JSON-LD for Linked Open Data and Knowledge Graphs

Dr. Jans Aasman (allegrograph.com)
Contents

- The world of Semantics and RDF and Graphs
- The world of JSON and Documents
- A powerful intersection: JSON-LD: JSON + RDF
- Store it in a document store or in a semantic graph database?
The world of Semantics and RDF

• Tim Berners-Lee in 1994: only humans can read the context on the web, computers can’t. We need a meta data language that gives meaning to objects on the web.

• Invented RDF to add Semantics to objects.
RDF at the core is simple

[1] Use IRIs as a universal identifier mechanism
[2] Describe any type of (meta) data as triples
   <subject> <predicate> <object>
   subjects* and predicates are always IRIs
   objects can be IRIs or any XSD datatype
[3] Add meaning to data by a self-describing logic called OWL

* ok, ok, subjects can also be blank nodes
There is a person born in 1958 with first name “Jans” and last name “Aasman” who lives in a place with the name “Moraga” (pop 16000) that is part of state “California (pop 34 M)
# Turtle, same information...

@prefix : <http://abc.com/> .
:person1 a :Person ;
   :firstName 'Jans' ;
   :lastName 'Aasman' ;
   :birthYear 1958 ;
   :livesIn :place1 .
:place1 a :Place
   :name 'Moraga' ;
   :population 16000 ;
   :partOf :state1 .
:state1 a :State;
   :name 'California' ;
   :population 340000000 .
Jans has two places, is married to Sophia. Jans pays taxes to California and USA and they have a son in Stockton, New Jersey.
Visual SPARQL: find two people that lives in two places in California where one person also pays taxes in California
```sparql
prefix : <http://abc.com/>
select ?fn1 ?ln1 ?fn2 ?ln2 where {
  ?place1 :partOf :state1 .
  ?person1 :paysTaxesTo :state1 ;
    :livesIn ?place1 , ?place2 ;
    :lastName ?ln1 ;
    :firstName ?fn1 .
  ?person2 :livesIn ?place1 , ?place2 ;
    :marriedTo ?person1 ;
    :firstName ?fn2 ;
    :lastName ?ln2 .
  filter ( ?place2 != ?place1 )
}
limit 1
```

<table>
<thead>
<tr>
<th>?fn1</th>
<th>?ln1</th>
<th>?fn2</th>
<th>?ln2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jans</td>
<td>Aasman</td>
<td>Sophia</td>
<td>Nysingh</td>
</tr>
</tbody>
</table>
And a tiny ontology in OWL

```
:Person a owl:Class ; rdfs:subClassOf :Mammal.
:Mammal a owl:Class .
:hasParents a owl:ObjectProperty .
:hasChild owl:inverseOf :hasParents .
:marriedTo rdfs:domain :Person ;
    rdfs:range   :Person .
```
Is Jans a Mammal

- And now we can reason too

```sparql
prefix :<http://abc.com/>
select * { ?p :firstName 'Jans' ; a :Mammal }
```
Does Jans have Children?

prefix : <http://abc.com/>
select * { ?person1 :hasChild ?person2 }

2 Results

<table>
<thead>
<tr>
<th>?person1</th>
<th>?person2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sophia Nysingh</td>
<td>Hans Aasman</td>
</tr>
<tr>
<td>Jans Aasman</td>
<td>Hans Aasman</td>
</tr>
</tbody>
</table>
Semantic Technology Standards

Main Page

In addition to the classic “Web of documents” W3C is helping to build a technology stack to support a “Web of data,” the sort of data you find in databases. The ultimate goal of the Web of data is to enable computers to do more useful work and to develop systems that can support trusted interactions over the network. The term “Semantic Web” refers to W3C’s vision of the Web of linked data. Semantic Web technologies enable people to create data stores on the Web, build vocabularies, and write rules for handling data. Linked data are empowered by technologies such as RDF, SPARQL, JSON-LD, OWL, SHACL and SKOS.

The goal of this wiki is to provide a “first stop” for more information on Semantic Web technologies, in particular on Semantic Web Standards published by the W3C. It does not aim to give a complete set on information on Semantic Web related events, conferences, ontologies or community efforts. There are already a number of sites maintained by the community that users can refer to (see some below).

