

RDF Serialization

Kseniia Blokhina

29/10/2024

Overview

- RDF (Resource Description Framework) allows us to specify graphs that are:
 - Directed (edges have a source and target)
 - Edge-labelled (each edge has one label)
 - A restricted form of multi-graphs (multiple edges can exist between the same vertices with different labels)

Graphs

- Graphs can be represented in various concrete ways, including graphical diagrams.
- Machines require an RDF serialization to interpret graphs, which allows storing and transmitting triples in a file.

Numerous syntactic formats are available for RDF serialization:

- **N-Triples** - simple, line-based format
- **Turtle** - adds convenient abbreviations to N-Triples
- **JSON-LD** - encodes RDF graphs in JSON
- **XML** - encodes RDF graphs in XML
- **RDFa** - embeds RDF graphs into HTML

N-Triples Basics

- **N-Triples** represent each RDF triple with:
 - Subject
 - Predicate
 - Object
- **Syntax:** Triple statements are separated by whitespace and terminated by a dot (‘.’) after each triple.

Example:

```
<http://orcid.org/0000-0002-3178-0201> <http://purl.org/dc/terms/created> "2014-12-22T22:25:56.900Z"^^<http://www.w3.org/2001/XMLSchema#dateTime>.  
<http://orcid.org/0000-0002-3178-0201> <http://www.loc.gov/mads/rdf/v1#hasAffiliation>  
<http://www.grid.ac/institutes/grid.152326.1>.  
<http://orcid.org/0000-0002-3178-0201> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>  
<http://xmlns.com/foaf/0.1/Person>.  
<http://orcid.org/0000-0002-3178-0201> <http://www.w3.org/2000/01/rdf-schema#label> "Julian Hillyer".
```

N-Triples: IRIs and Literals

IRIs

- In N-Triples, IRIs (Internationalized Resource Identifiers) must be written as absolute IRIs.
- They are enclosed in < > and may contain numeric escape sequences.

RDF Literals

- Literals are used to identify values, such as strings, numbers, and dates.
- Syntax includes:
 - A lexical form (the actual string value).
 - An optional language tag (preceded by @).
 - An optional datatype IRI (preceded by ^^).
 - Blank nodes are expressed with : *followed by a label* (e.g., :alice).

N-Triples: Example

```
<http://example.org/show/218> <http://www.w3.org/2000/01/rdf-schema#label> "That Seventies Show"^^<http://www.w3.org/2001/XMLSchema#string> . # literal with XML Schema string datatype
<http://example.org/show/218> <http://www.w3.org/2000/01/rdf-schema#label> "That Seventies Show" . # same as above
<http://example.org/show/218> <http://example.org/show/localName> "That Seventies Show"@en . # literal with a language tag
<http://example.org/show/218> <http://example.org/show/localName> "Cette Série des Années Septante"@fr-be . # literal outside of ASCII range with a region subtag
<http://example.org/#spiderman> <http://example.org/text> "This is a multi-line\nliteral with many quotes (\\"\\\\"\\\\"\\")\nand two apostrophes ('')." .
<http://en.wikipedia.org/wiki/Helium> <http://example.org/elements/atomicNumber>
"2"^^<http://www.w3.org/2001/XMLSchema#integer> . # xsd:integer
<http://en.wikipedia.org/wiki/Helium> <http://example.org/elements/specificGravity> "1.663E-4"^^<http://www.w3.org/2001/XMLSchema#double> . # xsd:double
```

Figure 1: Code Snippet 2

N-Triples: Example

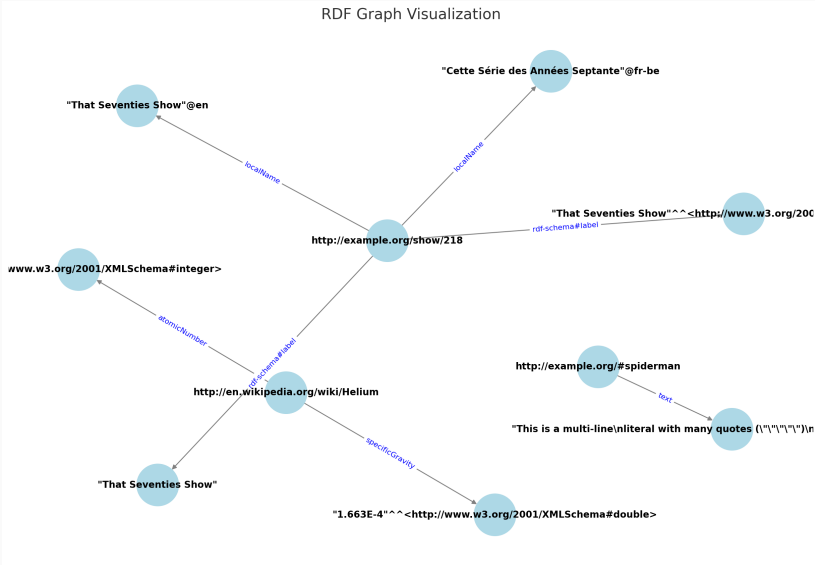


Figure 2: RDF Graph 1

Turtle Serialization

Features

- Prefix declarations and base namespaces allow shortening IRIs.
- Blank nodes can be encoded with square brackets, allowing in-line predicate-object pairs.
- Easy to parse and straightforward for humans to read.

