Semantic Web Technologies

Topic: Understanding Semantic Web Datasets

Olaf Hartig
olaf.hartig@liu.se
Goal

- Familiarize with a given dataset
- Achieve an initial understanding of the dataset and its structure
  - What types of entities does the dataset describe?
  - What vocabularies are used to represent properties of entities and relationships among them?

- Note, understanding is not about analyzing the data and deriving insights from it
Options

- Ontology visualization
- Exploratory queries
- Dataset summarization and profiling
Ontology Visualization
Diagram Types for Ontology Visualization

- Nested rectangles (treemaps)
  - e.g., OWL-VisMod, Jambalaya
- Nested circles
  - e.g., CropCircles
- Graphs (node-link diagrams)
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
  - Class hierarchy (i.e., sub-class relationships)
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
  - Class hierarchy (i.e., sub-class relationships)
  - Properties-based relationships between classes (i.e., domain and range of properties)
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
  - Class hierarchy (i.e., sub-class relationships)
  - Properties-based relationships between classes (i.e., domain and range of properties)
  - Other relationships between classes (e.g., disjointness)

Screenshot: OWLPropViz
Graph Visualizations of Ontologies

• Most common approach to visualize ontologies
• Tools differ by what elements they illustrate
  – Class hierarchy (i.e., sub-class relationships)
  – Properties-based relationships between classes (i.e., domain and range of properties)
  – Other relationships between classes (e.g., disjointness)
• Rendering of graphs
  – hierarchical
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
  - Class hierarchy (i.e., sub-class relationships)
  - Properties-based relationships between classes (i.e., domain and range of properties)
  - Other relationships between classes (e.g., disjointness)
- Rendering of graphs
  - hierarchical
  - radial
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
  - Class hierarchy (i.e., sub-class relationships)
  - Properties-based relationships between classes (i.e., domain and range of properties)
  - Other relationships between classes (e.g., disjointness)
- Rendering of graphs
  - hierarchical
  - radial
  - force-directed

- Characteristics:
  - Tends to place highly-connected classes to the center
  - All edges have roughly the same length
  - Tends to avoid edge crossings
- e.g., ProtégéVOWL / WebVOWL
Graph Visualizations of Ontologies

- Most common approach to visualize ontologies
- Tools differ by what elements they illustrate
  - Class hierarchy (i.e., sub-class relationships)
  - Properties-based relationships between classes (i.e., domain and range of properties)
  - Other relationships between classes (e.g., disjointness)
- Rendering of graphs
  - hierarchical
  - radial
  - force-directed
  - three-dimensional
## Features of Ontology Visualization Tools

<table>
<thead>
<tr>
<th></th>
<th>Classes</th>
<th>Enumeration</th>
<th>Property restrictions</th>
<th>Cardinality</th>
<th>Intersection</th>
<th>Union</th>
<th>Complement</th>
<th>subClassOf</th>
<th>equivalentClass</th>
<th>disjointWith</th>
<th>Object properties</th>
<th>Datatype properties</th>
<th>Instances</th>
<th>Annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>COE</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>CropCircles</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>FlexViz</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>GLOW</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>GrOWL</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Jambalaya</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>KC-Viz</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Knoocks</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>NavigOWL</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OntoGraf</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OntologyVisualizer</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OntoRama</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OntoSphere3D</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OWLGrEd</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OWLPropViz</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>OWLViz</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>RelFinder</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>SOVA</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>TGViz</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>TopBraid Composer</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>ProtégéVOWL</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>WebVOWL</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>

Balzer, Do, and Maseluk: *Comparison and Evaluation of Ontology Visualizations*. 2015. [http://dx.doi.org/10.18419/opus-3499](http://dx.doi.org/10.18419/opus-3499)
VOWL: Visual Notation for OWL Ontologies

- Comprehensive visual language for representing OWL ontologies
- Graph visualization
- Graphical Primitives:

<table>
<thead>
<tr>
<th>Primitive</th>
<th>Application</th>
<th>Primitive</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>○</td>
<td>classes</td>
<td></td>
<td>datatypes, property labels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>special classes/properties</td>
</tr>
<tr>
<td>(</td>
<td>properties</td>
<td></td>
<td>labels, cardinalities</td>
</tr>
<tr>
<td>△</td>
<td>property directions</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


Visual Elements of VOWL


# VOWL Color Scheme

<table>
<thead>
<tr>
<th>Name</th>
<th>Color</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>blue</td>
<td>classes, object properties, disjointness</td>
</tr>
<tr>
<td>External</td>
<td>blue</td>
<td>external classes and properties</td>
</tr>
<tr>
<td>Deprecated</td>
<td>gray</td>
<td>deprecated classes and properties</td>
</tr>
<tr>
<td>Datatype</td>
<td>yellow</td>
<td>datatypes, literals</td>
</tr>
<tr>
<td>Datatype property</td>
<td>green</td>
<td>datatype properties</td>
</tr>
<tr>
<td>Highlighting</td>
<td>red</td>
<td>circles, rectangles, lines, borders, arrows</td>
</tr>
</tbody>
</table>