Learn

Books, Presentations, FAQ

Main areas of SW

Linked Data, Vocabularies, Queries, Inference, Vertical Applications

The Standards

RDF, OWL, SPARQL, RDFa, JSON-LD, SKOS, RDFS, GRDDL, POWDER, PROV, RIF, SAWSDL, RDB2RDF, SHACL

Usage Examples

Look at what others have done

Develop

Consult the list of available development tools

Official publications

W3C SW Publications
Where is RDF Used

- Nearly every Fortune 500 company is building SKOS taxonomies
- In Linked Open Data for public data and enterprise data
- Nearly every Fortune 500 company is building Knowledge Graphs
RDF for taxonomies and ontologies

• All Fortune 500 companies are now building taxonomies, almost all of them RDF based
  • To improve search over documents
  • To improve NLP processes
  • To harmonize documents, databases, and processes
  • To build knowledge graphs
• Most taxonomies and ontologies are meant to be cross-enterprise and therefore mostly open source
• Two examples:
RDF based open source taxonomies and ontologies: Healthcare and Life Sciences are furthest ahead
About FIBO

The Financial Industry Business Ontology

The Financial Industry Business Ontology (FIBO) defines the sets of things that are of interest in financial business applications and the ways that those things can relate to one another. In this way, FIBO can give meaning to any data (e.g., spreadsheets, relational databases, XML documents) that describe the business of finance.

FIBO is hosted and sponsored by the Enterprise Data Management Council (EDMC) and is published in a number of formats for operating use and business definitions. FIBO is a trademark of EDM Council, Inc. It is also standardized by the Object Management Group (OMG).

FIBO is developed as an ontology in the Web Ontology Language (OWL). The language is codified by the World Wide Web Consortium (W3C), and it is based on Description Logic. The use of logic ensures that each FIBO concept is framed in a way that is unambiguous and that is readable both by humans and machines.
Linked Open Data (public)

- (semi) public (meta) data available in various RDF formats or even fully formed Knowledge Graphs
Linked Open Data (enterprise)

- Create enterprise digital asset catalogs to get a handle on your data lakes
It started in the Semantic Web Community
Federated queries over Linked Data

- What is the median income of the area where Barack Obama was born?
  - DPBedia to find Birthplace -> GeonamesID
  - GeoNames database to find other GeonamesIDs within 10 miles
  - Census to find median income for GeonamesIDs
Using service calls in SPARQL

```sparql
PREFIX geo: <http://franz.com/ns/allegrograph/3.0/geospatial/>
PREFIX geonames: <http://sws.geonames.org/>
PREFIX dbpedia_rsrc: <http://dbpedia.org/resource/>
PREFIX dbpedia_onto: <http://dbpedia.org/ontology/>
PREFIX dbpedia_prop: <http://dbpedia.org/property/>
PREFIX census: <tag:govshare.info,2005:rdf/census/>
PREFIX census_samp: <tag:govshare.info,2005:rdf/census/details/samp/>

SELECT distinct ?censusplace ?income {
  ?birthplace dbpedia_prop:hasGeonamesID ?geonamesresource .

  SERVICE <https://localhost:10000/catalogs/demos/repositories/geonames> {
    ?geonamesresource geonames:isAt5 ?location .
    ?otherplace geo:inCircleMiles (geonames:isAt5 ?location 10) .
    ?otherplace geonames:feature_code "PPL" .
    ?geonamesresource geonames:feature_code "PPL" .

    SERVICE <https://localhost:10000/catalogs/demos/repositories/census> {
      ?censusplace dbpedia_prop:hasGeonamesID ?otherplace .
      ?detail census_samp:population15YearsAndOverWithIncomeIn1999 ?d .
    }
  }
}
```

<table>
<thead>
<tr>
<th>?censusplace</th>
<th>?income</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iroquis point</td>
<td>33005</td>
</tr>
<tr>
<td>Pearl city</td>
<td>25776</td>
</tr>
<tr>
<td>Waimalu</td>
<td>34777</td>
</tr>
<tr>
<td>Maunawili</td>
<td>41886</td>
</tr>
</tbody>
</table>
Knowledge Graphs on the rise
All the big ones in the US heavily investing in it

