Aalto University
School of Science

SeCo

RDF Schema

- Extending RDF with Hierarchies and Constraints

Eero Hyvönen
Aalto University, Semantic Computing Research Group (SeCo) http://seco.cs.aalto.fi
University of Helsinki, HELDIG http://heldig.fi

eero.hyvonen@aalto.fi

W3C®

Learning objectives

Understanding class and property hierarchies in RDF

- for simple ontological modeling

Understanding the idea of domain and value constraints

Understanding the idea of RDF(S) semantics based on logic

Why RDF Schema?

Introducing classes and individuals (instances)

- A class is a set of individuals
  - *E.g., John and Mary are individuals of class Person*

Introducing constraints on using properties

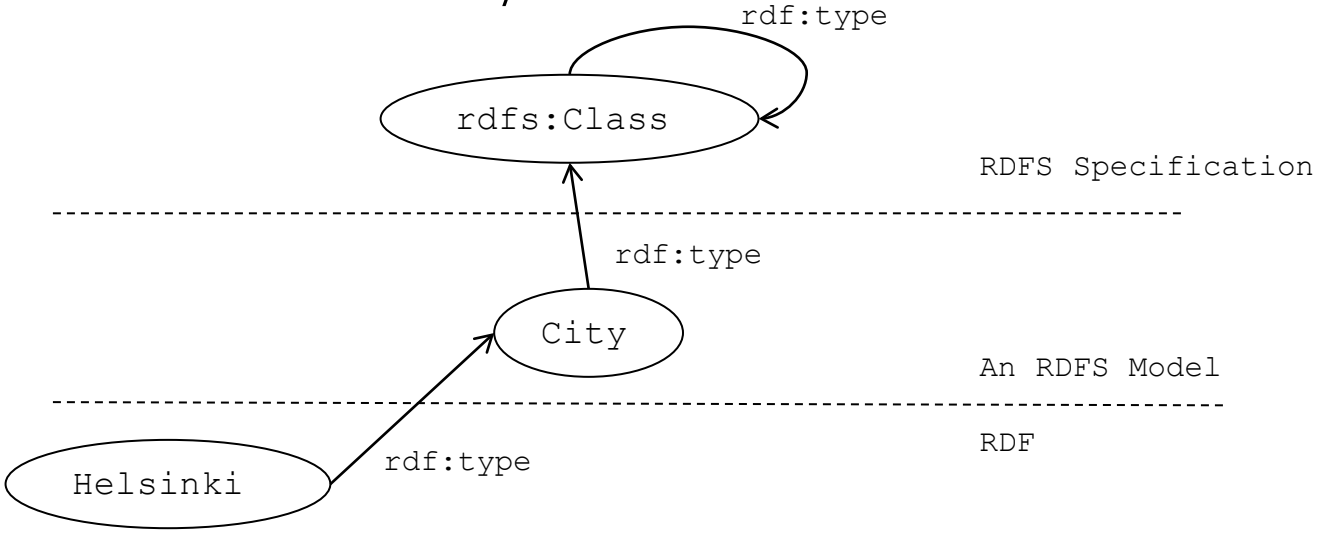
- **Domain** constraint: only individuals of a certain class can have certain properties
  - *E.g., only persons have a social security number*
- **Range** constraint: certain properties may have value of a certain class only
  - *E.g., a person's parent must be a person, too*

Introducing class and property hierarchies

Introducing semantics for validating data and for reasoning

Individuals

- Individual–class relationship is expressed by property **rdf:type**
- Classes are individuals of the (meta)class **rdfs:Class**
- **rdfs:Class** is an instance/individual of itself



