Previously on Introduction to Linked Data…

- We have learned:
  - how to read and write RDF,
  - how to access and publish RDF graphs on the web and
  - how to read and write SPARQL queries.

- These technologies (under the name of knowledge graphs) are used in industry

- In short, we could say that we have learned about database technologies (poorly) adapted for the web

- But the goal of the visionaries, in particular of Paul Otlet and Doug Engelbart, was more far-reaching…

The Goal Revisited: A Network of Data and Knowledge

- Interconnected
- Universal and all-encompassing
- To assist humans, organisations and systems with problem solving
- To enable innovation and increased productivity

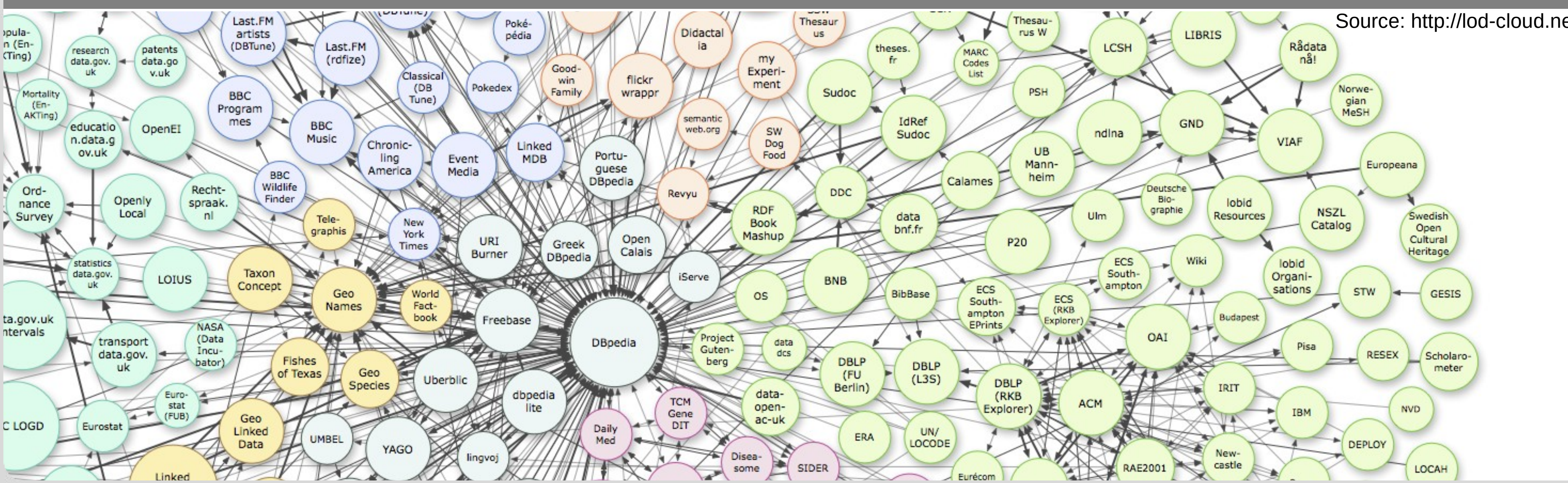
Image from https://www.maxpixel.net/Connection-Network-Connected-News-Web-System-3653380

C08 Vocabulary Descriptions and Data Modelling
How to describe things using identifiers with a shared meaning?

Version 2022-06-21

Lecturer: Prof. Dr. Andreas Harth

CHAIR OF TECHNICAL INFORMATINO SYSTEMS



Source: http://lod-cloud.net

CC - Creative Commons Licensing

- This set of slides is part of the lecture „Semantic Web Technologies“ held at Karlsruhe Institute of Technology
- The content of the lecture was prepared by Andreas Harth based on his book „Introduction to Linked Data“
- The slides were prepared by Benedikt Köhler and updated by Maribel Acosta and Andreas Harth, with additions by Stefan Decker, Lars Gleim and Tobias Käfer

Agenda

1. **Towards a Language of Pure Thought**
    1. Gottlob Frege
    2. Bertrand Russel
    3. Alfred Tarski
    4. Pat Hayes
2. Vocabularies and Ontologies
3. Modeling Individuals, Classes, and Properties
4. The RDFS Vocabulary
5. Domain-specific Vocabularies/Ontologies on the Web

Gottlob Frege
(1848 – 1925)

Language of Pure Thought

- Goal: a notation to describe (abstract) thought – mathematics
- Publication „Über Sinn und Bedeutung“ (1892) („On Sense and Meaning“ or „On Sense and Reference“)
- First step: identify things named via “singular terms”
- Distinguish between:
  - Word (the character string)
  - Sense
  - Reference (the thing in the world)

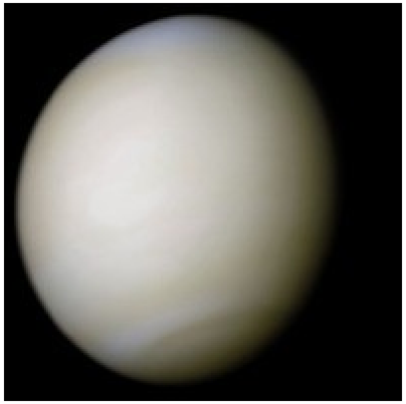
```
Word
 |
Sense
 |
Reference
```

“Now! *That* should clear up a few things around here!”

Gary Larson, The Far Side

Is Morning Star the same as Evening Star?