<table>
<thead>
<tr>
<th>Day 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00 Christos Boutsidis  Goldman Sachs Pythia: the Goldman Sachs Social Graph</td>
</tr>
<tr>
<td>9:20 Patricia Branum  Capital One Knowledge Graph Pilot Improves Data Quality While Providing a Customer 360 View</td>
</tr>
<tr>
<td>9:40 David Newman  Wells Fargo Knowledge Graphs and AI: The Future of Financial Data</td>
</tr>
<tr>
<td>10:00 Tim Baker  Refinitiv Financial Crime</td>
</tr>
<tr>
<td>break 10:20 - 11:00 Coffee and snacks will be provided</td>
</tr>
<tr>
<td>11:00 Denny Vrandecic  Google AI Wikidata, Knowledge Graphs, and Beyond</td>
</tr>
<tr>
<td>11:20 Pierre Haren  Causality Link Graphs</td>
</tr>
<tr>
<td>11:40 Dieter Fensel  OnLinx Talking Knowledge Graphs</td>
</tr>
<tr>
<td>12:00 Chris Brockmann  Eccenca Knowledge Graph for Digital Transformation in the Supply-Chain</td>
</tr>
<tr>
<td>12:20 Tom Plasterer  Astrazeneca FAIR Data Knowledge Graphs – From Theory to Practice</td>
</tr>
<tr>
<td>lunch 12:40 - 1:40 A light lunch will be provided</td>
</tr>
<tr>
<td>1:40 Subhabrata Mukherjee  Amazon Deep Learning for Knowledge Extraction and Integration to build the Amazon Product Graph</td>
</tr>
<tr>
<td>2:00 Teresa Tung  Accenture Using a Domain Knowledge Graph to Manage AI at Scale</td>
</tr>
<tr>
<td>2:20 Alfio Gliozzo  IBM Research Extending Knowledge Graphs using Distantly Supervised Deep Nets</td>
</tr>
<tr>
<td>2:40 Michael Tung  Diffbot Knowledge Graphs for AI</td>
</tr>
<tr>
<td>break 3:00 - 3:30 Coffee and snacks will be provided</td>
</tr>
<tr>
<td>3:30 Xiaoya Wei  Airbnb Knowledge Graph at Airbnb</td>
</tr>
<tr>
<td>3:50 Amy Hodler  Neo4j A Real-World Guide to Building Your Knowledge Graphs</td>
</tr>
<tr>
<td>4:10 Juan F. Sequeda  Capsenta Designing and Building Enterprise Knowledge Graphs from Relational Databases in the Real V</td>
</tr>
<tr>
<td>4:30 Ron Snyder  ITHAKA / JSTOR Why Wikibase? Why not?</td>
</tr>
<tr>
<td>4:50 Sören Auer  TIB Creating a knowledge graph based Enterprise Data Innovation Architecture</td>
</tr>
</tbody>
</table>
• Enterprises lack a **single source of truth** on their customers and their data
• Data scientists are expensive and waste 90% of their time on prep
• Current solutions (ETL, Data Lakes, MDM, Neo4J) **exacerbate the Silo problem**
• Knowledge Graphs:
  • Silo removal
  • KGs make Artificial Intelligence truly Intelligent
  • Basis for Enterprise Data Fabrics

• LinkedIn, ebay, Uber, Airbnb, Apple, Google, etc. are building Knowledge Graphs

CONFIDENTIAL – Do not copy or redistribute
Graph Market will grow from $1.9 billion in 2021 to $5.1 billion by 2026, CAGR of 22.5%

Gartner Client inquiry interest in data fabric in 2021 is approx. 900% of that in 2019
Graph > Knowledge Graph > Data Fabric

#3 Watch out for graph databases

Carl Olofson, Research Vice President, IDC

- Next revolution in database technology
- Potential for 600% growth over the decade
- Will slowly gain traction over the next five years
- Lots of use cases but not well understood
  - Semantic use cases (lots of AI examples)
  - Property graphs with dozens of use cases
- Graph databases dramatically different from traditional RDBMS

Gartner.

