# Lab: Ontology Engineering (1)

Camilo Thorne

Room 00.012
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
+49 (0) 711 685-81369
camilo.thorne@ims.uni-stuttgart.de

Semantic Web, SS 2017
(based on slides by W. Kessler)

Institut für Maschinelle Sprachverarbeitung

# Outline



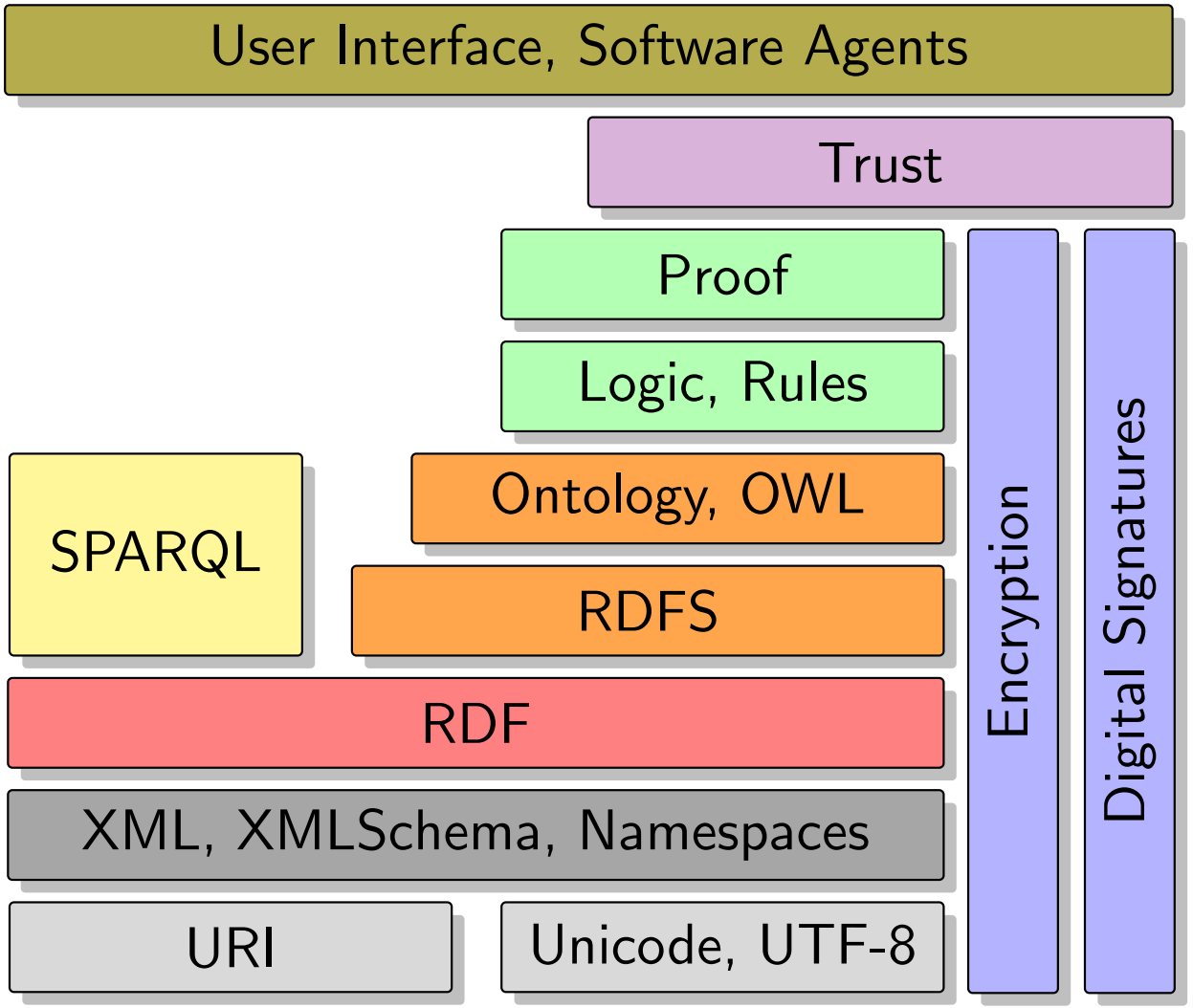
Institut für Maschinelle Sprachverarbeitung

