OWL

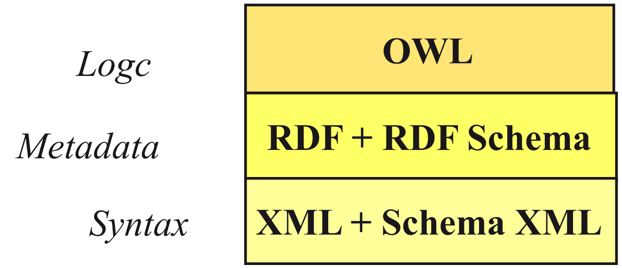
Amel Bouzeghoub
CSC4538

Introduction

- is based on research carried out in the field of description logic
- used to describe ontologies, i.e. it enables terminologies to be defined to describe concrete domains
- is an important step forward in the representation and organisation of knowledge available on the Web
- is designed as an extension of the Resource Description Framework (RDF) and RDF Schema (RDFS).

Introduction

- **RDF and RDFs alone are too limited :**
  - Cannot specify the nature of the relationships between resources (reflexivity, etc.)
  - No capacity for reasoning
  - Very limited logic
- **The need for OWL :**
  - Derives from RDF + RDFS
  - Logical connectors between classes (union, intersection, etc.)
  - Cardinality on properties
  - Characterization of properties (transitivity, inverse, etc.)

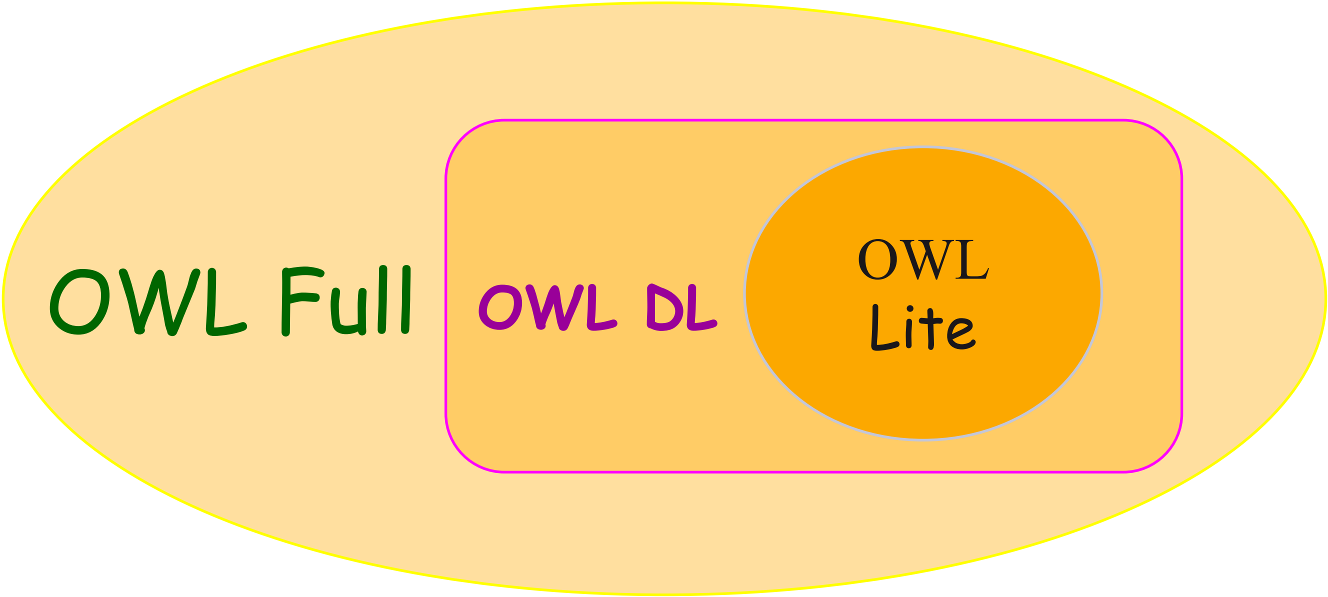


Main advantages

- Brings better integration, evolution, sharing and easier inference of ontologies
- Adds the concepts of equivalent classes, equivalent properties, equality of two resources, their differences, the opposite, symmetry and cardinality
- Thanks to its formal semantics based on a widely studied logical foundation, allows to define more complex associations of resources as well as the properties of their respective classes
- Is suitable for the Semantic Web, as it offers a strictly defined syntax, and depending on the level can allow automated reasoning on knowledge inferences and conclusions

OWL sub-languages

- OWL has three expressive languages for use by different communities of developers and users.



