Web Ontology Language OWL

OWL (Web Ontology Language)

OWL is an ontology language standard for web applications of ontologies (the semantic web). However, OWL is used in "web-independent" applications as well.

- OWL 1 language is a W3C Recommendation since February 2004.
- OWL 2 language is a new version, it is a W3C Recommendation since October 2009.

The World Wide Web Consortium (W3C), founded by Tim Berners-Lee, is an international community that develops standards for the Web. Another example of a W3C Recommendation is XML.

OWL 语言更偏向应用

Overview

From the overview of OWL2: The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide

- classes, ⟷ DL
- properties, 部分与DL对应
- individuals, and
- data values

and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents.

OWL 2 has five languages: OWL2 Full, OWL2 DL, and the profiles OWL2 QL, OWL2 EL, and OWL2 RL.

OWL2 Full and OWL2 DL

- **OWL Full**
  - very expressive language;
  - slightly problematic semantics;
  - is fully upward-compatible with RDF (syntactically and semantically):
    - any legal RDF document is also a legal OWL Full document
    - any valid RDF/S conclusion is also a valid OWL Full conclusion
  - is **undecidable** (no complete (or efficient) reasoning support)
- **OWL DL**
  - is a sublanguage of OWL Full corresponding approximately to $\mathcal{SHOIQ}$ with XML datatypes;
  - often permits reasonably efficient reasoning support (but not tractable).

OWL 2 Profiles

OWL 2 Profiles are sub-languages of OWL 2 that have advantages in particular application scenarios.

- OWL 2 EL is based on the description logic $\mathcal{EL}$. It enables polynomial time algorithms for all the standard reasoning tasks; it is particularly suitable for applications where very large ontologies are needed, and where expressive power can be traded for performance guarantees.
- OWL 2 QL is based on description logics similar to DL-Lite. It enables conjunctive queries to be answered in LogSpace (more precisely, AC0) using standard relational database technology; it is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to access the data directly via relational queries (e.g., SQL).

OWL 2 Profiles

- OWL 2 RL enables the implementation of polynomial time reasoning algorithms using rule-extended database technologies operating directly on RDF triples; it is particularly suitable for applications where relatively lightweight ontologies are used to organize large numbers of individuals and where it is useful or necessary to operate directly on data in the form of RDF triples.

三元组

namespace URI、唯一标识符，可以是个网页，也可以是网页上的某些内容

The OWL language

There are different syntactic forms of OWL:

- **RDF's XML-based** syntax (primary syntax for OWL)
- an **XML-based** syntax that does not follow the RDF conventions

  (more easily read by human users) see `http://www.w3.org/TR/owl-xmlsyntax/`
- an **abstract** syntax (used in the language specification document)

  (much more compact and readable) see `http://www.w3.org/TR/owl-semantics/`
- a graphic syntax based on the conventions of **UML**

  (Unified Modelling Language)

  (an easy way for people to become familiar with OWL)
- ... (?)

XML/RDF-based: OWL ontologies (header)

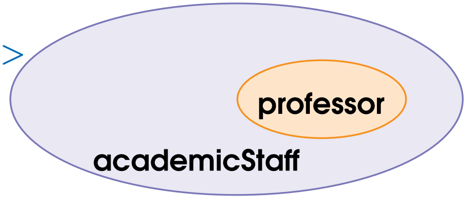
```
<rdf:RDF  xmlns:owl="http://www.w3.org/2002/07/owl#"
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
          xml:base="http://www.dcs.bbk.ac.uk/">

    <owl:Ontology  rdf:about="">
        <rdfs:comment>An example OWL ontology</rdfs:comment>
        <owl:priorVersion  rdf:resource="http://www.dcs.bbk.ac.uk/uni-old-ns"/>
        <owl:imports  rdf:resource="http://www.dcs.bbk.ac.uk/person"/>
        <rdfs:label>SCSIS Ontology</rdfs:label>
    </owl:Ontology>

    ...

</rdf:RDF>
```

XML/RDF-based: The OWL language (classes)

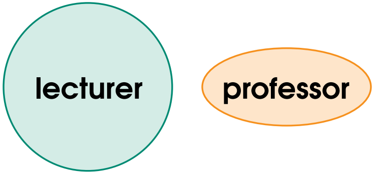
Classes are defined using an **owl:Class** element

(**owl:Class** is a subclass of **rdfs:Class**)

```
<owl:Class rdf:ID="professor">
  <rdfs:subClassOf rdf:resource="#academicStaff"/>
</owl:Class>
```




```
<owl:Class rdf:about="#professor">
  <owl:disjointWith rdf:resource="#lecturer"/>
</owl:Class>
```



Instead of going through XML/RDF-based syntax, a bit more about abstract syntax (for obvious reasons).