# Outline



Institut für Maschinelle Sprachverarbeitung

# The Semantic Web Stack [W3C, Tim Berners-Lee]




Institut für Maschinelle Sprachverarbeitung

# RDFS

- RDFS lets the user specify and constrain the **semantics** of the vocabulary used in RDF data models
- Distinguishes three types of resources:

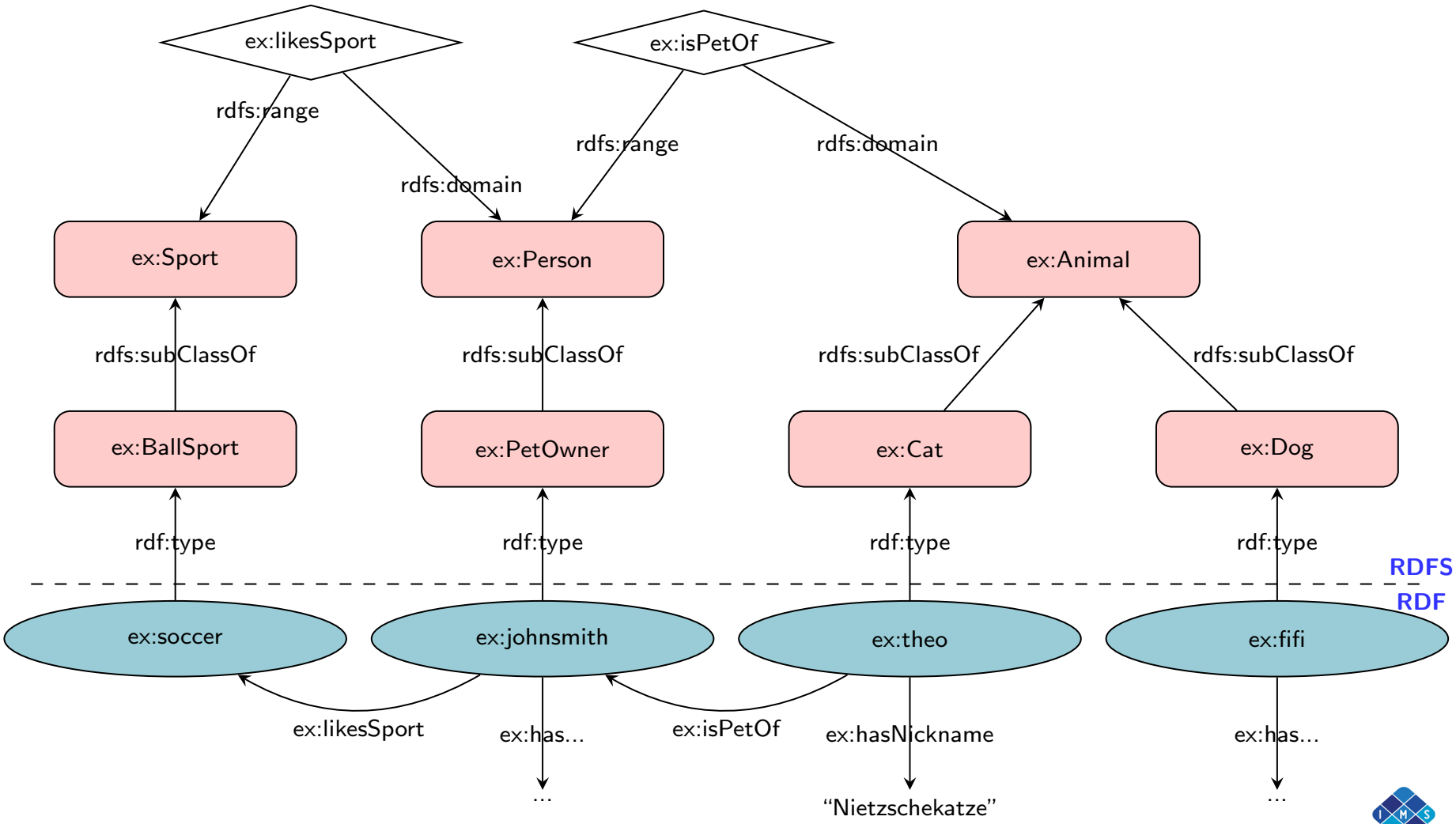
| | |
|---|---|
| **instances** | individual objects |
| **classes** | types of objects, templates (`rdfs:Class`) |
| **properties** | describe relations between resources `rdfs:Property` |

