Web sémantique

OWL

olivier.corby@inria.fr

Plan

1. RDF : Resource Description Framework
2. RDFS : RDF Schema
3. SPARQL : RDF Query Language
4. OWL : Web Ontology Language
5. Examen

2

Web Ontology Language (OWL)

- recommandation W3C
- logique de description
- primitives supplémentaires pour des ontologies plus complexes.
- définitions plus riches des classes et propriétés.
- permet de tirer plus de conclusions, de faire plus d'inférences.

3

OWL

Classes définies par des expressions logiques

Les parents sont des personnes qui ont au moins un enfant

Class Parent

subClassOf Person ;

subClassOf [property child ; minCardinality 1] .

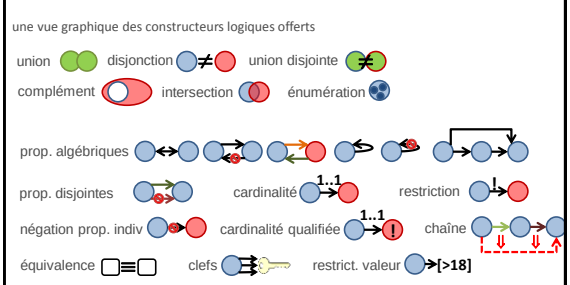
4

Syntaxe OWL

1. syntaxe RDF (Turtle ou RDF/XML)
2. syntaxe fonctionnelle

5

OWL in one…



6

espace de noms et préfixe

http://www.w3.org/2002/07/owl#

- espace des primitives OWL
- même principe que pour RDFS
- préfixe `owl:` dans la suite

7

OWL

1. **Relations de classes**
2. Caractérisation des propriétés
3. Equivalences et alignements
4. Restriction de propriétés
5. Gérer les schémas
6. Profiles OWL

8

classes énumérées

définir une classe en énumérant tous ses membres

```
@prefix ex: <http://example.org/schema/> .

ex:EyeColor a owl:Class ;
  owl:oneOf
    (ex:Blue ex:Green ex:Brown ex:Black).
```

9

union de classes

toute ressource des classes unies est aussi dans la classe union

```
ex:LegalAgent a owl:Class ;
  owl:equivalentClass [
    owl:unionOf ( ex:Person ex:Group )
  ] .
```

10

classes définies par intersection

toute ressource commune aux classes est aussi dans la classe intersection

```
ex:Man a owl:Class ;
 owl:equivalentClass [
   owl:intersectionOf (ex:Person ex:Male)
].
```

11

classes définies par négation

les ressources qui ne font pas partie d'une classe

```
ex:Inedible a owl:Class ;
  owl:complementOf ex:Edible .
```

12

disjonction de deux classes

une ressource ne peut pas appartenir aux deux classes en même temps

□≠○

```
ex:Square a owl:Class ;
  owl:disjointWith ex:Circle .
```

13

disjonction de plusieurs classes

une ressource ne peut, au plus, appartenir qu'à une seule de ces classes

□≠○
△≠○
□≠△

```
[] a owl:AllDisjointClasses ;
   owl:members
    ( ex:Square ex:Circle ex:Triangle ) .
```

14

union disjointe

diviser une classe en une partition complète de sous classes

```
ex:Passenger a owl:Class ;
  owl:disjointUnionOf
  (ex:Adult ex:Child ex:Pet) .
```

15

OWL

1. Relations de classes
2. **Caractérisation des propriétés**
3. Equivalences et alignements
4. Restriction de propriétés
5. Gérer les schémas
6. Profiles OWL

16

trois types de propriétés

1. `owl:ObjectProperty` sont des relations entre des ressources
2. `owl:DatatypeProperty` ont des valeurs littérales (typées)
3. `owl:AnnotationProperty` sont ignorées dans les inférences et utilisées pour documenter ou pour des extensions

17

propriétés symétriques

une relation qui, dès lors qu'elle existe, existe dans les deux sens (ex. marié_avec)

x R y ⇒ y R x

```
ex:hasSpouse a owl:SymmetricProperty .
```

18

propriétés asymétriques

une relation qui, dès lors qu'elle existe, n'existe que dans un seul sens.

$x\ R\ y \Rightarrow \neg\ y\ R\ x$

```
ex:hasChild a owl:AsymmetricProperty .
```

19

propriétés inverses

deux relations qui existent simultanément en sens inverse (ex. parent_de / enfant_de)

$x\ R_1\ y \Leftrightarrow y\ R_2\ x$

```
ex:hasChild owl:inverseOf ex:hasParent .
```

20

propriétés transitives

une relation qui se propage de proche en proche (ex. Tom ancêtre Jim ancêtre Jules)

$x\ R\ y\ \&\ y\ R\ z \Rightarrow x\ R\ z$

```
ex:hasAncestor a owl:TransitiveProperty .
```

21

propriétés disjointes

Des relations qui ne peuvent pas exister en même temps sur le même sujet et le même objet

```
ex:hasSon owl:propertyDisjointWith
ex:hasDaughter .
```

22

propriétés réflexives

Une relation qui relie les individus à eux mêmes

```
ex:hasRelative a owl:ReflexiveProperty .
```

23

propriétés chaînées

des relations qui mises bout à bout impliquent une autre relation
(ex. parent o frère = oncle)

$x\ P\ y\ \&\ y\ Q\ z \Rightarrow x\ R\ z$

```
ex:uncle owl:propertyChainAxiom
     (ex:parent ex:brother) .
```