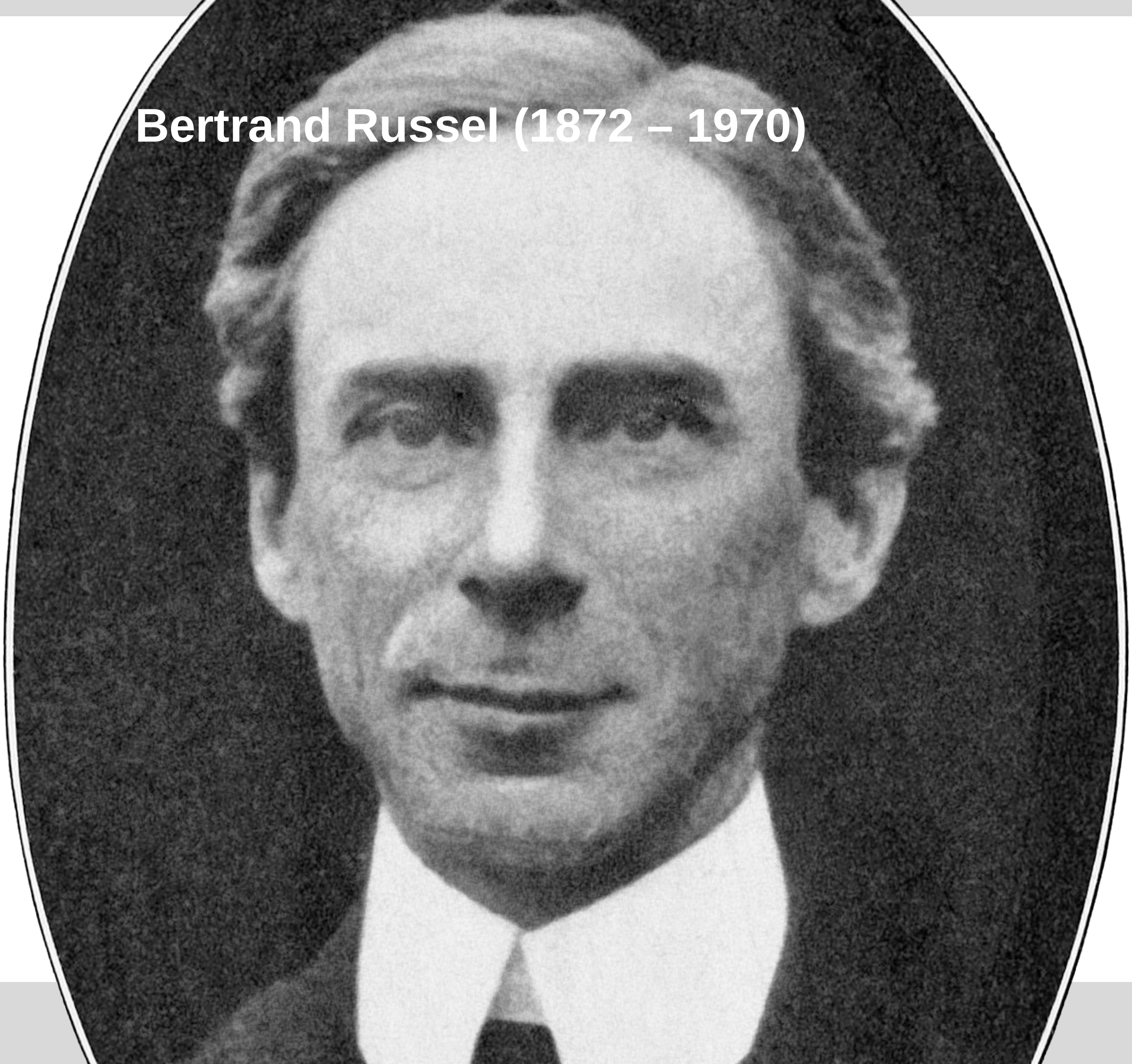
```
    "Morning Star"                    "Evening Star"
          |                                 |
the sense of the expression      the sense of the expression
    "Morning Star"                    "Evening Star"
          |                                 |
  the planet Venus                  the planet Venus
```



From Singular Terms to General Terms

- Basically: instance vs. classes
- Frege developed the word-sense-reference distinction for singular terms (proper names)
- Singular terms refer to exactly one individual thing in the world
- Frege also concerns himself with general terms
- General terms refer to universals (e.g., red, Human)

Bigelow, John C.. Frege exhumes universals. Universals, 1998, doi:10.4324/9780415249126-N065-1. Routledge Encyclopedia of Philosophy, Taylor and Francis, https://www.rep.routledge.com/articles/thematic/universals/v-1/sections/frege-exhumes-universals.


Bertrand Russel (1872 – 1970)

Objective Descriptions of the World

- Only consider the word and the reference (get rid of sense)
- Instead of using names to refer to things in the world,
  use descriptions to refer to things in the world

„Sir Walter Scott was a man“

vs.

„the author of Waverly was a man“

```
"Sir Walter Scott"
        |
the human being Walter Scott

"the author of 'Waverly'"
        |
the human being Walter Scott
```

- But what about: „the present king of France is bald“?

```
"the present King of France"
        |
      ?????
```


Alfred Tarski
(1901 – 1983)

No Connection to the Outer World

- We cannot rely on our sensory input, because our senses deceive us (e.g., optical illusions)
- The big question in philosophy: What can we know? How can we know anything?
- Materialists (reality exists independent of our minds) vs. idealists (reality is only in our minds)
- Tarski sought to develop a theory for the meaning of formal languages, and declared the metaphysical question (materialism vs. idealism) off limits

The Notion of Logical Truth

- Logical languages can give rise to the „liar paradox“: “This sentence is false”
- Tarski‘s solution: distinguish between
  - Object language
  - Metalanguage
- Factor out the statement about truth and falsehood from the object language

- Canonical example for Tarski’s formalisation:

„Snow is white“ is true if an only if snow is white

Object language expression — Metalanguage expression

A Classical Example for Entailment

- Premise: All men are mortal
- Premise: Socrates is a man
- Conclusion: Socrates is mortal

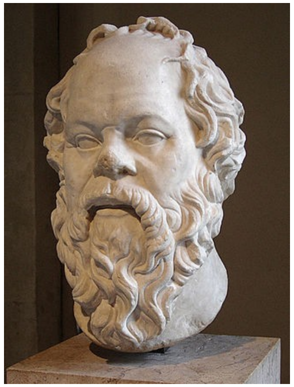


Photo from Wikipedia

- In RDF using RDFS vocabulary:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix : <#> .