- Introduces domain-independent **logical relations** with a fixed semantics:

| | |
|---|---|
| `rdf:type` | type of an instance (class membership) |
| `rdfs:subClassOf` | defines a class hierarchy |
| `rdfs:subPropertyOf` | defines a property hierarchy |
| `rdfs:domain` | the class of possible subjects for the property |
| `rdfs:range` | the class of possible objects for the property |

Institut für Maschinelle Sprachverarbeitung

# RDF(S) Ontologies




Institut für Maschinelle Sprachverarbeitung

# Outline



Institut für Maschinelle Sprachverarbeitung

# Disclaimer

- There is no one correct way to model a domain – there are always alternatives that may be just as good
- The best solution almost always depends on the application that you have in mind and the extensions that you anticipate
- Ontology development is an **iterative process**, the following steps are not a linear guide but will have to be repeated

Institut für Maschinelle Sprachverarbeitung

# Names of Resources

- It is important to distinguish between a **resource** and its **name**
- Resources represent entities, properties and classes in the domain, and not the words that denote these concepts
- Different word senses should be modeled as different resources
- Names can be pseudo-words, after all it's just a reference to a concept

Institut für Maschinelle Sprachverarbeitung

# Naming Conventions

- Distinguish between:

  1. **Instances:** use lower-cased names as `ex:theo`, ...
  2. **Properties:** use camel-cased lower-upper-case combinations as `ex:likesSport`, ...
  3. **Classes:** use camel-cased upper-cases as `ex:PetOwner`, `ex:Dog`, ...
  4. **Domain vs. logical properties:** `ex:likesSport` vs. `rdf:type`, ...

- Use only singular or plural names, do not mix, usually singular names are more intuitive
- Avoid abbreviations
- Some people include domain and range as part of the property name, e.g., `isPetOfPersonDog`

Institut für Maschinelle Sprachverarbeitung

# Naming Conventions

- Distinguish between:

  1. **Instances:** use lower-cased names as `ex:theo`, ...
  2. **Properties:** use camel-cased lower-upper-case combinations as `ex:likesSport`, ...
  3. **Classes:** use camel-cased upper-cases as `ex:PetOwner`, `ex:Dog`, ...
  4. **Domain vs. logical properties:** `ex:likesSport` vs. `rdf:type`, ...

- Use only singular or plural names, do not mix, usually singular names are more intuitive
- Avoid abbreviations
- Some people include domain and range as part of the property name, e.g., `isPetOfPersonDog`

**Golden rule:** adhere to your own convention consistently!

Institut für Maschinelle Sprachverarbeitung

# Outline



Institut für Maschinelle Sprachverarbeitung

# Overview of Steps

1. **Preparation**
   - Step 1: Determine the domain and scope of the ontology
   - Step 2: Consider reusing existing ontologies
   - Step 3: Enumerate important terms in the ontology
2. **Light-weight ontology** (RDFS)
   - Step 4: Define classes and the class hierarchy
   - Step 5: Define properties and their domain/range
3. **Instances**
   - Step 6: Create instances

Institut für Maschinelle Sprachverarbeitung

# Step 1: Determine the Domain and Scope of the Ontology

The first step in the creation of an ontology is actually the definition of the task we want to solve:

## Domain and Scope

1. What is the domain that the ontology will cover?
2. For what purpose are we going to use the ontology?
3. Who will use and maintain the ontology?

It is never possible to model **all** aspects of a domain, we need to determine what are the important aspects we need to cover

Institut für Maschinelle Sprachverarbeitung

# Step 2: Consider Reusing Existing Ontologies

- Places to look for existing ontologies:
  - DAML ontology library (http://www.daml.org/ontologies/)
  - Protege Ontology Library (http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library)
  - SchemaWeb (http://www.schemaweb.info/)
  - etc., ...
- Big ontologies:
  - DBPedia (http://wiki.dbpedia.org/)
  - OpenCyc (http://sw.opencyc.org/)
  - Yago (http://www.mpi-inf.mpg.de/yago-naga/yago/)
  - SUO (http://suo.ieee.org/)
  - etc., ...