Top Strategic Technology Trends for 2022: Data Fabric

Published 18 October 2021 - ID G00749680 - 15 min read

By Mark Beyer, Ehtisham Zaidi, and 2 more

Knowledge Graphs are the Key to Data Fabrics

<table>
<thead>
<tr>
<th>Connectible:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• KGs can bridge all data and metadata &quot;Silos&quot; for seamless data integration and management</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flexible:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Express any data and metadata: this is important as new data sources get added to the fabric and new metadata is collected</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integratable:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Offer the most complete, open and flexible APIs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expressive:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Capture semantically rich models to assist with the understanding of data and metadata</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Composable:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Make it easy to incrementally evolve a data fabric</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Future Proof:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Based on Standards</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Intelligent:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Integrate rules-based reasoning and machine learning: KGs provide the semantics layer that adds additional context and meaning enabling better and more informed analytics and AI/ML</td>
</tr>
</tbody>
</table>

CONFIDENTIAL – Do not copy or redistribute
An example open source knowledge graph for Long Covid Research

- 350 K Clinical Trials
- Pubmed Covid related
- Electronic Medical Records**
- VAERS*

* Adverse vaccine reactions
** Synthea
Now onto JSON-LD

- Knowledge Graphs are getting popular very fast but developers a little bit scared to learn the W3C Semantic Stack.
How do you make it easy as MongoDB to

• Add data to a knowledge graph

• Retrieve data from a knowledge graph

• Validate your data
Solution: JSON-LD

- JSON-LD will help you add, retrieve, validate, and delete objects to a Knowledge Graph as easy as MongoDB
First about JSON: it somehow won

- Messaging:
  - the lingua franca for messaging and data exchange
- Configuration:
  - JSON is replacing XML for configuration of nearly anything
- Document and key/value store:
  - JSON is the main data format stored in Document Stores (Couchbase, Mongo, etc…)
JSON – the good

• Simple standard:
  • Json.org spec is 5 pages, XML spec on W3C = 60 pages 😊
  • only a few datatypes and with arrays!
  • you can make your own complex data types if you want
• Easy to read and parse by humans and machines
• Easy to store in document stores
• Easy to program: support in every programming language
When JSON runs into trouble

- In a hospital setting you might have > 1000 JSON types that stream between various sources and knowledge graphs.
- All objects need to be persisted in a data lake but I don’t want to create silos of document stores.
- If I look at any random JSON object I have to find from the code to see what it means.
- There are graphs in the data but how do I point from one object to the other in a standardized way?
We need our JSON to have Semantics

- Every JSON object needs an Identity that other objects can point to
- Every JSON object needs a Type
- Every Type has an ontology (schema) that describes
  - The attributes and datatypes
  - Links to other types of objects
  - Version information, links to processes and apps that use it
- And a standard Validation Language
JSON-LD = 100 % JSON +

- Add basic **schema support** to JSON: (but SHACL more complete)
- Add **semantics** to JSON objects: what does this attribute mean
- Designed to **link** JSON objects together
- Enables **joins** and **graph search** in document stores

```
What does JSON-LD do?

Unstructured Data

<table>
<thead>
<tr>
<th>Unstructured Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSON-LD</td>
</tr>
</tbody>
</table>

"Data is messy and disconnected. JSON-LD organizes and connects it, creating a better Web."
```
Learn from JSON-LD.ORG

JSON-LD Playground

Play around with JSON-LD markup by typing out some JSON below and seeing what gets generated from it at the bottom of the page. Pick any of the examples below to get started.

NOTE: The playground uses jsonld.js which conforms to JSON-LD 1.0 syntax, API, framing, and errata, the W3C Community Group JSON-LD 1.1 syntax, API, and framing drafts, and partial support of the W3C Working Group JSON-LD 1.1 syntax, API, and framing drafts. Also see the classic JSON-LD 1.0 playground and the RDF Distiller.