:Man rdfs:subClassOf :Mortal .     # premise
:Socrates a :Man .                 # premise
--------------------------------------------------
:Socrates a :Mortal .              # conclusion
```

Pat Hayes
(1944 –)

First Mathematics, Then Physics

- Hayes‘ goal is to represent common-sense knowledge for processing with computers
- The goal: drawing conclusions from given premises, based on Tarskian semantics
- Meaning of RDF graphs should solely be defined via models (mathematical structures), as machines are supposed to process RDF, and machines do not deal well with „social semantics“ as advocated by Tim Berners-Lee in Linked Data
- Naive Physics Manifesto (1978)
- People still work on Hayes’ vision: e.g., Machine Common Sense (2018)

Broad Agency Announcement
Machine Common Sense (MCS)
HR001119S0005
October 19, 2018

Defense Advanced Research Projects Agency
Information Innovation Office
675 North Randolph Street
Arlington, VA 22203-2114

The Conflict

Sarven Capadisli
@csarven

Follow

#LDOW2018 @SchlobachStefan asks:

Who are you?

1. Engineer vs Scientist
2. CS vs Philosopher
3. Hacker vs Mathematician
4. Logician vs Web-ber

So, #LinkedData folks, who are you?

12:27 AM - 23 Apr 2018

Agenda

1. Towards a Language of Pure Thought
2. **Vocabularies and Ontologies**
3. Modeling Individuals, Classes, and Properties
4. The RDFS Vocabulary
5. Domain-specific Vocabularies/Ontologies on the Web

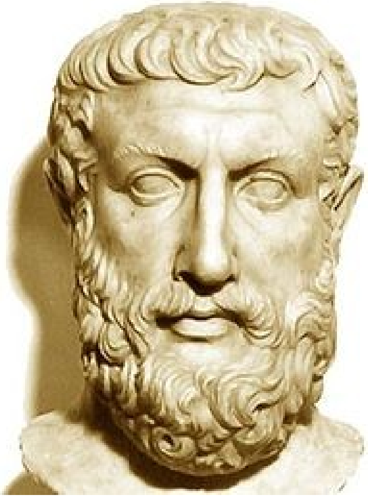
Web Standards

- Data providers publish data on web servers
- Data consumers access data with user agents

- Resource Description Framework
  - Graph-structured data: nodes (URIs, literals, blank nodes) and edges (URIs)
  - Interlink information (relationships)

- How can groups of people use RDF to
  - encode a shared understanding of a domain,
  - organise knowledge in a machine-processable way and
  - give meaning to data that can be exploited?

Ontology in Philosophy



- In philosophy, ontology is a fundamental branch of metaphysics
- Ontology is concerned with the study of the nature of being, existence or reality as such
- Find out what entities and types of entities exist
- Discussed by Aristotle (Socrates), Thomas von Aquin, Descartes, Kant, Hegel, Wittgenstein, Heidegger, Quine, ...

Ontology in Informatics

"An Ontology is a

formal specification > interpretable by machines
of a shared > based on consensus
conceptualisation > describes terminology
of a domain of interest" > models a specific topic

Studer, Benjamins and Fensel (1998) based on Gruber (1993) and Borst (1997)

- An ontology is an engineering artefact, consisting of:
  - A specific vocabulary (set of terms - URIs and literals) used to describe a certain reality, plus
  - A set of explicit assumptions regarding the intended meaning of the vocabulary

What Happens When Modelling (an Ontology)

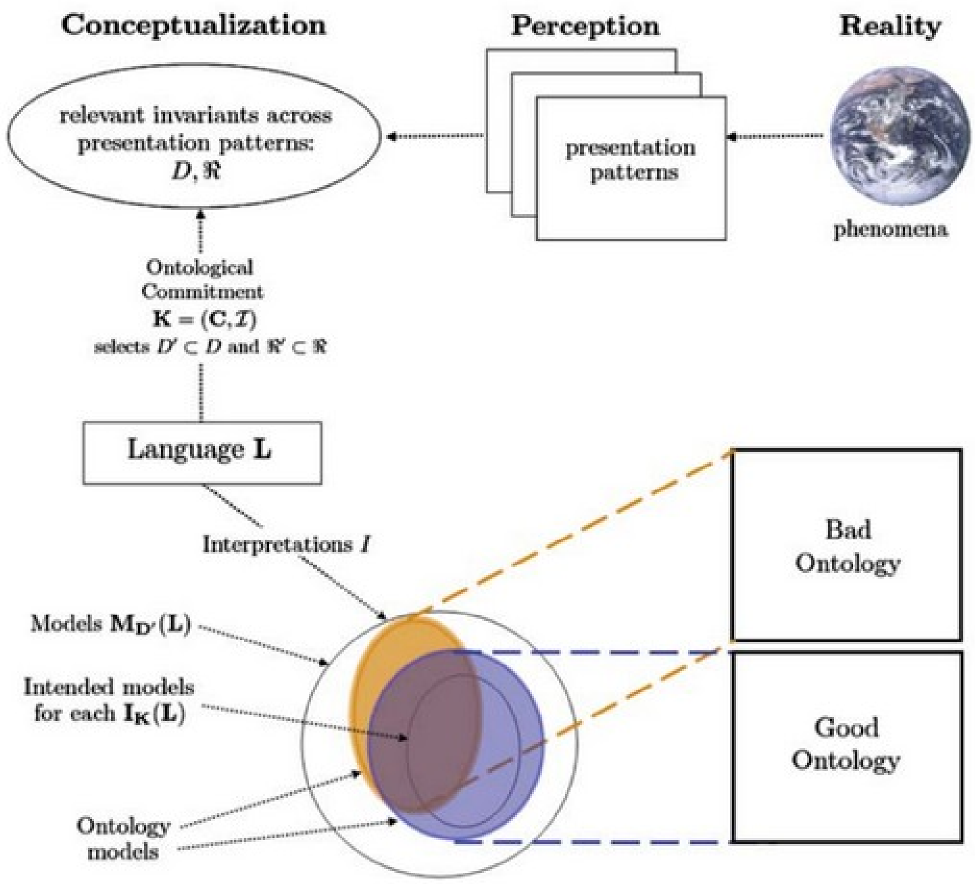
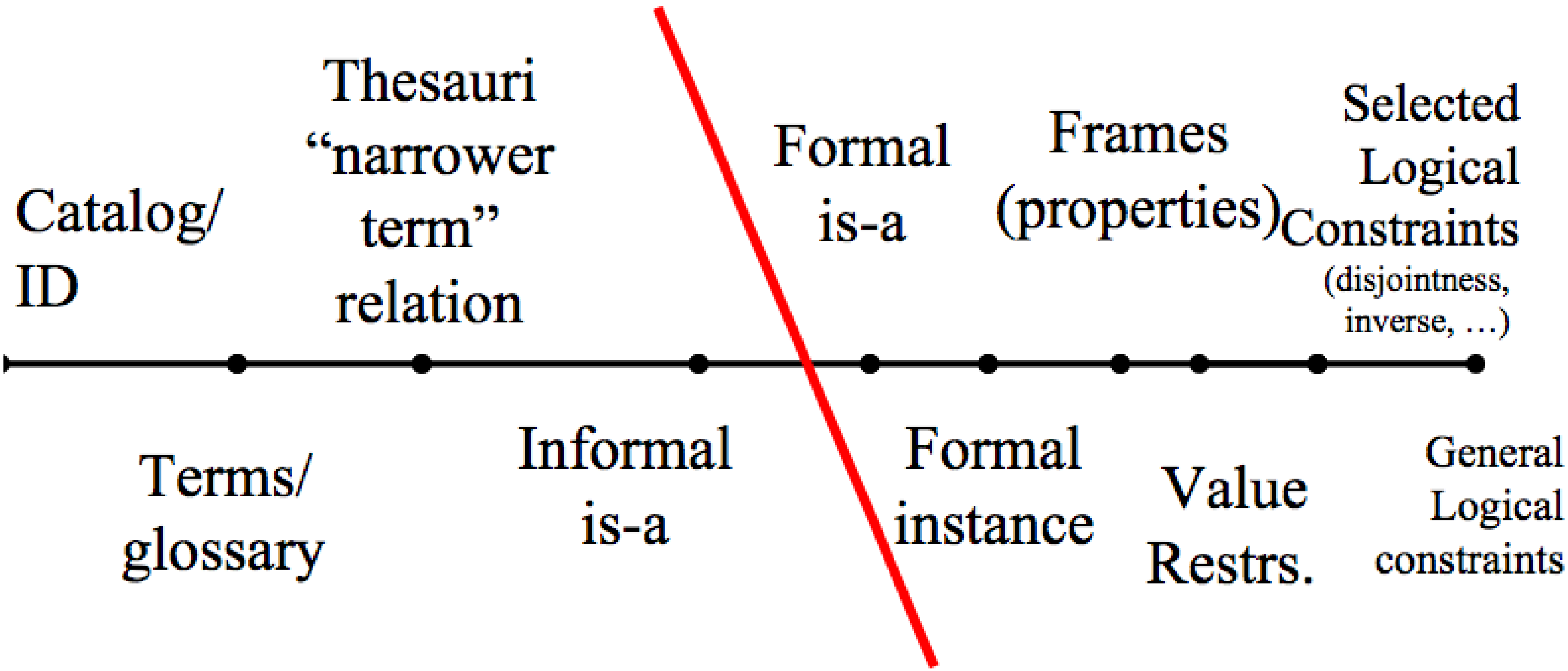


Fig. 2. The relationships between phenomena occurring in reality, their perception (at different times), their abstracted conceptualization, the language used to talk about such conceptualization, its intended models, and an ontology

Guarino, Oberle, Staab. "What is an ontology?." In Staab, Studer (eds.): Handbook on ontologies. Springer, 2009

Ontology Spectrum

From 99 AAAI panel with Gruninger, Lehmann, McGuinness, Ushold, Welty, 2000 Dagstuhl talk by McGuinness



Less expressive

More expressive

Vocabularies and Vocabulary Descriptions/Ontologies

- We call a set of schema-level identifiers (classes and properties) and possibly instance-level identifiers (individuals) a vocabulary.
- We call a vocabulary together with additional descriptions a vocabulary description (or ontology)
- Ontologies can broadly be seen as **agreement** between people on the definition and meaning of the data
- Ontologies are collections of names with defined meaning

Vocabulary Descriptions

- Vocabularies descriptions can include the following:
  - **Individuals:**
    Entities identified via a URI or blank node; a vocabulary description may include descriptions of identity (comes later)
  - **Classes:**
    Sets of individuals identified via URIs or blank nodes; a vocabulary description may include the characteristics of classes
  - **Properties:**
    Properties identified via URIs; a vocabulary description may include the characteristics of properties

Core Semantic Web Vocabularies

- To bootstrap meaning of vocabulary terms, we could use terms that are widely agreed; how about we use mathematics?
- The W3C standardised fundamental vocabularies (based on mathematics) that can be used to express other vocabularies.
- **RDF[1]:** We consider the RDF vocabulary, i.e., the URIs defined as part of the RDF W3C Recommendation.
- **RDFS[2]:** We examine RDF Schema, a simple ontology language that offers means to describe characteristics of classes and properties.

- Throughout the slides, assume the following prefix declarations:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix : <#> .
```