Structure of an owl ontology

- **Based on RDFS**
- **An OWL ontology is an *OWL document* (file extension .rdf or .owl) with:**
  - Namespace declarations (owl, rdf, and others)
  - The header (<owl:Ontology>) to describe the content of the ontology
  - The definition of classes
  - The definition of properties
  - Assertion of facts
- **Extensibility of existing ontologies :**
  - <owl:import> to use other OWL ontolgies and extend them :

Definition of OWL classes

- **A class can be declared in several ways:**
  - By naming the class or,
  - By enumeration of its individuals
  - By restricting the properties of its individuals
  - By intersection (AND), union (OR) or complement (NOT) of another class

  ==> Anonymous Classes: The members of an anonymous class are the set of Individuals that satisfy its logical definition
- **There is an inheritance mechanism** (<owl:subClassOf>)
- **The superclass** owl:Thing **is the mother of all the other classes**
- owl:Nothing **is subclass of all classes**
- **In OWL Full, a class can be an instance of another class (a "**metaclass**").**
- **The set of instances of a class is called "the extension".**

Restriction Types

$\exists$	Existential, someValuesFrom	"Some", "At least one"
$\forall$	Universal, allValuesFrom	"Only"
$\ni$	hasValue	"equals x"
$=$	Cardinality	"Exactly n"
$\leq$	Max Cardinality	"At most n"
$\geq$	Min Cardinality	"At least n"

Property Characteristics

- **Domain and range can be set**
- **OWL offers a mechanism for property inheritance:**

```
owl:ObjectProperty rdf:ID="aPourFrere">
<rdfs:subPropertyOf rdf:resource="#estDeLaFamilleDe" />
<rdfs:range rdf:resource="#Humain" />
<rdfs:domain rdf:resource="#Humain" />
</owl:ObjectProperty>
```

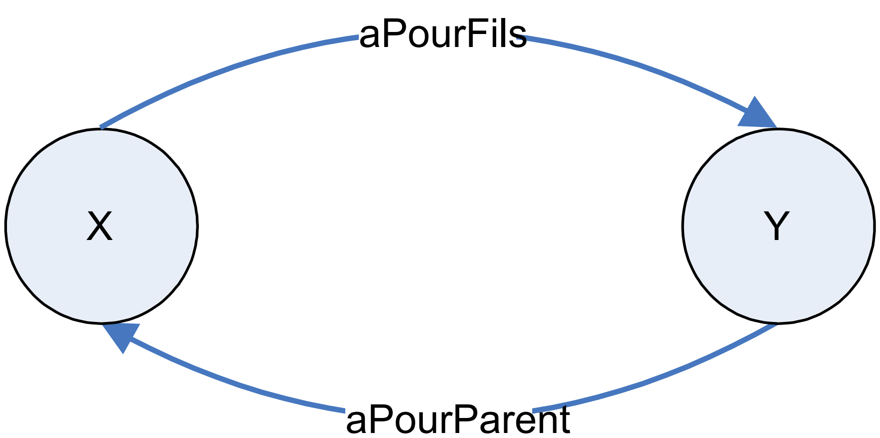
- **Properties can be characterized:**
  - Inverse
  - Transitivité
  - Symétrie
  - Fonctionnelle
  - fonctionnelle Inverse

```
<<owl:ObjectProperty rdf:ID="aPourFrere">
<rdf:type rdf:resource="&owl;SymmetricProperty" />
<rdfs:range rdf:resource="#Humain" />
<rdfs:domain rdf:resource="#Humain" />
</owl:ObjectProperty>
```