Class hierarchies

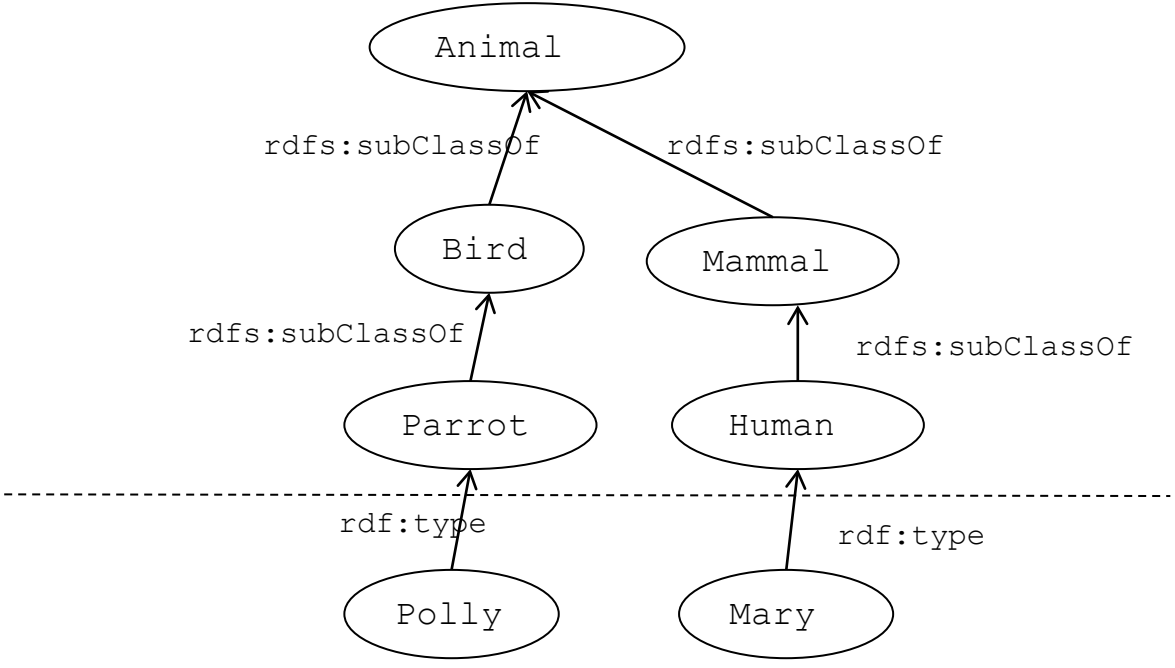
Classes can be organized in hierarchies

- A is a subclass of B if every instance of A is also an instance of B
- Then B is a superclass of A

A class may have multiple superclasses

- Multiple inheritance
- A subclass graph is then not a tree

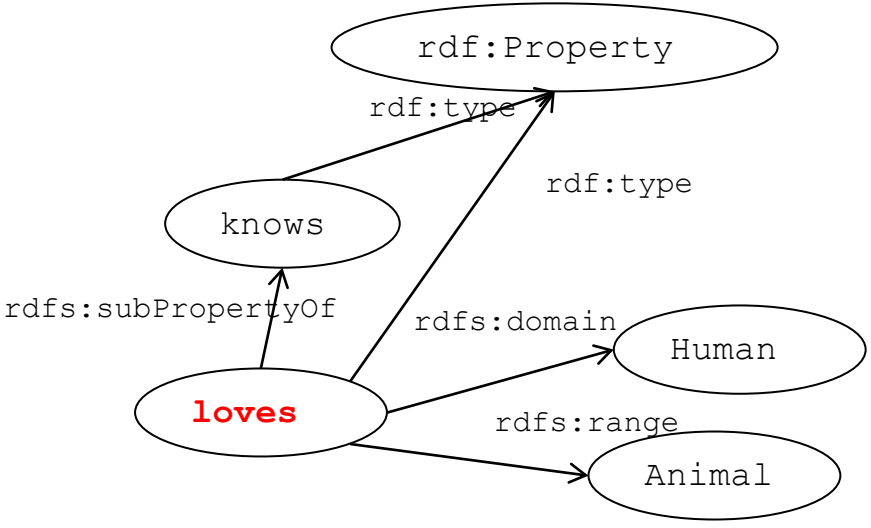
Class hierarchy: rdfs:subClassOf



Properties and constraints

- In RDF(S) properties are first class citizens with identifiers
- Property types are individuals of the class **rdf:Property**
- Properties can form hierarchies using **rdfs:subPropertyOf**
- A property may have domain and/or range constraints, expressed by properties **rdfs:domain** and **rdfs:range**
  - Domain constraints tell where an arc can start
  - Range constraints tell where an arc can point to