[1] https://www.w3.org/TR/rdf11-primer/
[2] https://www.w3.org/TR/rdf-schema/

ACM Queue

Development

May 13, 2019

Volume 17, issue 2

PDF

Industry-scale Knowledge Graphs: Lessons and Challenges

Five diverse technology companies show how it's done

Natasha Noy, Google; Yuqing Gao, Microsoft; Anshu Jain, IBM Watson; Anant Narayanan, Facebook; Alan Patterson, eBay; Jamie Taylor, Google

Agenda

1. Towards a Language of Pure Thought
2. Vocabularies and Ontologies
3. **Modeling Individuals, Classes, and Properties**
4. The RDFS Vocabulary
5. Domain-specific Vocabularies/Ontologies on the Web

Individuals

- **Individuals** are identified via singular terms
  - Example: India, Karlsruhe, Tim Berners-Lee…
  - On the web, we identify individuals with URIs

- Individuals are described with properties:
  - Example: The German label of India is “Indien”

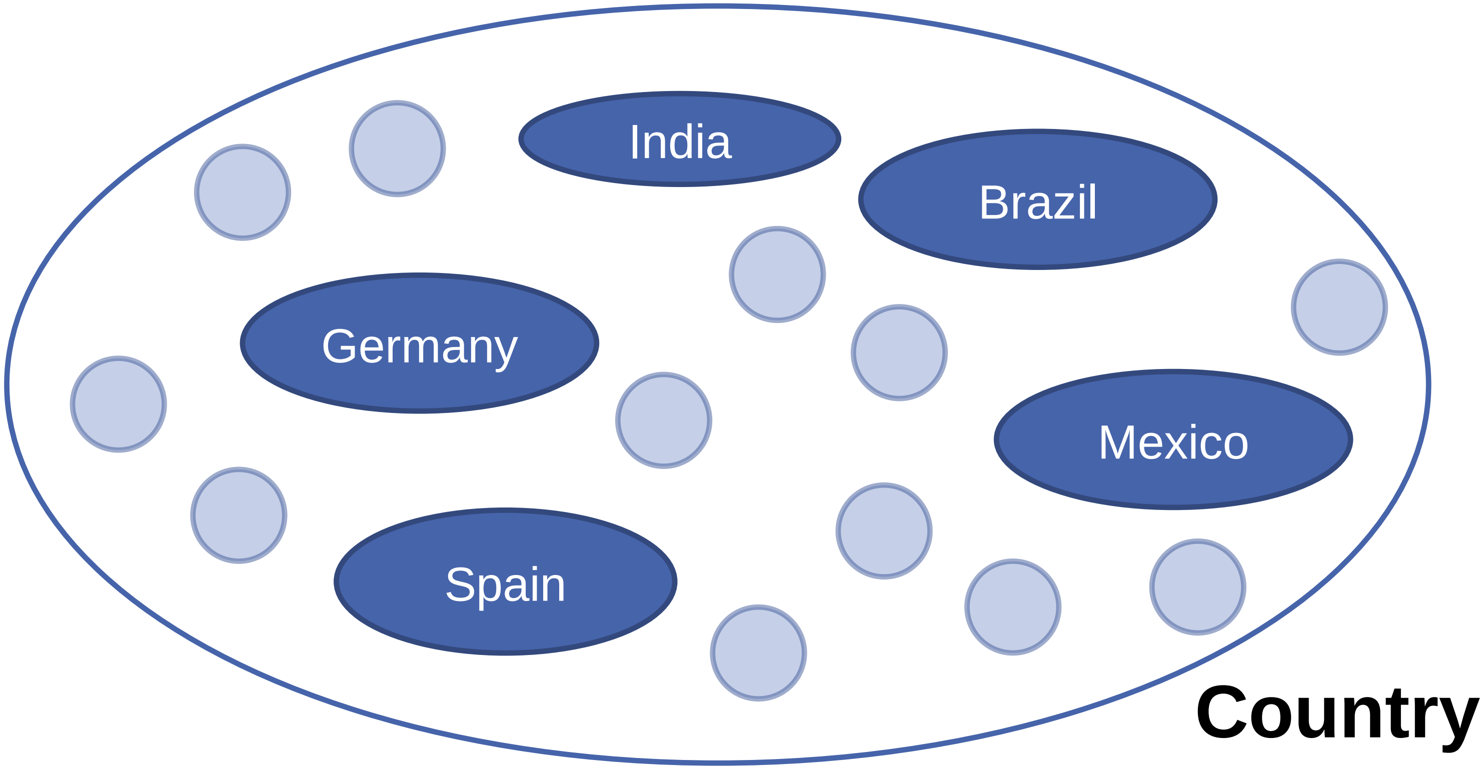
```
@prefix : <#> .

:India   :label   "Indien"@de .
```

