potentially leads to problems when writing queries. Also, it does not solve the problem if both the NY Times and the Washington Post agree on the date of Bob and Alice’s marriage. Thus, we offer the original example as an illustration of a shortcoming in the RDF-star model.

Use case: Digital twin

Similarly to the provenance and lineage use case above, this one uses edge properties. We want to model an industrial plant, with machinery and various “connections” between them: ducts, pipes, wiring, etc. For the sake of brevity, let’s look at pipes only. We will represent machinery as nodes and connections as edges. Connection attributes (e.g., pipe size and schedule) are represented as edge properties. A pipe connection between two machines would be a statement like this:

```
:M1 :pipe :M2 .
```

and the pipe attributes would be given like this;

```
:M1 :pipe :M2 .
<< :M1 :pipe :M2 >> :size "DN 100" ;
                    :schedule “30” .
```

(this would represent a pipe of 4.5“ outside diameter and 0.188” wall thickness [3]). If we now wanted to add another pipe between the same two machines, we would end up with the same problem as in our previous use case, since there is *only one* unique statement `<< :M1 :pipe :M2 >>`. A natural workaround would be to represent the pipe connection itself as a node (typical RDF higher arity relation trick), but it leads to a more verbose graph.

References

[1] https://aws.amazon.com/neptune/

[2] Ora Lassila, Michael Schmidt, Brad Bebee, Dave Bechberger, Willem Broekema, Ankesh Khandelwal, Kelvin Lawrence, Ronak Sharda, and Bryan Thompson: “Graph? Yes! Which one? Help!”. 1st Workshop on Squaring the Circle on Graphs (SCG2021), SEMANTiCS 2021. https://arxiv.org/abs/2110.13348

[3] https://en.wikipedia.org/wiki/Nominal_Pipe_Size