TELECOM SudParis

Inverse Property

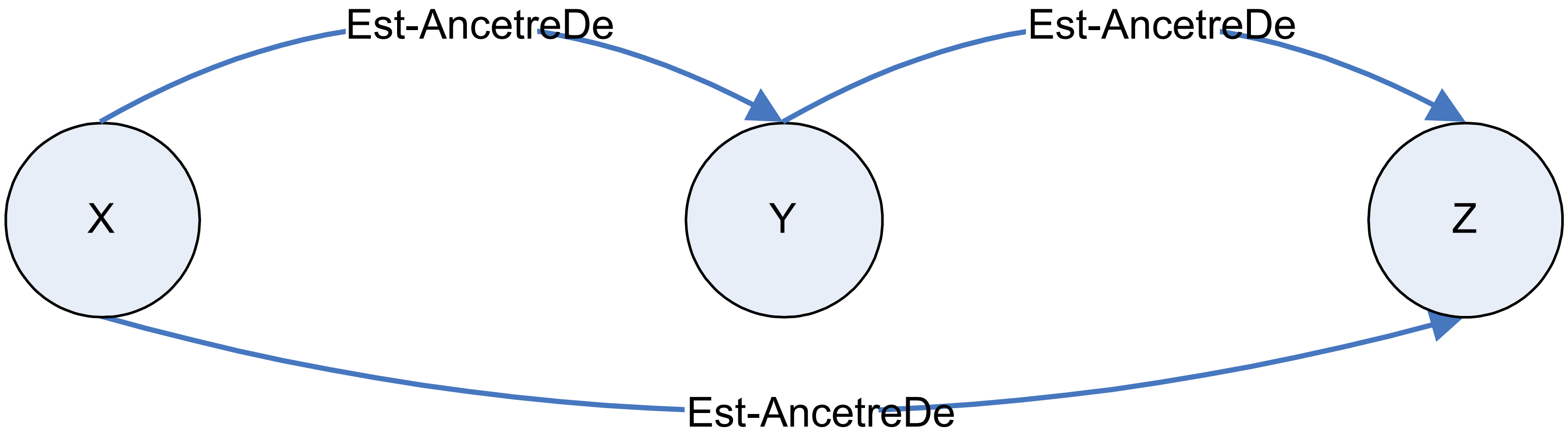
- P1(X ,Y) iff P2(Y , X)



- X mange Y iff Y estMangéPar X
- X aPourParent Y iff Y aPourFils X

Transitive Property

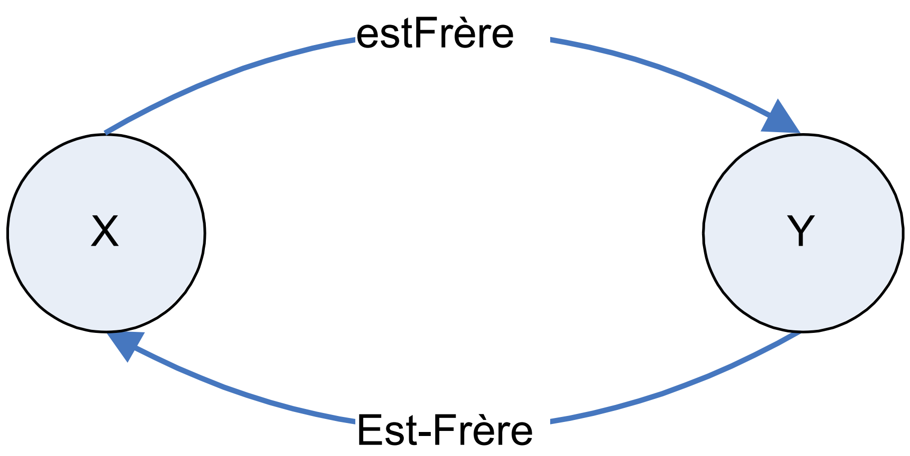
- If P(X,Y) and P(Y,Z) then P(X,Z)



- X ancetreDe Y, Y ancetreDe Z, then X ancetreDe Z

Symmetric Property

- P(X,Y) iff P(Y,X)