Classes

- To be able to talk about groups of individuals, we introduce the notion of a class
  - Example: Country, Person
- A class groups a **set of *things*** with common characteristics
- Thus, we can make statements about the set of things, rather than requiring to make the same statement for each of the things over and over again
- As with individuals, we identify classes with URIs (or blank nodes)

Classes – Analogy to Set Theory

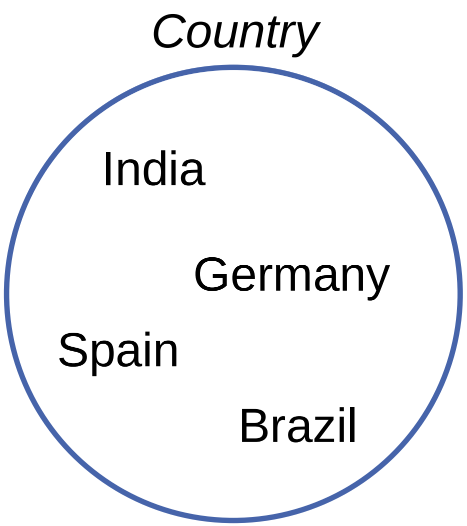


- Individuals represent elements of a set
- Classes represent a set that is identified via a URI or a blank node

Classes: Example (1)

- To define the class:
  - `rdf:type rdfs:Class`
- To relate instances to the class:
  - `rdf:type`

The class of *countries*



URI of the class

```
:Country rdf:type rdfs:Class .
```

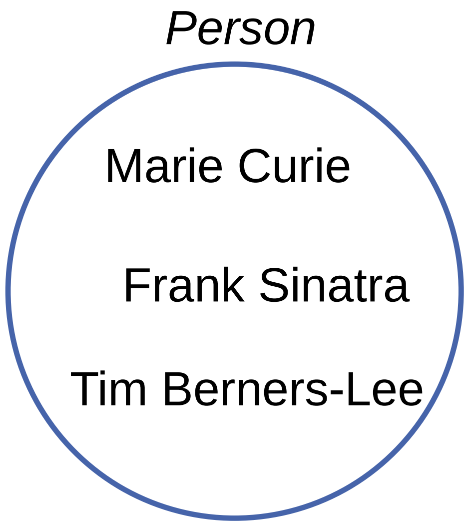
Instances of the class

```
:India   rdf:type :Country .
:Germany rdf:type :Country .
:Spain   rdf:type :Country .
:Brazil  rdf:type :Country .
```

Classes: Example (2)

- To define the class:
  - `rdf:type rdfs:Class`
- To relate instances to the class:
  - `rdf:type`

The class of *persons*



URI of the class ←

```
:Person rdf:type rdfs:Class .
```

Instances of the class

```
:Marie_Curie     rdf:type  :Person .
:Frank_Sinatra   rdf:type  :Person .
:Tim_Berners-Lee rdf:type  :Person .
```

Class Hierarchies

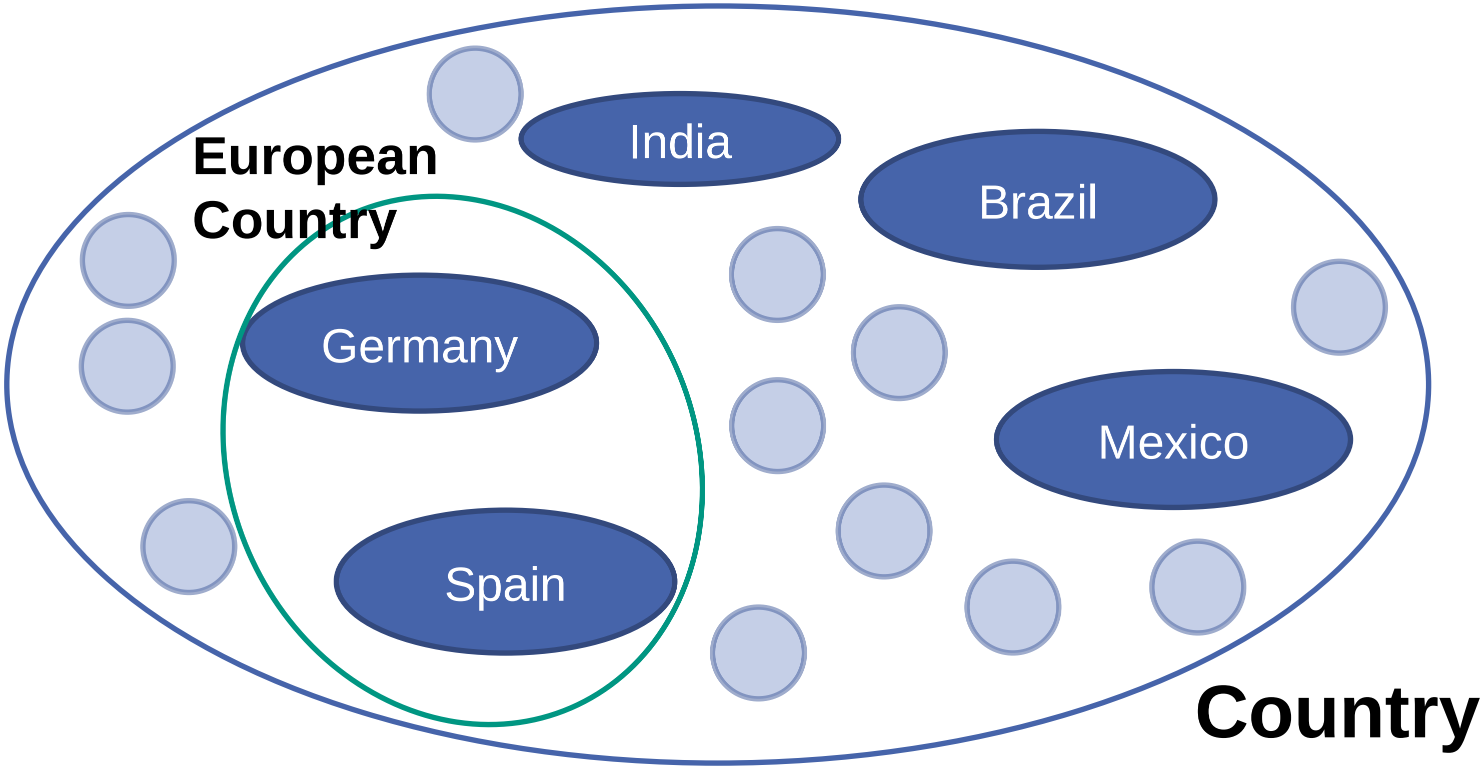
- Given several classes, we can specify a hierarchical relationship between them: the subclass relation