Institut für Maschinelle Sprachverarbeitung

# Step 3: Important Terms and Questions

- It is useful to write down a list of all terms we would like to make statements about or to explain
- At this step do not worry about overlaps in the concepts or what relations terms have to each other
- **Semantics comes first:** draw the graph model we want to represent or describe with the (RDFS) ontology

Institut für Maschinelle Sprachverarbeitung

# Step 3: Important Terms and Questions

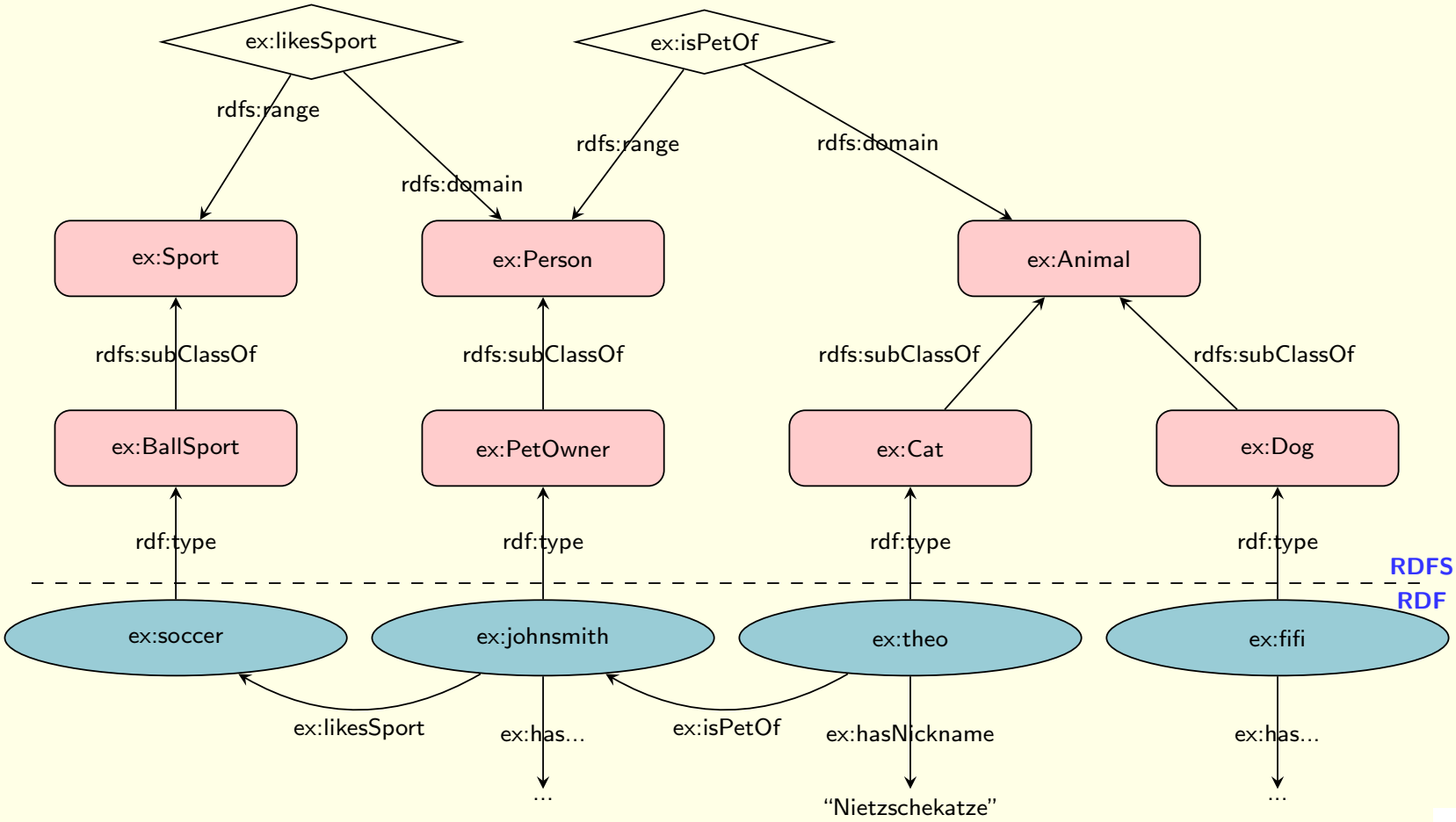
- It is useful to write down a list of all terms we would like to make statements about or to explain
- At this step do not worry about overlaps in the concepts or what relations terms have to each other
- **Semantics comes first:** draw the graph model we want to represent or describe with the (RDFS) ontology