ProtégéVOWL
WebVOWL Demo

- Create a VOWL description of the Semantic Sensor Network ontology (SSN)
  
  ```
  java -jar owl2vowl.jar -iri "http://www.w3.org/ns/ssn/
  ```

- Rename the resulting `default.json` file to `ssn.json` and copy it into the `data` directory of WebVOWL

- Add an option for SSN to the `index.html` of WebVOWL

- Open the `index.html` in a browser
WebVOWL Demo
Exploratory Queries
Exploratory Queries

• Idea: issue a number of SPARQL queries to explore the content of a given dataset

• Example: What properties are used in the data?

```sparql
SELECT DISTINCT ?p WHERE {
  ?s ?p ?o
}
```

• What classes are used?

```sparql
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT DISTINCT ?t WHERE {
  ?s rdf:type ?t
}
```
Exploratory Queries (cont'd)

- **What properties are used and how often?**
  
  ```sql
  SELECT ?p (COUNT(?p) AS ?count) WHERE {
  ?s ?p ?o
  }
  GROUP BY ?p
  ORDER BY DESC(?count)
  ```

- **What classes are used and how often?**
  
  ```sql
  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

  SELECT ?t (COUNT(?t) AS ?count) WHERE {
  ?s rdf:type ?t
  }
  GROUP BY ?t
  ```
Exploratory Queries (cont'd)

- *List a few example instances of a particular class*

  ```sql
  PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
  PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>

  SELECT ?s WHERE {
    ?s rdf:type ssn:SensingDevice
  }
  LIMIT 10
  ```
Further Examples

- Bob DuCharme's “Learning SPARQL” (2nd edition!) has an “Exploring the Data” section in Chapter 11
  - [http://learningsparql.com/](http://learningsparql.com/)
  - What classes are declared?
  - What properties are declared?
  - Which classes have instances?
  - What properties are used?
  - Which classes use a particular property?
  - How much is a given property used?
  - How much is a given class used?
  - A given class has lots of instances. What are these things?
  - What data is stored about a class?
  - What values does a given property have?
Dataset Summarization and Profiling
Summarization and Profiling Approaches

- RDFStats (Langegger and Wöß, 2009)
- ExpLOD (Khatchadourian and Consens, 2010)
- LODStats (Auer et al., 2012)
- ProLOD (Böhm et al., 2010)
- ProLOD++ (Ziawasch et al., 2014)
- LODSight (Dudás et al., 2015)
- Loupe (Mihindukulasooriya et al., 2015)
Loupe

- Understand which vocabularies are used (classes and properties), incl. statistics and frequent triple patterns
  - Start from high-level statistics,
  - zoom into details,
  - all the way down to the corresponding triples
- **Class explorer**: Which classes? How many instances? Which properties used by these instances?
- **Property explorer**: Which properties? How many triples? Instances of which classes use a property?
- Online demo: [http://loupe.linkeddata.es/loupe/index.jsp](http://loupe.linkeddata.es/loupe/index.jsp)
- The summary data is obtained by querying the dataset using SPARQL ([http://loupe.linkeddata.es/loupe/methods.html](http://loupe.linkeddata.es/loupe/methods.html))
LODSight

- Visual summary of a dataset as an interactive graph
  - Nodes represent classes
  - Edges represent predicates that connect instances of the classes in the dataset
  - Example instances can be shown in the graph
LODSight

• Visual summary of a dataset as an interactive graph
  - Nodes represent classes
  - Edges represent predicates that connect instances of the classes in the dataset
  - Example instances can be shown in the graph

• Features of the visualization tool:
  - Ontology filter
  - Predicate filter
  - Example instances

• The summary data is obtained by querying the dataset via a SPARQL endpoint (no support for RDF files!)

• Not trivial to set up (but possible, in contrast to Loupe)
RDFStats

- Generates statistical metadata for a dataset by executing several SPARQL queries
  - Dataset may be given in an RDF file or accessed via a SPARQL endpoint
- Generated metadata includes:
  - an URI histogram over URI subjects
  - number of anonymous subjects (blank nodes)
  - a histogram for each property and associated ranges (depending on the ranges of a property, different histograms are available, e.g., integer / double / boolean / date /string histogram)
- Generated statistics captured in RDF using a specific RDFStats vocabulary that is based on SCOVO