The different “Shapes” of RDF(S) and OWL: a fragmented history

Or: are Semantic Web standards (still) a good basis for Knowledge Graphs?
Great to be back 😊
How are Knowledge Graphs actually doing in 2024?

On the one hand...
What has changed?

- Adoption of the concept by major commercial players
- Fueled by “AI success stories”
- Standards (RDF, SPARQL) adopted by major vendors
- The focus has shifted
  - from (deductive) reasoning towards data quality (constraints)
  - towards “context”

- Are Semantic Web languages (in particular RDFS and OWL ...) still fit for this purpose?
Fueled by “AI success stories” 1/3

Google – User Experience:

- Rich Snippets
- Personalised recommendations across services:
Fueled by “AI success stories” 2/3

**IBM Watson:**
- Pre-LLM !!!
- Used DBpedia as one of its underlying knowledge bases! Essentially: formulating SPARQL queries underneath and using confidence scores.

https://youtu.be/P0Obm0DBvwI?t=951
Fueled by “AI success stories” 3/3
“The Future of Knowledge Graphs in a World of LLMs”

Denny Vrandečić, WikimediaFoundation, Keynote [KGC23]

Trend: RAG – Search+KG+LLMs!
ISWC2024 workshop: *Retrieval-Augmented Generation Enabled by Knowledge Graphs (RAGE-KG)*
Standards (RDF, SPARQL) adopted by major vendors

<table>
<thead>
<tr>
<th>Rank</th>
<th>DBMS</th>
<th>Database Model</th>
<th>Score Mar 2024</th>
<th>Feb 2024</th>
<th>Mar 2023</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Oracle</td>
<td>Relational, Multi-model</td>
<td>1221.06</td>
<td>-20.39</td>
<td>-40.23</td>
</tr>
<tr>
<td>2.</td>
<td>MySQL</td>
<td>Relational, Multi-model</td>
<td>1101.50</td>
<td>-5.17</td>
<td>-81.29</td>
</tr>
<tr>
<td>3.</td>
<td>Microsoft SQL Server</td>
<td>Relational, Multi-model</td>
<td>845.81</td>
<td>-7.76</td>
<td>-76.20</td>
</tr>
<tr>
<td>4.</td>
<td>PostgreSQL</td>
<td>Relational, Multi-model</td>
<td>634.91</td>
<td>+5.50</td>
<td>+21.08</td>
</tr>
<tr>
<td>5.</td>
<td>MongoDB</td>
<td>Document, Multi-model</td>
<td>424.53</td>
<td>+4.18</td>
<td>-34.25</td>
</tr>
<tr>
<td>6.</td>
<td>Redis</td>
<td>Key-value, Multi-model</td>
<td>157.00</td>
<td>-3.71</td>
<td>-15.45</td>
</tr>
<tr>
<td>7.</td>
<td>Elasticsearch</td>
<td>Search engine, Multi-model</td>
<td>134.79</td>
<td>-0.95</td>
<td>-4.28</td>
</tr>
<tr>
<td>8.</td>
<td>IBM Db2</td>
<td>Relational, Multi-model</td>
<td>127.75</td>
<td>-4.47</td>
<td>-15.17</td>
</tr>
<tr>
<td>9.</td>
<td>Snowflake</td>
<td>Relational</td>
<td>125.38</td>
<td>-2.07</td>
<td>+10.98</td>
</tr>
<tr>
<td>10.</td>
<td>SQLite</td>
<td>Relational</td>
<td>118.16</td>
<td>+0.88</td>
<td>-15.66</td>
</tr>
</tbody>
</table>

(Quoting Souripriya Das from Dagstuhl Seminar 24061 a month ago ;-) )
The focus has shifted

• towards “context”
• from (deductive) reasoning towards data quality (constraints)

Extremely rich, collaborative Knowledge Graph, directly integrated in Wikipedia

Available as RDF and can be queried in SPARQL

Rich contextual knowledge

Fine grained data available about context on a statement level:

• time
• provenance
• Information source
• edit information
• constraints

Revision history of "Lionel Messi" (Q615)

View log for this item (view above log)

Filter revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Action</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004-01-01</td>
<td>add</td>
<td>city of citizenship (p:P27)</td>
</tr>
<tr>
<td>2003-01-01</td>
<td>add</td>
<td>city of citizenship (p:P27)</td>
</tr>
<tr>
<td>2002-01-01</td>
<td>add</td>
<td>city of citizenship (p:P27)</td>
</tr>
<tr>
<td>2001-01-01</td>
<td>add</td>
<td>city of citizenship (p:P27)</td>
</tr>
</tbody>
</table>

https://www.wikidata.org/wiki/Q615
So, what happened to RDFS and OWL?

