

# Next Version of RDF or: Reification Considered Annoying

Ora Lassila | Amazon Neptune

Adrian Gschwend | Zazuko

Pierre-Antoine Champin | INRIA & W3C

with contributions from the **W3C RDF-star WG**

---

# Speaker Introduction

## **Ora Lassila**

Member of the Amazon Neptune graph database team

Co-creator of RDF and the Semantic Web

Former W3C Fellow and an elected member of the W3C AB

Co-chair of the RDF-star WG

## **Adrian Gschwend**

Founder and CEO of Zazuko GmbH

Co-founder of Qleverize AG

Long-time practitioner of all things RDF

Co-chair of the RDF-star WG

## **Pierre-Antoine Champin**

Member of the research staff at INRIA

W3C staff liaison to the RDF-star WG

# Game Plan

1. Some background to RDF
2. Reification, statements about statements, "Reification Done Right"
3. RDF-star → RDF 1.2
4. RDF vs. LPG in the context of KGs

# Introduction to RDF

Simple KR language, cornerstone of the “Semantic Web stack”

Widely adopted and deployed

About **half of all Web pages** contain some embedded RDF

**All Adobe documents** contain some RDF

etc.

***“RDF has turned out to be the most **catastrophically successful failure** there is.”***

*- Charles Ivie (AWS)*

# RDF Timeline

May 1997	First version of the PICS-NG proposal
October 1997	First public “working draft” of something called RDF
February 1999	RDF “Model and Syntax”
February 2004	RDF “Model and Syntax” revised (“RDF 1.0”)
January 2008	SPARQL 1.0
March 2013	SPARQL 1.1
February 2014	RDF 1.1
December 2021	RDF-star CG final report
August 2022	RDF-star WG formed
Summer 2025	First RDF 1.2 Proposed Recommendations (?)

# Statements about statements?

Up until now done by “modeling” RDF statements in RDF

Reification: cumbersome, misunderstood, unpopular

One of the most persistent criticisms of RDF, particularly when comparing to Labeled Property Graphs (LPGs) - this may mostly be a problem for the “LPG people”

Reification could be seen as the RDF counterpart of Labeled Property Graph “edge properties” (RDF is more expressive, though)

Over the years, many ideas and proposals on how to fix this

Most notably, “Reification Done Right” by Hartig & Thompson in 2014

RDR eventually parlayed into “RDF-Star”, first a CG, now a WG

# Modeling practices and conventions in RDF

Common problem:

**:A :p :B**

(but what if you need to say things about the relationship?)

Standard reification:

**[ ] a rdf:Statement ; rdf:subject :A ; rdf:predicate :p ; rdf:object :B**

Typical solution - "lift" the relationship to be its own node:

**[ ] a :p\_as\_node ; :source :A ; :sink :B**

(for all intents and purposes equivalent with the standard reification)

# Some use cases

Provenance

Qualifying statements

“Marginalia”, annotations

Modalities (beliefs, etc.)

Also: bringing RDF closer to LPGs (aka “RDF/LPG alignment”)

(more on this later)

# Challenges

Triples in an RDF graph are unique (remember: set semantics)

**:Elizabeth :married :Richard**

We want to qualify this with a year

but... Elizabeth married Richard twice (1964 and 1975)

Breaking existing RDF (1.1) semantics is highly undesirable

Introducing new semantics could have consequences for RDFS & OWL

Relationship between statements and named graphs?

Is a reified statement a singleton named graph?

# Basic Idea of How RDF 1.2 Extends RDF 1.1

New type of term: **Triple Term**

(The other ones are IRI, blank node, and literal)

A triple term can be thought of as an un-asserted triple

Triple terms can be given an identifier (a “reifier”) via a new predicate **rdf:reifies**

```
:Marriage1 rdf:reifies <<( :ElizabethTaylor :married :RichardBurton )>>
```

**:Marriage1** is now an “occurrence” of the original triple

Triple terms cannot be used in the subject position of a triple

A triple term can have multiple reifiers → multiple “occurrences”

A reifier can identify multiple triple terms

(this was controversial in the WG)

# Semantics

RDF 1.1 model theory has been extended to cover the additions RDF 1.2 introduces

RDF adds the new property **rdf:reifies** and a new datatype **rdf:dirLangString**

RDF Schema adds a new class **rdfs:Proposition** such that

**rdf:reifies rdfs:range rdfs:Proposition**

Semantics has been defined with careful consideration to allow other things to be constructed on top of RDF 1.2 (e.g., OWL)

# Syntax

We have extended the RDF serialization formats to include triple terms  
(except RDF/XML)

In extended syntaxes, triple terms look like this: `<<( :A :p :B )>>`

In addition, some syntactic sugar is provided

Without the use of triple terms, the syntaxes remain the same as in RDF 1.1

(except that we now also allow bi-directional text in literals: datatype `rdf:dirLangString`)

JSON-LD is worked on by a separate WG

# Reification Syntax

```
<< :Elizabeth :married :Richard ~ :Marriage1 >> :date "1964-03-15"^^xsd:date .
```

Fully expanded, this is

```
:Marriage1 rdf:reifies <<(:Elizabeth :married :Richard )>> .
```

```
:Marriage1 :date "1964-03-15"^^xsd:date .
```

Note that the reified triple is not asserted

("not asserted" == "not part of the graph")

# Annotation Syntax

```
:Elizabeth :married :Richard ~ :Marriage1 { | :date "1964-03-15"^^xsd:date | }
```

This is syntactic sugar: the reified triple is also asserted, so this is the same as

```
:Elizabeth :married :Richard .
```

```
<< :Elizabeth :married :Richard ~ :Marriage1 >> :date "1964-03-15"^^xsd:date
```

which is the same as

```
:Elizabeth :married :Richard .
```

```
:Marriage1 rdf:reifies <<(:Elizabeth :married :Richard )>> .
```

```
:Marriage1 :date "1964-03-15"^^xsd:date .
```

# What About Multiple Triple Occurrences?

One triple, two marriages

```
:Elizabeth :married :Richard .
```

```
<< :Elizabeth :married :Richard ~ :Marriage1 >> :date "1964-03-15"^^xsd:date .
```

```
<< :Elizabeth :married :Richard ~ :Marriage2 >> :date "1975-10-10"^^xsd:date .
```

You could also use the annotation syntax to achieve this

# Where are we now? When will this end?

(Caveat: This work is **very hard**...)

All big problems have been solved and settled

Some details remaining to discuss

Some issues will be left to the next WG

Many older documents had to be brought up the current documentation standard

(lots of grunt work which nobody really sees)

Most documents (20+) approaching the "CR" stage

# RDF vs. LPG?

Question: Can LPGs be used for building KGs? (If so, do we really need RDF?)

Answer: They can, **but why would you?** RDF gives you a lot of stuff for free

Question: Does RDF 1.2 move RDF closer to LPGs? (If so, again, why RDF?)

Answer: Yes, **but RDF is more expressive** (and see previous answer also)

Comparison with modern art:

*"Hey, I **could** have done that!"*

*"Yeah, but you **didn't**."*

# Summary

Standard reification in RDF has been misunderstood and considered awkward

RDF is criticized for the lack of “edge properties”

RDF-star, now renamed RDF 1.2, simplifies the reification process

Syntax and semantics have been extended for a new term: triple term

Work is nearing completion

# Questions? Comments?

# #KGC2025

Join the Conversation



@TheKnowledgeGraphConference



oralassila  
adriangschwend  
pierreantoinechampin



@TheKnowledgeGraphConference



[www.knowledgegraph.tech](http://www.knowledgegraph.tech)