Examples: Person Event Place Product Recipe Library Activity

JSON-LD Input

```
{
  "@context": "http://schema.org/",
  "@type": "Person",
  "id": "foaf:person-1",
  "name": "Jane Doe",
  "jobTitle": "Professor",
  "telephone": "(425) 123-4567",
  "url": "http://www.janedoe.com"
}
```

Expanded

<http://xmlns.com/foaf/0.1/person-1> <http://schema.org/jobTitle> "Professor" .
It is everywhere: let’s look at this product
Search for @context in the source
JSON alone would lead to confusion, JSON-LD and SCHEMA.ORG to the rescue

```
{"aggregateRating":
   {"ratingValue":4.6,
    "reviewCount":66},
   "description":"Make peace with your inner hippie while hydrating & protecting against photoaging....Mad Hippie's preservative-free Antioxidant Facial Oil is truly the most natural way to moisturize.",
   "brand":"Mad Hippie",
   "name":"Antioxidant Facial Oil",
   "image":"https://images.ulta.com/is/image/Ulta/2530018",
   "productID":"2530018",
   "offers":
      {"availability":"http://schema.org/InStock",
       "price":"24.99",
       "priceCurrency":"USD"}

{"@context":"http://schema.org",
 "@type":"Product",
 "aggregateRating":
   {"@type":"AggregateRating",
    "ratingValue":4.6,
    "reviewCount":66},
   "description":"Make peace with your inner hippie while hydrating & protecting against photoaging....Mad Hippie's preservative-free Antioxidant Facial Oil is truly the most natural way to moisturize.",
   "brand":"Mad Hippie",
   "name":"Antioxidant Facial Oil",
   "image":"https://images.ulta.com/is/image/Ulta/2530018",
   "productID":"2530018",
   "offers":
      {"@type":"Offer",
       "availability":"http://schema.org/InStock",
       "price":"24.99",
       "priceCurrency":"USD"}
```
**AggregateRating**

**Thing > Intangible > Rating > AggregateRating**

The average rating based on multiple ratings or reviews.

- Canonical URL: [http://schema.org/AggregateRating](http://schema.org/AggregateRating)
- Leave public feedback on this term
- Check for open issues.

<table>
<thead>
<tr>
<th>Property</th>
<th>Expected Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemReviewed</td>
<td>Thing</td>
<td>The item that is being reviewed/rated.</td>
</tr>
<tr>
<td>ratingCount</td>
<td>Integer</td>
<td>The count of total number of ratings.</td>
</tr>
<tr>
<td>reviewCount</td>
<td>Integer</td>
<td>The count of total number of reviews.</td>
</tr>
</tbody>
</table>

**Properties from Rating**

<table>
<thead>
<tr>
<th>Property</th>
<th>Expected Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>author</td>
<td>Organization</td>
<td>The author of this content or rating. Please note that author is special in that HTML 5 provides a special mechanism for indicating authorship via the rel tag. That is equivalent to this and may be used interchangeably. The highest value allowed in this rating system. If bestRating is omitted, 5 is assumed.</td>
</tr>
<tr>
<td>bestRating</td>
<td>Number or Text</td>
<td>A short explanation (e.g. one to two sentences) providing background context and other information that led to the conclusion expressed in the rating. This is particularly applicable to ratings associated with &quot;fact check&quot; markup using ClaimReview.</td>
</tr>
<tr>
<td>ratingExplanation</td>
<td>Text</td>
<td>The rating for the content.</td>
</tr>
</tbody>
</table>
Demo JSON-LD in Python

- Based on crunch base data from early 2000 till 2014
  - Core objects: Investments, acquisitions, investors, companies

- For developers: how can you implement basic CRUD with AllegroGraph JSONLD
  - You can add and retrieve Python dictionaries directly
  - Like many other document databases
  - Objects are indexed with triples but can also be stored as blobs
  - You can retrieve parts of objects in a SPARQL queries
  - And you can retrieve as dictionaries.
Conclusion: JSON-LD and SHACL for Knowledge Graphs

- Make life easier for User Experience and Application Developers that need to work with Knowledge Graphs.

- JSON-LD hides complexity of semantics and graphs

- SHACL easy way to validate new data.
Thank you