• Wikidata does not even use OWL and RDFS

• Are Semantic Web languages (in particular **RDFS and OWL** ...) still fit for this purpose?
Starting point/disclaimer:

• RDF (A-Box) Graph:
  \( :s :p :o . \)

• RDFS “T-Box Graph”:
  \( :p \text{rdfs:subClassOf} :q. \)

• OWL “T-Box Graph”:
  \( :p \text{rdfs:subClassOf} :q. \)
  \( :p \text{rdf:type} \text{owl:inverseFunctionalProperty}. \)

• RDFS “Vocabulary Graph”:
  \( \text{rdfs:Property} \text{rdf:type} \text{rdfs:Class} \)

In this talk, I mainly consider RDFS and OWL as RDF graphs.

When I talk about OWL fragments, I mean which of the OWL(+RDFS+RDFS) Vocabulary can be used how (syntactically) in an RDF graph.
Recovering history:
1999-2000 First versions of RDF + RDFS:

1999
• The first recommendation version of the RDF syntax and model (all XML):
  • Ora Lassila, Ralph Swick 22 February 1999
  • https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/
    • or: http://web.archive.org/web/20000815062516id_/http://www.w3.org/TR/REC-rdf-syntax/
• The actually first version of the RDF namespace document was published a bit before:
  • http://web.archive.org/web/19990508090931id_/http://www.w3.org/1999/02/22-rdf-syntax-ns

1999
• The first recommendation of RDF-Schema:
  • 1999 Proposed Recommendation version
    • never became a Standard, but advanced to Rec only with the RDF 2004 version!
• The first version of the RDF-Schema namespace document:
  • https://web.archive.org/web/20000816181854id_/www.w3.org/2000/01/rdf-schema

“RDF is a foundation for processing metadata”
Origins rather “Web metadata exchange format” than a“(Graph) data format”
First mention of RDF Schema in a W3C published document actually already 1998:

- W3C Note 1998 ([https://www.w3.org/TR/?filter-tr-name=RDF](https://www.w3.org/TR/?filter-tr-name=RDF))

---

**A Discussion of the Relationship Between RDF-Schema and UML**

**W3C Note 04-Aug-1998**

**This document:**  
http://www.w3.org/TR/1998/NOTE-rdf-uml-19980804

**Author:** Walter W. Chang, Advanced Technology Group, Adobe Systems

**Status of This Document**

This document is a **NOTE** made available by W3C for discussion only. This indicates no endorsement of its content, nor that W3C has had any editorial control in its preparation, nor that W3C has, is, or will be allocating any resources to the issues addressed by the NOTE.

Comments may be sent to [www-rdf-comments@w3.org](mailto:www-rdf-comments@w3.org). All mail is archived and available for review.

**Introduction**

This note summarizes the relationship between RDF-Schema and UML, the generic industry standard object-oriented modeling framework for information systems modeling. This note will briefly describe these systems then relate them to each other.

**RDF-Schema**
2002: first (draft) version of OWL

- Namespace document first version online:
  www.w3.org/2002/07/owl

2004: RDF and RDFS 1.0

- 10 February 2004: Rehaul of the RDF and RDFS vocabulary

2004: OWL1

• 10 February 2004: First official Recommendation of OWL

• http://web.archive.org/web/20040405111643id_/http://www.w3.org/2002/07/owl
2012: OWL2

- **11 December 2012**: Quite substantial extension of OWL1
  - various new language features
  - 3 sub”dialects”:
    - OWL RL
    - OWL EL
    - OWL QL

- What changed? Let’s check!
2014: RDF1.1 +RDF Schema 1.1

- 25 February 2014
- https://www.w3.org/TR/rdf11-concepts/
- https://www.w3.org/TR/rdf11-schema/

What’s new?
- https://www.w3.org/TR/rdf11-new/
  - IRIs instead of URIs and special characters allowed in IRIs.
  - New datatypes:
    - rdf:langString
    - rdf:HTML and rdf:XMLLiteral are non-normative in RDF 1.1
  - A table of RDF-compatible XSD datatypes has been added to RDF 1.1 Concepts and Abstract Syntax. Any XSD datatypes not represented in this table are incompatible with RDF
2024: RDF1.2!