▷ An alternative way is to sketch a list of questions that an application should be able to answer based on the ontology (think of DB design)

Institut für Maschinelle Sprachverarbeitung

# Exercise 1: Modeling

Discuss the running example:



# Step 4: Define the Classes and the Class Hierarchy

- Definition of a hierarchy can start with the most general classes (top-down), the least general classes (bottom-up) or the most salient concepts of the domain
- Organize the classes into a taxonomy by asking if by being an instance of one class, a concrete entity will necessarily (i.e., by definition) be an instance of some other class
- Different parallel hierarchies are possible, classes are not disjoint by default

Institut für Maschinelle Sprachverarbeitung

# Step 5: Define Properties and their Domain/Range

- Many of the terms from the list created in step 3 that are not classes will likely be **properties**
- For each property in the list, we must determine:
  1. to which class it applies: determine its domain
  2. from which class it takes its values: determine its ranges
- A class where only part of its subclasses are possible subjects with the property, is not a good domain (range)
- Properties should have as a domain (range) the most general class possible rather than a long conjunction of very special classes

Institut für Maschinelle Sprachverarbeitung

# Step 6: Create Instances

- In a typical ontology the number of instances is many orders of magnitude larger than the number of classes
- Instances can be created manually, by extraction from structured sources (a data base) or by information extraction from unstructured text
- Defining an instance of a class requires choosing a class and filling in the values for properties

Institut für Maschinelle Sprachverarbeitung

# Outline



Institut für Maschinelle Sprachverarbeitung

# Protégé

- We really don't want to write RDF/RDFS by hand all the time
- Thankfully, there exist tools to help us!
- Protégé[1] is a well-known and powerful ontology editor
- Protégé is free, open source and implemented in Java
- Protégé is maintained by the Stanford Center for Biomedical Informatics Research and the community


[1] http://protege.stanford.edu/

Institut für Maschinelle Sprachverarbeitung

# Protégé

- The different parts of the ontology you are looking at are shown in different tabs (views)
- You choose which tabs to show/hide in “Window” – “Tabs”
- Protégé allows plugins to add their own tabs

Institut für Maschinelle Sprachverarbeitung

# Protégé

- The different parts of the ontology you are looking at are shown in different tabs (views)
- You choose which tabs to show/hide in "Window" – "Tabs"
- Protégé allows plugins to add their own tabs

We will look at more advanced features of Protégé once we look at more powerful ontology languages/semantic web standards

Institut für Maschinelle Sprachverarbeitung

# Exercise 2: Create Your Own Ontology

- Create a light-weight ontology with instances
- It is not important that your ontology is complete, more important is the discussion about design decisions you have to make and any doubts or questions that arise
- Serialize the RDFS ontology into turtle and check for its validity using an RDF validator
- What are the pros and cons of Protégé?

Institut für Maschinelle Sprachverarbeitung

# Outline



Institut für Maschinelle Sprachverarbeitung

# Suggested Reading


Boris Villazón-Terrazas, Nuria García-Santa, Yuan Ren, Alessandro Faraotti, Honghan Wu, Yuting Zhao, Guido Vetere, Jeff Z. Pan: Knowledge Graph Foundations. In: Exploiting Linked Data and Knowledge Graphs in Large Organisations, 2017. (pp. 17-55) https://link.springer.com/chapter/10.1007%2F978-3-319-45654-6_2

Stefan Decker, Sergey Melnik, Frank Van Harmelen, Dieter Fensel, Michel Klein, Jeen Broekstra, Michael Erdmann, and Ian Horrocks. The semantic web: The roles of XML and RDF. In: IEEE Internet computing 4, no. 5, 2000. (pp 63-73) http://ieeexplore.ieee.org/abstract/document/877487/


Institut für Maschinelle Sprachverarbeitung