Unfortunately, it's quite costly to parse compared to N-Triples.

Syntax

- Numbers can be written directly without using quotes or specifying types, as they are automatically treated as default types like integer, decimal, or double.
- Booleans can also be written as true or false directly

Turtle: Example

- If two triples share both the same subject and predicate, the two objects can be separated by commas

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix orcid: <http://orcid.org/> .  
  
orcid:0000-0002-3178-0201 rdf:type foaf:Person;  
                           rdfs:label "Julian Hillyer",  
                                   "J. Hillyer".
```

Figure 3: Code Snippet 3

Turtle: Example

- If several triples share the same subject, the predicates and objects can be listed, separated by semicolons

```
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<#green-goblin>
  rel:enemyOf <#spiderman> ;
  a foaf:Person ;
  foaf:name "Green Goblin" .

<#spiderman>
  rel:enemyOf <#green-goblin> ;
  a foaf:Person ;
  foaf:name "Spiderman", "Spiderman"@en .|
```

Figure 4: Code Snippet 4

Turtle: Example

RDF Graph: Spiderman and Green Goblin Relationship

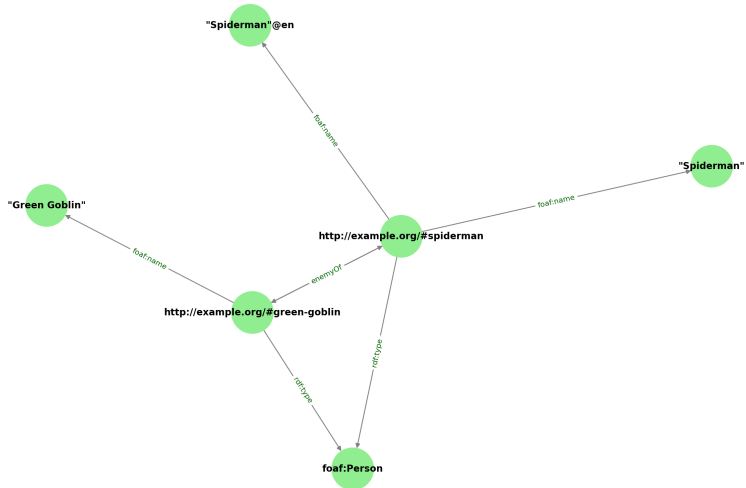


Figure 5: RDF Graph 2

RDF/XML is the oldest RDF serialization format, initially chosen due to widespread XML support.

Challenges: Blends XML's tree structure with RDF's graph model, making it verbose and hard to understand.

Primary Components:

- **Graph nodes:** `rdf:Description`
 - `rdf:about` attribute can be added if the node is an IRI
- **Predicate arcs** are nested within nodes: `ex:editor`

Multiple Property

Example

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
          </rdf:Description>
        </ex:homePage>
      </rdf:Description>
    </ex:editor>
  </rdf:Description>

<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:fullName>Dave Beckett</ex:fullName>
    </rdf:Description>
  </ex:editor>
</rdf:Description>

<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <dc:title>RDF 1.1 XML Syntax</dc:title>
</rdf:Description>
```

... to abbreviation

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
          </rdf:Description>
        </ex:homePage>
        <ex:fullName>Dave Beckett</ex:fullName>
      </rdf:Description>
    </ex:editor>
    <dc:title>RDF 1.1 XML Syntax</dc:title>
  </rdf:Description>
```

Figure 7: Code Snippet 5

... to abbreviation

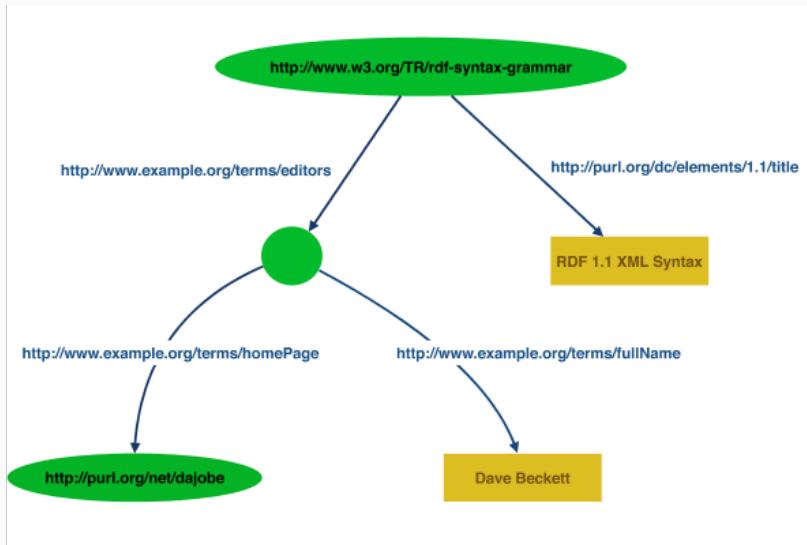


Figure 8: RDF Graph 3

More abbreviations

- When a predicate arc in an RDF graph points to an object node which has no further predicate arcs this form can be shortened
- When a property element's content is string literal, it may be possible to use it as an XML attribute on the containing node element.

1 Using multiple property elements on a node element

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage>
        <rdf:Description rdf:about="http://purl.org/net/dajobe/">
          <rdf:Description>
            <ex:fullName>Dave Beckett</ex:fullName>
          </rdf:Description>
        </ex:homePage>
      <ex:fullName>Dave Beckett</ex:fullName>
    </rdf:Description>
  </ex:editor>
</dc:title>RDF 1.1 XML Syntax</dc:title>
</rdf:Description>
```