RDF Schema with property constraints and hierarchy-based reasoning



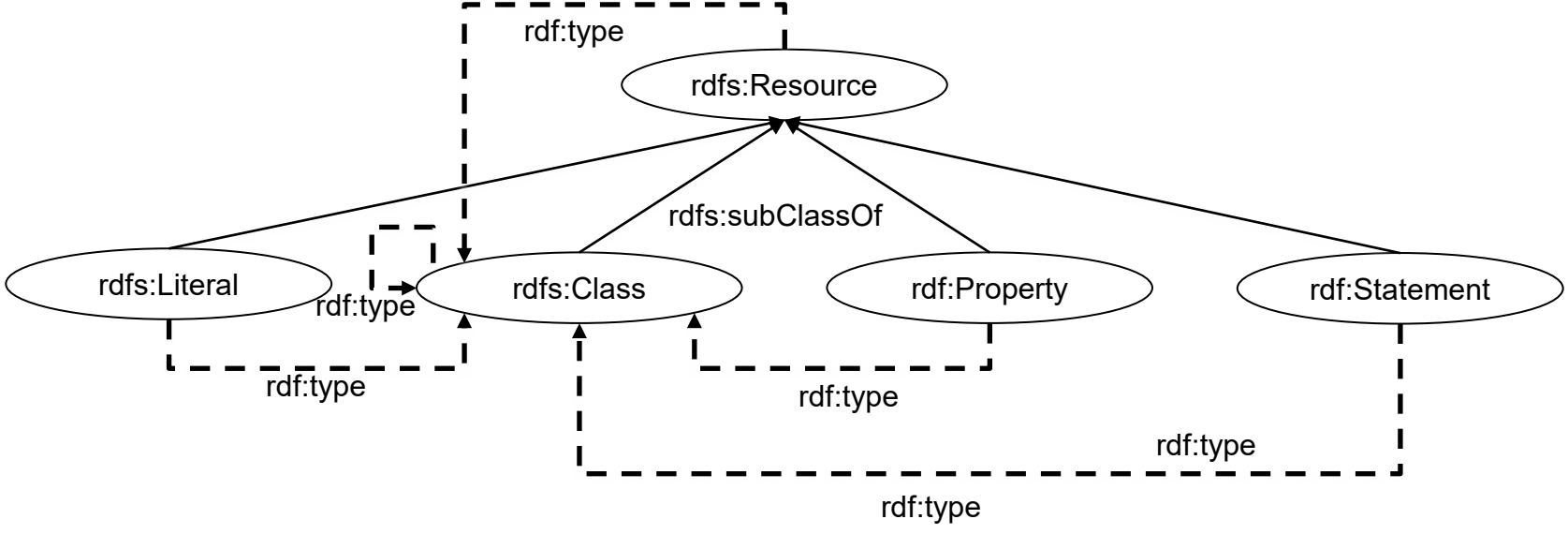
- Constraints: Only humans may love only animals (of any kind)
- Inheritance reasoning:
  - Property hierarchy: Since humans love they also know animals
  - Class hierarchy: Polly (a bird and therefore an animal) can be loved by humans such as Mary

RDF(S) core classes

- **rdfs:Resource**
- **rdfs:Class**
- **rdfs:Literal**
- **rdf:Property**
- **rdf:Statement**

Hierarchy of RDF(S) core classes

These classes form a hierarchy in the RDF(S) specification written by RDF(S) itself!



RDF(S) core properties

- **rdf:type**

- **rdfs:subClassOf**
- **rdfs:subPropertyOf**

- **rdfs:domain**
- **rdfs:range**

Useful utility properties in RDF(S)

Utility properties

- **rdfs:label** human-readable label
- **rdfs:comment** for commenting
- **rdfs:seeAlso** related explaining resource
- **rdfs:isDefinedBy** subproperty of **rdfs:seeAlso**

There are also some other primitives in the specifications

Other constructs in RDF(S)

Reification mechanism

- For adding metadata to individual triples (statements) (like in quads)
- ”John believes Mary loves Polly”
  - ```
    :s1 rdf:type rdfs:Statement;
        rdf:subject :Mary;
        rdf:predicate :loves;
        rdf:object :Polly .
    :John :believe :s1 .
    ```

Collection class rdf:List for representing lists

(:a :b (:c :d) :e)

Container subclasses of rdfs:Container for generic data structures

- **rdf:Bag** Bags (a set with possibly multiple similar members)
- **rdf:Seq** Sequence (resourses in a given order)
- **rdf:Alt** Alternatives (set of resources of which one can be selected)
    ```

Human readable RDF Schema specification:

https://www.w3.org/TR/rdf-schema/

W3C Recommendation

RDF Schema 1.1

W3C Recommendation 25 February 2014

This version:
http://www.w3.org/TR/2014/REC-rdf-schema-20140225/
Latest published version:
http://www.w3.org/TR/rdf-schema/
Previous version:
http://www.w3.org/TR/2014/PER-rdf-schema-20140109/
Editors:
Dan Brickley, Google
R.V. Guha, Google
Previous Editors:
Brian McBride

Please check the **errata** for any errors or issues reported since publication.

This document is also available in this non-normative format: diff w.r.t. 2004 Recommendation

The English version of this specification is the only normative version. Non-normative translations may also be available.

Abstract

RDF Schema provides a data-modelling vocabulary for RDF data. RDF Schema is an extension of the basic RDF vocabulary.

Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This document is an edited version of the 2004 RDF Schema Recommendation. The purpose of this revision is to make this document available as part of the RDF 1.1 document set. Changes are limited to errata, revised references, terminology updates, and adaptations to the introduction. The title of the document was changed from "RDF Vocabulary Description Language 1.0: RDF Schema" to "RDF Schema 1.1". The technical content of the document is unchanged. Details of the changes are listed in the Changes section. Since the edits to this document do not constitute a technical change the Director decided no new implementation report was required.

This document was published by the RDF Working Group as a Recommendation. If you wish to make comments regarding this document, please send them to public-rdf-comments@w3.org (subscribe, archives). All comments are welcome.

This document has been reviewed by W3C Members, by software developers, and by other W3C groups and interested parties, and is endorsed by the Director as a W3C Recommendation. It is a stable document and may be used as reference material or cited from another document. W3C's role in making the Recommendation is to draw attention to the specification and to promote its widespread deployment. This enhances the functionality and interoperability of the Web.

RDF(S) vocabulary specifications for machines in RDF(S)

Namespace IRIs of RDF and RDFS contain the specifications for 1) classes and 2) properties in RDF(S)

- http://www.w3.org/1999/02/22-rdf-syntax-ns#
- http://www.w3.org/2000/01/rdf-schema#

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .

<http://www.w3.org/2000/01/rdf-schema#> a owl:Ontology ;
    dc:title "The RDF Schema vocabulary (RDFS)" .

rdfs:Resource a rdfs:Class ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label "Resource" ;
    rdfs:comment "The class resource, everything." .

rdfs:Class a rdfs:Class ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label "Class" ;
    rdfs:comment "The class of classes." ;
    rdfs:subClassOf rdfs:Resource .

rdfs:subClassOf a rdf:Property ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label "subClassOf" ;
    rdfs:comment "The subject is a subclass of a class." ;
    rdfs:range rdfs:Class ;
    rdfs:domain rdfs:Class .

rdfs:subPropertyOf a rdf:Property ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label "subPropertyOf" ;
    rdfs:comment "The subject is a subproperty of a property."
    rdfs:range rdf:Property ;
    rdfs:domain rdf:Property .

rdfs:comment a rdf:Property ;
    rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
    rdfs:label "comment" ;
    rdfs:comment "A description of the subject resource." ;
    rdfs:domain rdfs:Resource ;
    rdfs:range rdfs:Literal .
```