24

propriétés fonctionnelles

une relation pour laquelle une ressource ne peut avoir qu'une valeur (ex. naissance)

$x\ R\ y\ \&\ x\ R\ z \Rightarrow y = z$

```
ex:birthDate a owl:FunctionalProperty .
```

25

propriétés inverses fonctionnelles

une relation pour laquelle une même valeur implique la même ressource (ex. NSS)

$x\ R\ y\ \&\ z\ R\ y \Rightarrow x = z$

```
ex:socialSecurityNumber a
    owl:InverseFunctionalProperty .
```

26

identification par des clés

deux instances qui ont même(s) valeur(s) de clé(s) sont en réalité la même instance

$x\ c_1\ v_1\ ;\ c_2\ v_2\ \&\ y\ c_1\ v_1\ ;\ c_2\ v_2 \Rightarrow x = y$

```
ex:Person owl:hasKey ( ex:hasSSN ) .
```

27

OWL

1. Relations de classes
2. Caractérisation des propriétés
3. **Equivalences et alignements**
4. Restriction de propriétés
5. Gérer les schémas
6. Profiles OWL

28

équivalences de classes

deux classes rassemblent exactement les mêmes ressources.

```
ex:Human owl:equivalentClass foaf:Person .
```

29

équivalences de propriétés

deux propriétés expriment exactement la même relation.

```
ex:name owl:equivalentProperty my:label .
```

30

ressources identiques

deux URI identifient exactement la même chose.

```
ex:Bill owl:sameAs ex:William .
```

31

transitivité: l'identité se propage

- URI_1 owl:sameAs URI_2
- URI_2 owl:sameAs URI_3

…

- URI_1 owl:sameAs URI_3

32

ressources différentes

deux URI dont on sait qu'ils identifient des choses différentes.

```
ex:Good owl:differentFrom ex:Evil .
```

33

OWL

1. Relations de classes
2. Caractérisation des propriétés
3. Equivalences et alignements
4. **Restriction de propriétés**
5. Gérer les schémas
6. Profiles OWL

34

restriction de toutes les valeurs de propriété

classe pour laquelle toutes les valeurs d'une propriété sont issues d'une classe spécifique

```
ex:Herbivore a owl:Class ;
  rdfs:subClassOf ex:Animal ;
  rdfs:subClassOf
    [ a owl:Restriction ;
     owl:onProperty ex:eats ;
     owl:allValuesFrom ex:Plant ] .
```

35

restriction de certaines valeurs de propriété

au moins certaines valeurs de la propriété sont issues d'une classe spécifique

```
ex:Sportive a owl:Class ;
 owl:equivalentClass
  [ a owl:Restriction ;
   owl:onProperty ex:hobby ;
   owl:someValuesFrom ex:Sport ] .
```

36

restriction à une seule valeur de propriété

la classe définie ne peut avoir qu'une seule valeur pour la propriété visée

```
ex:Bicycle a owl:Class ;
 rdfs:subClassOf
  [ a owl:Restriction ;
   owl:onProperty ex:nbWheels ;
   owl:hasValue 2 ] .
```

37

restriction de relation avec soi-même

classe définie par les instances ayant elles-mêmes comme valeur d'une propriété

```
ex:NarcisticPerson a owl:Class ;
rdfs:subClassOf
[ a owl:Restriction ;
    owl:onProperty ex:love ;
    owl:hasSelf true ]
```

38

restriction de cardinalité

contrainte sur le nombre de fois qu'une propriété peut être utilisée avec des valeurs différentes sur le même sujet : minimum, maximum, nombre exact

```
ex:Person a owl:Class ;
 rdfs:subClassOf
  [ a owl:Restriction ;
   owl:onProperty ex:name ;
   owl:maxCardinality 1 ] .
```

39

restriction de cardinalité qualifiée

contrainte sur le nombre de fois qu'une propriété peut être utilisée avec des valeurs d'un certain type sur le même sujet : minimum, maximum, nombre exact

```
ex:Human a owl:Class ;
 rdfs:subClassOf
  [ a owl:Restriction ;
   owl:onProperty ex:hasParent ;
   owl:onClass ex:Male ;
   owl:qualifiedCardinality 1 ] .
```

40

OWL

1. Relations de classes
2. Caractérisation des propriétés
3. Equivalences et alignements
4. Restriction de propriétés
5. **Gérer les schémas**
6. Profiles OWL

41

documenter les schémas et leurs versions

- L'ontologie est aussi une ressource
- L'ontologie a un URI qui nous permet d'en parler
- OWL fournit des primitives pour décrire cette ressource qu'est l'ontologie

42

description de l'ontologie

une classe (owl:Ontology) et plusieurs propriétés (owl:imports, owl:versionInfo, owl:priorVersion, owl:backwardCompatibleWith, owl:incompatibleWith)

```
<http://inria.fr/2005/humans/> a owl:Ontology ;
rdfs:comment "An example OWL ontology" ;
owl:priorVersion <http://inria.fr/2004/humans/> ;
owl:imports <http://cnrs.fr/animals/> ;
rdfs:label "Bio Ontology" .
```

43

changements de classes ou de propriétés

marquer une classe ou une propriété comme obsolète

```
ex:Mammal a owl:DeprecatedClass .

ex:age a owl:DeprecatedProperty .
```

44