2 Empty property elements

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar">
  <ex:editor>
    <rdf:Description>
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
        <ex:fullName>Dave Beckett</ex:fullName>
      </rdf:Description>
    </ex:editor>
  <dc:title>RDF 1.1 XML Syntax</dc:title>
</rdf:Description>
```

3 Replacing property elements with string literal content into property attributes

```
<rdf:Description rdf:about="http://www.w3.org/TR/rdf-syntax-grammar"
  dc:title="RDF 1.1 XML Syntax">
  <ex:editor>
    <rdf:Description ex:fullName="Dave Beckett">
      <ex:homePage rdf:resource="http://purl.org/net/dajobe/">
    </rdf:Description>
  </ex:editor>
</rdf:Description>
```

Figure 9: Abbreviations

- RDFa (RDF in Attributes) - an HTML embedding of RDF triples; used for HTML document annotations (e.g., with schema.org)
 - By adding attributes to HTML elements, you can give semantic context to the content inside your webpages
- Mostly used by web crawlers (e.g., Google) to enhance search previews.

How RDFa Works

It combines RDF with view data (HTML)

Example

```
<div about="https://www.w3.org/People/Berners-Lee/">
  <p>
    Tim is born on
    <span property="http://schema.org/birthDate">1955-06-07</span>
    in
    <a
      property="http://schema.org/birthPlace"
      href="http://dbpedia.org/resource/London"
    >
      London
    </a>
  </p>
</div>
```

Figure 10: Code Snippet 6

Use RDFa if you want to add lightweight RDF support to existing HTML rather than full RDF compatibility

- **But:** It makes HTML documents larger and harder to manage.

For higher data volumes or ease of use, consider JSON-LD or N-Triples

What is JSON-LD?

- JSON-LD is an extension of JSON, widely used in web applications.
- It's fully compatible with JSON, allowing easy integration with existing JSON APIs. Ideal for RESTful JSON APIs where RDF parsing performance is not critical.

How it Works:

- Converts JSON data into RDF by adding an @context object to map keys to RDF Classes and Properties.

Example

```
{
  "@context": {
    "dbpedia": "http://dbpedia.org/resource/",
    "schema": "http://schema.org/"
  },
  "@id": "https://www.w3.org/People/Berners-Lee/",
  "schema:birthDate": "1955-06-08",
  "schema:birthPlace": {
    "@id": "dbpedia:London"
  }
}
```

Figure 11: Code Snippet 7

HDT

- HDT is both a compact data structure and a binary serialization format for RDF.
- it's designed to save space and bandwidth.

Key Features

- Highly Efficient
- Built-in Indexing
- HDT compression is resource-intensive

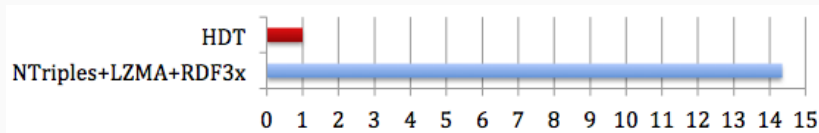


Figure 12: Total time

Thank you for your attention!

Sources

1. <https://iccl.inf.tu-dresden.de/w/images/d/d2/KG2020-Lecture-02-overlay.pdf>
2. <https://heardlibrary.github.io/digital-scholarship/lod/serialization/>
3. <https://www.w3.org/TR/turtle/>
4. <https://ontola.io/blog/rdf-serialization-formats>
5. <https://www.w3.org/TR/rdf-syntax-grammar/>
6. Martínez-Prieto, Miguel A., Mario Arias Gallego, and Javier D. Fernández. “Exchange and consumption of huge RDF data.” The Semantic Web: Research and Applications: 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings 9. Springer Berlin Heidelberg, 2012.