RDF Scema namespace

http://www.w3.org/2000/01/rdf-schema#

RDF(S) semantics

RDFS semantics

Based on first-order predicate logic

RDF data can therefore be used for reasoning new data

= Adding new triples in the graph

Two ways of defining the same semantics in logic

- Axiomatic semantics by representing RDF constructs in terms of logical axioms
- Direct inference rule-based semantics
  - *Simpler and more intuitive way*

Axiomatic semantics: an example (Semantic Web Primer)

- An RDF statement (triple) **(R, P, V)** is represented using predicate **PropVal(P, R, V)**
- Predicate **Type(R, T)** is a shorthand for **PropVal(rdf:type, R, T)**
- **subClassOf** is a property:

  Type(rdfs:subClassOf, rdfs:Property)

- **type** can be applied to resources and has a class as its value:

  Type(?r, ?c) → (Type(?r, rdfs:Resource) ∧ Type(?c, rdfs:Class))

- If a class C1 is a subclass of a class C2, then all instances of C1 are also instances of C2:

  PropVal(rdfs:subClassOf, ?c1, ?c2) ↔
  (Type(?c1, rdfs:Class) ∧ Type(?c2, rdfs:Class) ∧
  ∀?x (Type(?x, ?c1) → Type(?x, ?c2)))

RDFS semantics based on inference rules

Semantics in terms a rule system instead of restating RDF in terms of first-order logic

Rule system consists of inference rules of the form:

IF	**Graph E contains certain triples**
THEN	**add to E certain additional triples**

where E is a set of RDF triples

Examples of inference rules

IF E contains the triple (?x, ?p, ?y)
THEN E also contains (?p, rdf:type, rdf:Property)

IF E contains the triples (?u, rdfs:subClassOf, ?v)
and (?v, rdfs:subclassOf, ?w)
THEN E also contains the triple (?u, rdfs:subClassOf, ?w)

IF E contains the triples (?x, rdf:type, ?u)
and (?u, rdfs:subClassOf, ?v)
THEN E also contains the triple (?x, rdf:type, ?v)

Examples of inference rules (2)

Any resource **?y** which appears as the value of a property **?p** can be inferred to be a member of the range of **?p**

- This shows that range definitions in RDF Schema are not used to restrict the range of a property, but rather to infer the membership of the range

IF **E contains the triples (?x, ?p, ?y)**
and (?p, rdfs:range, ?u)
THEN **E also contains the triple (?y, rdf:type, ?u)**

Summary

- RDF Schema provides a mechanism for describing concepts of specific applications domains
  - *RDF Schema is a primitive ontology language*
- Key concepts of RDF (Schema) are
  - *Classes and instances*
  - *Type and subclass relations for class hierarchies*
  - *Property and subproperty relations for property hierarchies*
  - *Domain and range restrictions connecting properties and classes*