- In RDFS, a class may have several subclasses, and a class can be a subclass of several (super)classes

- Example:
  - We have two classes: `:Country` and `:EuropeanCountry`
  - We want to say that everything that is a European country is also a country
  - That is, `:EuropeanCountry` is a subclass of `:Country`
  - We use `rdfs:subClassOf` to specify the subclass relationship:

```
:EuropeanCountry rdfs:subClassOf :Country .
```

Class Hierarchies – Analogy to Set Theory

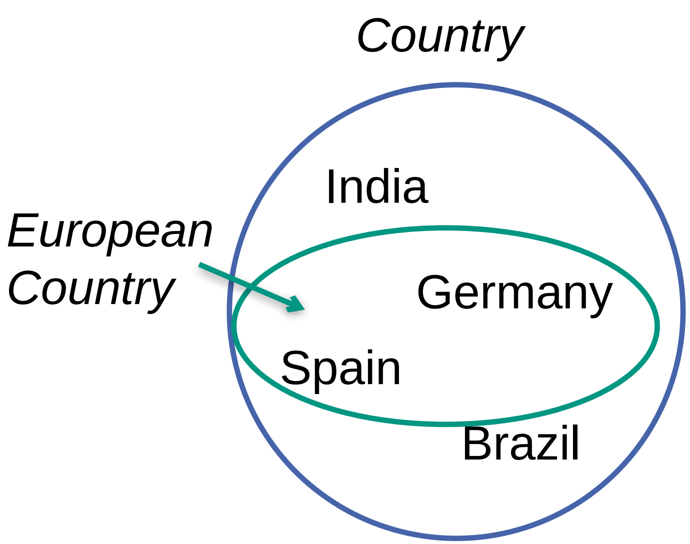


- `rdf:type` corresponds to ∈
- `rdfs:subClassOf` corresponds to ⊆

Class Hierarchies: Example (1)

Model the following statements:

- The set of European countries is a subclass (or specialization) of the set of countries
- From the set of given countries, only Germany and Spain are European Countries

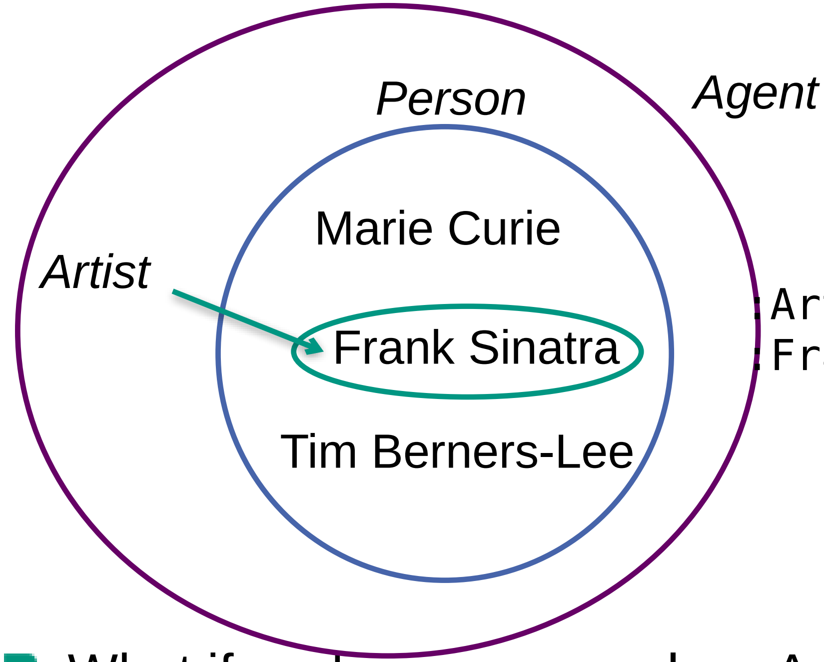



```
:EuropeanCountry rdfs:subClassOf :Country .
:Germany rdf:type  :EuropeanCountry .
:Spain   rdf:type  :EuropeanCountry .
```

Class Hierarchies: Example (2)

Model the following statements:

- The set Artists is a subclass (or specialization) of the set of persons
- From the set of given persons, only Frank Sinatra is an artist




```
:Artist rdfs:subClassOf :Person .
:Frank_Sinatra rdf:type :Artist .
```

- What if we have a superclass Agent that includes the set of persons?

```
:Person rdfs:subClassOf :Agent .
```

Think-Pair-Share

“Rap is something you do, Hip Hop is something you live” [1]

Express the following in RDF using `rdf:type` and `rdfs:subClassOf`:

- Romano is a German hip hop artist.
- Kendrick Lamar is a gangsta rap artist.
- Eminem is a hardcore hip hop artist.
- Gil Scott-Heron is a rap artist.
- Rap music artists are hip hop artists.

[1] KRS-One: “Hip Hop Vs. Rap” (1993). *Digital*. Ed. KRS-One (2003). Los Angeles (CA), USA: Cleopatra Records.
Image from https://www.flickr.com/photos/24102786@N06/4100380091

Properties and Property Hierarchies

- Properties can be formally defined using the class `rdf:Property`
  - Example: `:friendOf` **`rdf:type rdf:Property`** `.`
    `:Dan :friendOf :Tim .`

- Similar to class hierarchies, we can have hierarchies of properties

- With `rdfs:subPropertyOf` we are able to model property hierarchies, for example:

  `:friendOf` **`rdfs:subPropertyOf`** `:knows .`

Property Restrictions

- Often a property only makes sense to be used in conjunction with individuals belonging to a particular class
- For example, a person can be friend of another person:
  it would not make sense to say a document is friend of another document

- Or consider the author relationship: a person can be author of a document

```
:Tim :authorOf :InformationManagementProposal .
```

- Now, we can define the property `:authorOf` so that it only can be used to relate a `:Document` with a `:Person`

```
:authorOf rdfs:domain :Person ;
          rdfs:range :Document .
```

Modelling with RDF and RDFS

Construct	Syntactic form	Description
Class (a class)	s rdf:type rdfs:Class	s (a resource) is an RDF class
Property (a class)	p rdf:type rdf:Property	p (a resource) is an RDF property
type (a property)	s rdf:type o	s (a resource) is an instance of o (a class)
subClassOf (a property)	s rdfs:subClassOf o	s (a class) is a subclass of o (a class)
subPropertyOf (a property)	s rdfs:subPropertyOf o	s (a property) is a sub-property of o (a property)
domain (a property)	s rdfs:domain o	domain of s (a property) is o (a class)
range (a property)	s rdfs:range o	range of s (a property) is o (a class)

[1] Guus Schreiber and Yves Raimond,editors. RDF 1.1 Primer. W3C Working Group Note, 24 June 2014 .
http://www.w3.org/TR/rdf11-primer/#section-vocabulary