- [https://www.w3.org/TR/rdf12-schema/](https://www.w3.org/TR/rdf12-schema/)

**What’s new?**
- Quoted triples
- rdf:dirLangString
- rdf:JSON
- rdf:HTML and rdf:XMLLiteral now normative
Let’s put these back on our timeline...

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First Recommendations</td>
<td>RDF 1.0</td>
<td>RDF 1.1</td>
<td>OWL2</td>
</tr>
<tr>
<td>RDF 1999</td>
<td>RDFS 2000</td>
<td>OWL 2002</td>
<td>RDF 2004</td>
</tr>
<tr>
<td>RDF 2004</td>
<td>RDFS 2004</td>
<td>OWL 2004</td>
<td>RDF 2014</td>
</tr>
<tr>
<td>RDF 2014</td>
<td>RDFS 2014</td>
<td>OWL 2012</td>
<td>RDF 2024</td>
</tr>
<tr>
<td>RDF 2024</td>
<td>RDFS 2024</td>
<td>OWL 2024</td>
<td></td>
</tr>
</tbody>
</table>

... and have a closer look:
- Interesting asynchronicity of the standard’s evolution...
- What was there from the beginning?
- Some things came and went...
- Are all of these constructs needed/used? in practice?

~2000
Let’s put these on a Timeline…

1999-2002

First Recommendations RDF 1.0


2001-2004

OWL 2002 RDF 2004 RDFS 2004

2012-

RDF 1.1

OWL 2012 RDF 2024 RDFS 2024

2022-

RDFS 2024 OWL 2024

2024-

... and have a closer look:

• Interesting asynchronicity of the standard’s evolution…
• What was there from the beginning?
• Some things came and went…
• Are all of these constructs needed/used? in practice?

~2004
Let’s put these on a Timeline…

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First Recommendations</td>
<td>RDF 1.0</td>
<td>RDF 1.1 OWL2</td>
</tr>
</tbody>
</table>

… and have a closer look:

- Interesting asynchronicity of the standard’s evolution...
- What was there from the beginning?
- Some things came and went...
- Are all of these constructs needed/used? in practice?

~2014

... and RDF1.2 adding a couple more as we speak ;-)
Even the smallest fragment...

- ... allows things (syntactically) that don’t make intuitive sense, or at least seem to be “distracting”... to most people who do NOT come from an RDF world.

Note: I’d argue that this is possibly one of the reasons for “slow” adoption.
“axiomatic” triples

• The vast majority of axiomatic triples seem to be an unnecessary burden, e.g.:

Only there to make the integration of axioms into the graph work, in a way to “justify” the “mix” of syntax and semantics.
This IS possible...

```
rdf:subClassOf
    a owl:SymmetricProperty .
```

```
rdf:type
    rdf:subPropertyOf rdfs:subClassOf, owl:imports.
```

```
rdfs:subClassOf
```

BTW, you here can sure think of similar issues in “SHACL graphs”...

Bottomline requirement:

You want to have the axioms and constraints represented in/with the graph, but you want to *syntactically* ensure, it keeps *separable*
Apart from the official W3C standards

• There’s a long list of OWL “fragments”
  • partially syntactically
  • partially semantically motivated:
    • **OWL “ter Horst”** (2005) syntactic/semantic (Horn Logic)
    • **OWL Flight** (2005) semantic (CWA/Constraint reading)
    • **RDFS-** ... Minimal RDFS (2007) syntactic/semantic
    • **OWL LD** (2012) syntactic/usage-motivated
    • Other fragments under discussion in the course of **OWL2**, such as **“RDFS3.0”**:
      • [https://www.w3.org/2007/OWL/wiki/Fragments](https://www.w3.org/2007/OWL/wiki/Fragments)
Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary

Herman J. ter Horst
“Semantic” fragment:

• Datalog-Based semantics

• Unique Names Assumption

• Important thing: proposing alternative constraint reading of property restrictions!
OWL IC (2010)

• Similar idea!
• Read (some) OWL axioms as constraints
• E.g.:

```xml
CatOwner rdfs:subClassOf [ a owl:Restriction;
    owl:onProperty owns;
    owl:somevaluesFrom Cat ]
```

• “deductive” reading: there is a (possibly unknown) cat
• vs.
• “constraining” reading: there has to be a (known) owned cat

• Problem: what about UNA? What about CWA?