OWL: abstract syntax

For details see `http://www.w3.org/TR/owl-semantics/syntax.html#2.3.2.1`

OWL constructs for classes vs DL concepts

OWL construct	DL	Example
owl:Thing	$\top$	
owl:Nothing	$\bot$	
intersectionOf($C_1 \ldots C_n$)	$C_1 \sqcap \cdots \sqcap C_n$	**Human** $\sqcap$ **Male**
unionOf($C_1 \ldots C_n$)	$C_1 \sqcup \cdots \sqcup C_n$	**Doctor** $\sqcup$ **Lawyer**
complementOf(C)	$\neg C$	$\neg$**Male**
oneOf($a_1 \ldots a_n$)	$\{a_1, \ldots, a_n\}$	{**john, mary**}
restriction(r allValuesFrom(C))	$\forall r.C$	$\forall$**hasChild.Doctor**
restriction(r someValuesFrom(C))	$\exists r.C$	$\exists$**hasChild.Doctor**
restriction(r minCardinality(C))	$\geq n\ r.C$	≥ 2 **hasChild.Lawyer**
restriction(r maxCardinality(C))	$\leq n\ r.C$	≤ 2 **hasChild.Lawyer**
restriction(r value(a))	$\exists r.\{a\}$	$\exists$**citizen_of**.{**France**}

and XML Schema datatypes: int, string, real, etc.

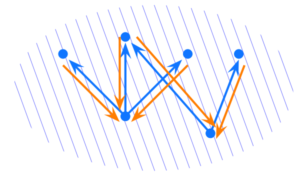
OWL class relationships vs DL inclusions

OWL axiom	DL	Example
Class(A partial $C_1 \ldots C_n$))	$A \sqsubseteq C_1 \sqcap \ldots C_n$	**Human** $\sqsubseteq$ **Physical_Object**
Class(A complete $C_1 \ldots C_n$))	$A \equiv C_1 \sqcap \ldots C_n$	**Man** $\equiv$ **Human** $\sqcap$ **Male**
SubClassOf(C_1 C_2)	$C_1 \sqsubseteq C_2$	**Human** $\sqsubseteq$ **Animal** $\sqcap$ **Biped**
EquivalentClasses(C_1 C_2)	$C_1 \equiv C_2$	**Man** $\equiv$ **Human** $\sqcap$ **Male**
DisjointClasses(C_1 C_2)	$C_1 \sqsubseteq \neg C_2$	**Male** $\sqsubseteq$ $\neg$**Female**
SameIndividual(a_1 a_2)	$\{a_1\} \equiv \{a_2\}$	PresidentBush=G.W.Bush
DifferentIndividual(a_1 a_2)	$\{a_1\} \sqsubseteq \neg\{a_2\}$	Bush≠Obama

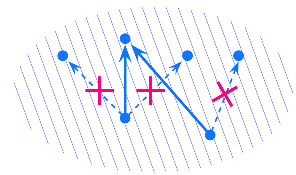
Object Properties vs Role Inclusions

SubPropertyOf$(R\ \ S)$	$R \sqsubseteq S$
EquivalentProperty$(R\ \ S)$	$R \equiv S$
ObjectProperty$(R\ \ \ldots)$	
super(S)	$R \sqsubseteq S$
inverseOf(S)	$R \equiv S^-$
Transitive	*transitive*(R)
Functional	$\top \ \sqsubseteq \ (\leq 1\ R\ \top)$
InverseFunctional	$\top \ \sqsubseteq \ (\leq 1\ R^-\ \top)$
Symmetric	$R^- \sqsubseteq R$
range(C)	$\top \ \sqsubseteq \forall R.C$
domain(C)	$\exists R.\top \ \sqsubseteq \ C$

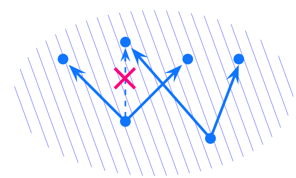
property R and its inverse R^-



R is functional



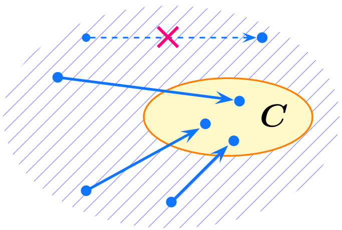
R is inverse functional



OWL vs DL: Domain and Range Constraints

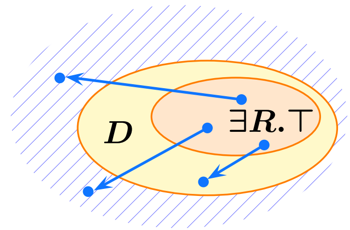
ObjectProperty$(R$ **range**$(C))$

$$\top \sqsubseteq \forall \boldsymbol{R}.\boldsymbol{C}$$



ObjectProperty$(R$ **domain**$(D))$

$$\exists \boldsymbol{R}.\top \sqsubseteq \boldsymbol{D}$$



NB: another way to represent the domain constraint:

$$\top \sqsubseteq \forall \boldsymbol{R}^{-}.\boldsymbol{D}$$

From DL to OWL

Note that DL inclusions can be represented in OWL in various ways. For instance,

$$A \sqsubseteq B \qquad (A \text{ and } B \text{ are concept names})$$

corresponds to

1. **Class**(A **partial** B)
2. **SubClassOf**(A B)
3. **DisjointClasses**(A **complementOf**(B))

$$\top \sqsubseteq \leq 1\, R \qquad (R \text{ is a role name})$$

corresponds to

1. **ObjectProperty**(R **functional**)
2. **SubClassOf(owl:Thing restriction**(R **maxCardinality**(1)))
3. **DisjointClasses(owl:Thing**
   complementOf(restriction(R **maxCardinality**(1))))