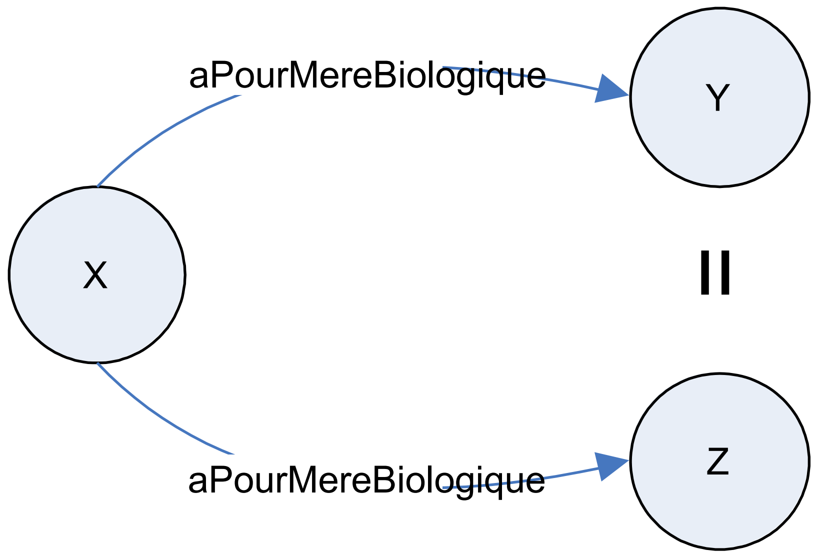


- X estFrère Y iff Y estFrère X

Functional Property

- **Unicity**
  - Only one instance can be linked

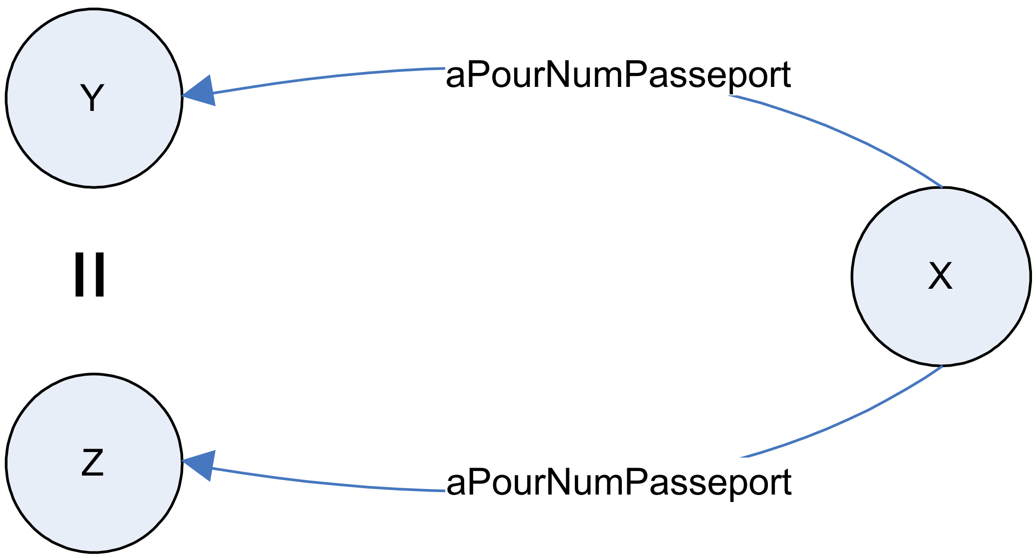
If P(X,Y) and P(X,Z) then Y=Z



- If X aPourMereBiologique Y and X aPourMereBiologique Z then Y=Z

Inverse Functional Property

- P(Y,X) and P(Z,X) then Y=Z



- Y aPourNumPasseport X and Z aPourNumPasseport X then Y=Z

OWL Class Constructors

Constructor	DL Syntax	Example	FOL Syntax
intersectionOf	$C_1 \sqcap \ldots \sqcap C_n$	Human $\sqcap$ Male	$C_1(x) \wedge \ldots \wedge C_n(x)$
unionOf	$C_1 \sqcup \ldots \sqcup C_n$	Doctor $\sqcup$ Lawyer	$C_1(x) \vee \ldots \vee C_n(x)$
complementOf	$\neg C$	$\neg$Male	$\neg C(x)$
oneOf	$\{x_1\} \sqcup \ldots \sqcup \{x_n\}$	{john} $\sqcup$ {mary}	$x = x_1 \vee \ldots \vee x = x_n$
allValuesFrom	$\forall P.C$	$\forall$hasChild.Doctor	$\forall y.P(x,y) \rightarrow C(y)$
someValuesFrom	$\exists P.C$	$\exists$hasChild.Lawyer	$\exists y.P(x,y) \wedge C(y)$
maxCardinality	$\leqslant nP$	$\leqslant$1hasChild	$\exists^{\leqslant n} y.P(x,y)$
minCardinality	$\geqslant nP$	$\geqslant$2hasChild	$\exists^{\geqslant n} y.P(x,y)$

OWL Axioms

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human $\sqsubseteq$ Animal $\sqcap$ Biped
equivalentClass	$C_1 \equiv C_2$	Man $\equiv$ Human $\sqcap$ Male
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq$ $\neg$Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} $\equiv$ {G_W_Bush}
differentFrom	$\{x_1\} \sqsubseteq \neg\{x_2\}$	{john} $\sqsubseteq$ $\neg${peter}
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter $\sqsubseteq$ hasChild
equivalentProperty	$P_1 \equiv P_2$	cost $\equiv$ price
inverseOf	$P_1 \equiv P_2^-$	hasChild $\equiv$ hasParent$^-$
transitiveProperty	$P^+ \sqsubseteq P$	ancestor$^+$ $\sqsubseteq$ ancestor
functionalProperty	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant$1hasMother
inverseFunctionalProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant$1hasSSN$^-$