... and it's pretty ugly to write this as RDF triples
• Goal: Define a fragment of “really used” OWL based on vocabulary usage
  • RDF Schema features amongst the most prominently used
  • OWL 2 features not used a lot prominently
• Goal: Define a fragment of “really used” OWL based on vocabulary usage
  • RDF Schema features amongst the most prominently used
  • OWL 2 features not used a lot prominently
  • Mostly single-triple expressible axioms

→ Essential idea: “Single-triple axiom OWL RL”
RDFS- (2007)

Minimal Deductive Systems for RDF

Sergio Muñoz¹, Jorge Pérez²,³, and Claudio Gutierrez⁴

¹ Universidad Católica de la Santísima Concepción, Chile
² Pontificia Universidad Católica de Chile
³ Universidad de Talca, Chile
⁴ Universidad de Chile

• Arguing (well!) that only a minimal subset of the RDFS vocabulary is semantically relevant, obviously, this is a subset of OWL LD
So, let’s maybe dare a “fresh start” on Ontologies & Shapes?

Idea:

• Let’s dare to keep it simple and constrain ourselves! 😊

• Start minimal.
Incremental Proposal, how could it look?

- Start from
  - *standard-use* of the
  - *minimal RDFS* vocabulary
- And extend this fragments by *features* (from **OWL LD**):
  - both syntactically and semantically
  - start with UNA, CWA, add (limited) equality reasoning later
- Goal: build up – gradually –
  - **Useful** and “**Safe**” OWL fragment(s)
  - Canonical means to fall back/repair non-compliant OWL ontologies to meet the required restrictions.
- Hope (hidden goal): these safe fragments are also “compatible” with
  - New standards for constraints and SHAPES (SHACL, ShEx, etc.)
  - Modeling **context**!
Reasonable starting points 1/3:
Standard use of the RDF, RDFS, and OWL vocabulary

Further restrictions well conceivable, and expressible in SHAPEs:

- Use annotation properties only on URIs that denote an ontology.
- Don’t explicitly use classes in G_res

What Are Links in Linked Open Data? A Characterization and Evaluation of Links between Knowledge Graphs on the Web

ARMIN HALLER, Australian National University, Australia
JAVIER D. FERNÁNDEZ, Complexity Science Hub Vienna, Austria
MAULIK R. KAMDAR, Stanford University, USA
AXEL POLLERES, Vienna University of Economics and Economics and Business, Austria
Reasonable starting points 1/3: Going beyond “Standard use”

• More “tool-supportable” OWL fragments, e.g. enforce (or repair) what makes sense for ontology editors:
  
  • Keep DatatypeProperties and ObjectProperties separate, i.e. ensure all properties are either DatatypeProperties xor ObjectProperties
  
  • Disallow meta-modelling (or enable canonical ways to disambiguate user-defined URIs used as classes and instances).
  
  • Disallow “cycles” in taxonomies
  
  • Disable “URI hijacking”
  ... 

  **Open question:**
  *(How) can we also enforce this by syntactic restrictions on vocabulary usage?*
Reasonable starting points 2/3:

- **Minimal RDFS:**
  - Argue – essentially that for RDFS, only the properties
    - `rdfs:subPropertyOf [sp]`,
    - `rdfs:subClassOf [sc]`,
    - `rdfs:domain [dom]`,
    - `rdfs:range [range]`
    - `rdf:type [type]`
  are relevant.

---

**Minimal Deductive Systems for RDF**

Sergio Muñoz¹, Jorge Pérez²,³, and Claudio Gutierrez⁴

¹ Universidad Católica de la Santísima Concepción, Chile
² Pontificia Universidad Católica de Chile
³ Universidad de Talca, Chile
⁴ Universidad de Chile

---

1. **Simple:**
   - (a) $\frac{s}{s'}$ for a map $\mu : G' \rightarrow G$
   - (b) $\frac{s}{s'}$ for $G' \subseteq G$

2. **Subproperty:**
   - (a) $\frac{(A,sp,B) (B,sp,C)}{(A,sp,C)}$
   - (b) $\frac{(A,sp,B) (A',sp,A)}{(A,B',Y)}$