Agenda

1. Towards a Language of Pure Thought
2. Vocabularies and Ontologies
3. Modeling Individuals, Classes, and Properties
4. **The RDFS Vocabulary**
5. Domain-specific Vocabularies/Ontologies on the Web

But First: RDF Vocabulary Recap

- The RDF vocabulary contains a few URIs with defined meaning on top of the triple-structured data model[1]
- In other words: RDF is both
  - a data model (the triples) and
  - a set of URIs with defined meaning
- The URIs that are part of the RDF vocabulary (e.g., `rdf:type`) are themselves described in the triple-structured data model

[1] http://www.w3.org/1999/02/22-rdf-syntax-ns

Summary of RDF Vocabulary Terms

- The following table lists all RDF terms, other than the container membership properties `rdf:_1, rdf:_2, rdf:_3 ...`

Class URIs	Property URIs	Datatype URIs	Instance URIs
rdf:Property	rdf:type	rdf:langString	rdf:nil
rdf:List	rdf:first	rdf:HTML	
rdf:Bag	rdf:rest	rdf:XMLLiteral	
rdf:Alt	rdf:value	rdf:PlainLiteral	
rdf:Seq	rdf:subject		
rdf:Statement	rdf:predicate		
	rdf:object		

RDF Schema: Vocabulary Description Language 1.0

- RDF triples can be used to describe resources independent of the nature of the resource URIs
- In practice, RDF triples are used in conjunction with vocabulary terms (URIs with agreed-upon meaning) to describe resources
- Vocabularies are described in RDF triples

- The RDF vocabulary provide very basic modelling constructs
- RDF Schema (RDFS) provides means to specify characteristics of classes and properties

- RDFS is **domain-independent** and allows us to specify the semantics of classes and properties from domain-dependent vocabularies.
- In a sense, RDFS is a “meta-vocabulary” and a “weak” ontology language

[1] Dan Brickley and Ramanathan V. Guha, editors. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, February 2004 . http://www.w3.org/TR/rdf-schema/

rdfs:Class

- `rdfs:Class` can be use to define a class, e.g.,

```
:Person rdf:type rdfs:Class .
```

- `rdfs:Class` is also the class of all classes, so

```
rdfs:Class rdf:type rdfs:Class .
```

rdfs:subClassOf – Class Membership Inference

- Specifying hierarchical relationships can be useful to draw conclusions from given triples
- We will treat this on a theoretical level later, for now consider this example:

```
:Person rdfs:subClassOf :Agent .
```

- Example:

```
:Person rdfs:subClassOf :Agent .
:Tim rdf:type :Person .
```

we can thus conclude that

```
:Tim rdf:type :Agent .
```

rdfs:subClassOf – Reflexivity and Transitivity

- The `rdfs:subClassOf` property is reflexive, that is, every class is a subclass of itself:

```
:Person rdfs:subClassOf :Person .
:Agent rdfs:subClassOf :Agent .
```

- The `rdfs:subClassOf` property is transitive.

- Example of transitivity of "subclass of":

```
       :Artist rdfs:subClassOf :Person .
:Person rdfs:subClassOf :Agent .
```

we can thus conclude that:

```
:Artist rdfs:subClassOf :Agent .
```

rdfs:subPropertyOf –
Reflexivity, Transitivity and Further Inferences

- Similar to `rdfs:subClassOf`, `rdfs:subPropertyOf` is also reflexive and transitive.

- We can also infer new relations between resources that are related via subproperties of other properties.

  - Example:

```
:friendOf rdfs:subPropertyOf :knows.
:Dan :friendOf :Tim .
```

we can thus conclude that:

```
:Dan :knows :Tim.
```

rdfs:domain and rdfs:range – Class Membership Inference

- Using `rdfs:domain` and `rdfs:range`, one can infer the membership of individuals to classes
- Example: Consider the following triples

```
:authorOf rdfs:domain :Person ;
          rdfs:range :Document .

:Tim :authorOf :InformationManagementProposal .
```

From `rdfs:domain`, we can conclude that:

```
:Tim rdf:type :Person .
```

And from `rdfs:range`, we can conclude that:

```
:InformationManagementProposal
rdf:type :Document .
```

Unintended Consequences with Property Restrictions

- Be careful when modelling property restrictions!
- Property restrictions are valid globally and conjunctively
- With the following:

```
:authorOf
rdfs:domain :FictionAuthor , :NonFictionAuthor .
```

- all individuals that are subject of triples with the predicate `:authorOf` are member of both the `:FictionAuthor` and `:NonFictionAuthor` classes

Think-Pair-Share

- Given the following RDF graph:

```
:authorOf rdfs:domain :FictionWriter , :NonFictionWriter .
:Larson :authorOf [ :title "The Girl with the Dragon Tattoo" ]
.
:Halmos :authorOf [ :title "Naïve Set Theory" ] .
```

- What class memberships can we infer for `:Larson` and `:Halmos`?

- What modelling would better capture the intended meaning of the `:authorOf` property and the notion of fiction and non-fiction?

Other RDFS Terms

- There are several other URIS defined in the RDFS vocabulary
  - **`rdfs:label`**: specifies a human-readable label for a resource
  - **`rdfs:comment`**: specifies a human-readable description for a resource
  - **`rdfs:seeAlso`**: specifies a generic link
  - **`rdfs:isDefinedBy`**: relate a resource to its definition. Can be used to indicate a vocabulary in which the resource is defined.
- The property `rdfs:member` represents the super-property of all container membership properties, in detail later
- The `rdfs:member` property has specific characteristics, described both in the W3C recommendation and in the RDF version of the RDFS vocabulary.

Other Predefined RDFS Classes

- (`rdf:Statement` – class of reified RDF triples)
- (`rdf:Property` – class of property resources)
- `rdfs:Resource` – class of resources
- `rdfs:Literal` – class of literal values such as strings or integers
- `rdfs:Container` – the superclass of the RDF container classes, i.e., `rdf:Bag`, `rdf:Seq`, `rdf:Alt`
- `rdfs:Datatype` – class of datatypes; subclass of `rdfs:Literal`; instances: `rdf:langString`, `rdf:HTML`, `rdf:XMLLiteral`, `rdf:PlainLiteral`

- Yes, `rdf:Property`, `rdf:langString`, `rdf:XMLLiteral`… are in the RDF vocabulary already, but only RDFS declares them to be classes.