3. **Subclass:**
   - (a) $\frac{(A,sc,B) (B,sc,C)}{(A,sc,C)}$
   - (b) $\frac{(A,sc,B) (A,type,A)}{(A,type,B)}$

4. **Typing:**
   - (a) $\frac{(A,range,B) (X,A,Y)}{(X,type,X)}$
   - (b) $\frac{(A,range,B) (X,A,Y)}{(Y,type,Y)}$

5. **Implicit Typing:**
   - (a) $\frac{(A,dom,B) (C,sp,A) (X,C,Y)}{(X,type,B)}$
   - (b) $\frac{(A,range,B) (C,sp,A) (X,C,Y)}{(Y,type,Y)}$

6. **Subproperty Reflexivity:**
   - (a) $\frac{(X,A,Y)}{(A,sp,A)}$
   - (b) $\frac{(A,sp,B)}{(A,sp,B)}$
   - (c) $\frac{(p,p,p)}{(p,p,p)}$ for $p \in \text{rdf}$
   - (d) $\frac{(A,p,X)}{(A,sp,A)}$ for $p \in \{\text{dom, range}\}$

7. **Subclass Reflexivity:**
   - (a) $\frac{(A,sc,B)}{(A,sc,A)}$
   - (b) $\frac{(A,p,X)}{(A,sc,A)}$ for $p \in \{\text{dom, range, type}\}$
Reasonable Starting points 3/3: Connecting RDF to Property Graphs (PGs)...

... needs **Reification**, but reification does not necessarily complicate things!
Reasonable Starting points 3/3: Connecting RDF to Property Graphs (PGs)...

• ... needs **Reification**, but reification does not necessarily complicate things!

e.g. Singleton reification (with reasonable syntactic constraints) can cover PGs...
  • i.e., something like:

  1. Drop namespaces (or restrict to 1 namespace)
  2. `edgeType` a owl:inverseFunctionalProperty.
  3. `edgeType` rdfs:subPropertyOf rdfs:subPropertyOf.
  4. Shape constraint:
     each other property used on a property in the domain of `edgeType`
     is constrained to be a owl:DatatypeProperty

---

**Don’t Like RDF Reification?**

**Making Statements about Statements Using Singleton Property**

<table>
<thead>
<tr>
<th>Name</th>
<th>Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vinh Nguyen</td>
<td>Kno.e.sis Center</td>
</tr>
<tr>
<td></td>
<td>Wright State University</td>
</tr>
<tr>
<td></td>
<td>Ohio, USA</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:vinh@knoesis.org">vinh@knoesis.org</a></td>
</tr>
<tr>
<td>Olivier Bodenreider</td>
<td>National Library of Medicine, National Institute of Health</td>
</tr>
<tr>
<td></td>
<td>Maryland, USA</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:olivier@nlm.nih.gov">olivier@nlm.nih.gov</a></td>
</tr>
<tr>
<td>Amit Sheth</td>
<td>Kno.e.sis Center</td>
</tr>
<tr>
<td></td>
<td>Wright State University</td>
</tr>
<tr>
<td></td>
<td>Ohio, USA</td>
</tr>
<tr>
<td></td>
<td><a href="mailto:amit@knoesis.org">amit@knoesis.org</a></td>
</tr>
</tbody>
</table>
Take-home messages:

- RDF remains a great “graph exchange” format...
  - ... although it was not created for that!
  - A lot of work and thought have been put into it, over 20+ years, which we should probably not re-invent.
  - Some things are more complicated than needed/actually useful.

- The RDF, RDFS and OWL vocabulary allow us to store axioms within the data
  - Feature or bug? ... probably it’s a feature
  - BUT: the reserved vocabulary should (IMHO) not be tempered with \( \rightarrow \) needs syntactic constraints (“shapes”)

- Unifying RDF, PGs (and even RDB) under one roof should be nicely possible under an RDF “roof”
  - If we enforce syntactic restrictions to constrain (reserved) vocabulary usage by shape constraints.

- Which language for “shape constraints”?
  - Partially, the OWL and RDFS vocabulary can be “read as constraints” itself (OWL IC, OWL Flight approaches)
  - SHACL? SheX?
  - Probably more features needed for things (e.g. acyclicity checks) covered by neither

Let’s dare to take step(s) back and (re-)start simple(r)!