Summary of RDFS Vocabulary Terms

Class URIs	Property URIs
rdfs:Resource	rdfs:subClassOf
rdfs:Class	rdfs:subPropertyOf
rdfs:Literal	rdfs:domain
rdfs:Container	rdfs:range
rdfs:Datatype	rdfs:member
rdfs:ContainerMembershipProperty	rdfs:label
	rdfs:comment
	rdfs:seeAlso
	rdfs:isDefinedBy

Agenda

1. Towards a Language of Pure Thought
2. Vocabularies and Ontologies
3. Modeling Individuals, Classes, and Properties
4. The RDFS Vocabulary
5. **Domain-specific Vocabularies/Ontologies on the Web**

Use of Domain Vocabularies on the Web

- Vocabularies provide terms that are commonly used on the Semantic Web
- There is a common understanding of their meaning
- The re-use of terms from vocabularies across many web data sources provides shared terminology useful for integration and querying

Dublin Core: DC

METADATA Dublin Core® Metadata Initiative INNOVATION [1]

- The early Dublin Core vocabulary consists of a simple set of properties from the **library community**, mainly used to describe books and other media
- An updated version of Dublin Core introduces several classes, and adds range definitions to the properties[2]
- Dublin Core is useful for applications combining descriptions from multiple libraries (e.g., integrated access to description of books and authors on the VIAF site, see later)

- Commonly used prefix definition:

```
@prefix dcterms: <http://purl.org/dc/terms/> .
```

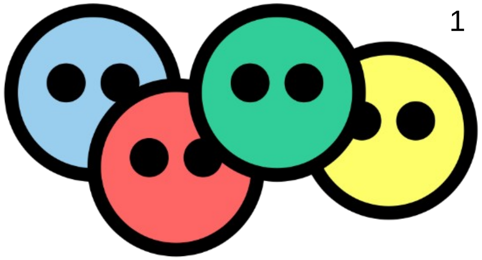
[1] http://dublincore.org/
[2] For detailed info and complete list: http://dublincore.org/documents/2012/06/14/dcmi-terms/

DC – Example

- For example, Paul Otlet’s book could be described in Dublin Core as follows, in a file available at `http://example.org/classification.ttl`:

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix : <#> .
:cdb dcterms:creator :PaulOtlet ;
   dcterms:title "Classification décimale universelle. Études et
projets.
               Les subdivisions communes. Rapport préliminaire."@fr
;
   dcterms:identifier "PPN: 111332591" ;
   dcterms:publisher "Institut international de documentation" ;
   dcterms:issued "1932" .
```

Friend-of-a-Friend: FOAF



1

- FOAF - Friend-of-a-Friend is a vocabulary created by Dan Brickley and Libby Miller to describe **people and related areas**[2]
- It can be used as a vocabulary for machine-readable homepages
- FOAF is one of the earliest domain ontologies on the Semantic Web, and defines less then ten classes and around twenty properties

- Commonly used prefix definition:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

[1] https://de.wikipedia.org/wiki/FOAF
[2] For detailed info and complete specification: http://xmlns.com/foaf/spec/

FOAF – Example

- For example, the person Paul Otlet could be described in FOAF as follows in a file available at `http://example.org/paul-otlet.ttl`:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix pre: <http://example.org/classification.ttl#> .
@prefix : <#> .
:po     a foaf:Person ;
        foaf:givenName "Paul" ;
        foaf:familyName "Otlet" ;
        foaf:img <http://upload.wikimedia.org/Wikipedia
        /en/d/dc/Paul_Marie_Ghislain_Otlet.jpg> ;
        foaf:made [ foaf:page <http://www.worldcat.org/oclc/562162413> ] ,
         pre:cdb .
```

ACM Queue

Databases

December 15, 2015
Volume 13, issue 9

PDF

Schema.org: Evolution of Structured Data on the Web

Big data makes common schemas even more necessary.

R.V. Guha, Google
Dan Brickley, Google
Steve Macbeth, Microsoft

Modeling Domain Vocabularies on the Web

- Before modeling your own vocabulary, check whether you can reuse terms from existing domain vocabularies
- If you have to create new terms, then start with competency questions
  - Competency questions allow to validate whether the resulting vocabulary or ontology covers what it is intended to cover

- Identify classes and their characteristics (e.g., subclasses)

- Identify properties and their characteristics (e.g., domain and range)

- Whenever possible, map (or link) the terms in your vocabulary to terms defined in well-known vocabularies using, for example, `rdfs:subClassOf` or `rdfs:subPropertyOf`.

```
:Person rdfs:subClassOf foaf:Person .
```

Michael Grüninger, Mark S. Fox. Methodology for the Design and Evaluation of Ontologies. 1995.
Natalya F. Noy and Deborah L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. 2001.

Some Practices around Publishing Vocabularies

- **Create resolvable URIs for the terms you defined**
- Decide on Hash vs. Slash URIs
- Consider opaque URIs
- Consider permanent URIs
- **Document your terms in the RDF (rdfs:label, rdfs:comment, ...)**
- Provide licensing information in the RDF
- Register the ontology

See Garijo and Poveda: “Best Practices for Implementing FAIR Vocabularies and Ontologies on the Web”. 2020 https://arxiv.org/pdf/2003.13084.pdf

Recap (1)

- **Vocabularies, Vocabulary descriptions/Ontologies:**
  - Vocabulary: Individuals + Classes + Properties
  - Ontology: formal descriptions of the characteristics of the individuals, classes and properties

- **Modeling with Semantic Web Core Vocabularies:**
  - **Individuals**
  - **Classes**: set of individuals
    - `rdfs:Class`: class of all classes
    - `rdfs:subClassOf`: property to model class hierarchies
    - `rdf:type` to associate resources to classes
  - **Properties**: relations between classes or individuals
    - `rdf:Property`: class of all properties
    - `rdfs:subPropertyOf`: property to model property hierarchies
    - `rdfs:domain` and `rdfs:range`: define features over the types of subjects (domain) and objects (range) of a property

Recap (2)

- **The RDF Vocabulary (see Chapter 3)**
  - `rdf:type` and `rdf:Property`
  - Collections (`rdf:List`): closed set of elements
  - Containers (`rdf:Bag`, `rdf:Seq`, `rdf:Alt`): open set of elements
  - Reification (`rdf:Statement`): triples that refer to other triples
  - n-ary relations using blank nodes, and `rdf:value` to represent main values

- **The RDFS Vocabulary**
  - `rdfs:Class`
  - `rdfs:subClassOf` (reflexive and transitive)
  - `rdfs:subPropertyOf` (reflexive and transitive)
  - `rdfs:domain` and `rdfs:range`
  - Other terms

Inference

Recap (3)

- **Domain-specific Vocabularies/Ontologies**
  - DC: describes books and media items
  - FOAF: describes people and organizations
  - When creating domain-specific vocabularies, map terms to well-known vocabularies, e.g., using `rdfs:subClassOf` or `rdfs:subPropertyOf`

Learning Goals

- G 8.1 Distinguish between classes, properties and individuals in data modelling.
- G 8.2 List and describe the relations that can be expressed using the RDF and RDFS vocabularies between classes and between individuals and classes.
- G 8.3 List and describe the relations that can be expressed using the RDFS vocabulary between properties and between properties and classes.
- G 8.4 Design domain vocabularies in RDFS and relate terms from your vocabulary to terms from existing vocabularies.

Outlook – Chapter 9

- The formal semantics of the RDF and RDFS vocabularies via model theory