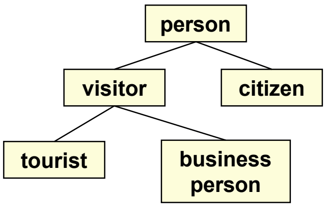
What is an Ontology?

In Greek philosophy:
Ontology is the study of being and existence, of the nature of reality

In Computer Science:
An ontology is a formal explicit specification of a shared conceptualization of a domain.

A specification consisting of
- classes
- relations between classes
- individuals
- axioms.

"Ontology" is often misused for a taxonomy of concept names without specification of formal relations between classes.



1

Important Aspects of Ontologies (1)

An ontology is a formal explicit specification of a shared conceptualization of a domain.

Ontologies are formal:
- Meaning of ontology is unambiguous, avoids misunderstanding
- Specification using formal language enables reasoning
- Specification may hamper consensus

Ontologies use explicit specifications:
- To make domain assumptions explicit for reasoning
- To support understanding of domain.

Note principle of minimal ontological commitment:
“Make as little as explicit as possible, while keeping ontology useful”
- Too much explicit => no consensus
- Too little explicit => ontology unusable

2

Important Aspects of Ontologies (2)

An ontology is a formal explicit specification of a shared conceptualization of a domain.

Ontologies provide a shared conceptualization of a domain:

- **Ontologies must be restricted to a specific application area (domain) to be manageable**
- **Shared conceptualizations enable knowledge pooling**
- **Sharing means consensus - may not be easy to obtain**

Example for Ontology-Based Inferences

- **Consistency check of E-business internet catalogue:**

SPECIAL OFFER:

MMZ100, year 2000, à EUR 75,-

Elsewhere in the internet:

MMZ100 is a Multimedia Center
MMZ100 has a list price of DM 150,-
All entertainment systems built before 2002 are sold with 20% rebate on the list price
A Multimedia Center is a special TV set
A TV set is an entertainment system
1 EUR = 1,95583 DM

information is inconsistent !

Web Ontology Language OWL

- W3C Recommendation since 2004
- OWL semantics constitute fragment of FOL
- Three variants:
  OWL Lite $\subseteq$ OWL DL $\subseteq$ OWL Full
- No Reification in OWL DL (DL = Description Logics)
  => RDFS is fragment of OWL Full
- OWL DL is decidable
- OWL Full is not decidable
  => correct and complete algorithms for consistency check and other inferences do not exist
- An OWL document is often called an "ontology"

Basic Semantics of OWL

- **Defines the basic concepts (resources) of a domain in terms of classes:**
  - classes can be viewed as "sets" of possible individuals
  - hierarchies of concepts can be defined as subclasses
- **Properties are defined by:**
  - constraints on their range and domain, or
  - specialization (sub-properties)
- **Structure is based on RDF**
- **Expressiveness and inferences equivalent to expressive Description Logics**

The Three OWL Variants

- OWL Full
  - contains OWL DL and OWL Lite
  - only OWL variant which completely contains RDFS
  - semantics problematic from a logical point of view
  - undecidable
  - only partial support by existing software tools
- OWL DL
  - contains OWL Lite and is contained in OWL Full
  - decidable.
  - almost complete support by existing software tools
  - complexity NExpTime (worst-case)
- OWL Lite
  - contained in OWL DL and OWL Full
  - decidable
  - not very expressive
  - complexity ExpTime (worst-case).

7

OWL Documents (1)

OWL documents
- are RDF documents
- consist of
  - head with general information
  - main body with ontology

Namespaces are defined in the RDF root:

```
<rdf:RDF
    xmlns ="http://www.semanticweb-grundlagen.de/beispielontologie#"
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd ="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs ="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl ="http://www.w3.org/2002/07/owl#">
...
</rdf:RDF>
```

8

OWL Documents (2)

General information is defined in an OWL:Ontology element:

```
<owl:Ontology rdf:about="">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string“>
    SWRC Ontologie in der Version vom Dezember 2005
    </rdfs:comment>
    <owl:versionInfo>v0.5</owl:versionInfo>
    <owl:imports rdf:resource="http://www.semanticwebgrundlagen.de/foo"/>
    <owl:priorVersion rdf:resource="http://ontoware.org/projects/swrc"/>
</owl:Ontology>
```

Inherited from RDFS:

rdfs:comment
rdfs:label
rdfs:seeAlso
rdfs:isDefinedBy

Versioning tags:

owl:versionInfo
owl:priorVersion
owl:backwardCompatibleWith
owl:incompatibleWith
owl:DeprecatedClass
owl:DeprecatedProperty

9

Simple Class Definitions

Classes can be defined using the RDFS notation:

```
<rdf:Description rdf:about="Professor">
    <rdf:type rdf:resource="&owl;Class"/>
</rdf:Description>
```

... or in short:

```
<owl:Class rdf:about="Professor"/>
```

Predefined classes:

owl:Thing	**class containing all individuals**
owl:Nothing	**class containing no individual**

Compare with description logics notation: $\top$ $\bot$

10

Individuals

Individuals can be assigned to classes using the RDFS notation:

```
<rdf:Description rdf:ID="BerndNeumann">
    <rdf:type rdf:resource="#Professor"/>
</rdf:Description>
```

... or in short:

```
<Professor rdf:ID="BerndNeumann"/>
```

11

Properties or Roles

Property (RDF term) = role (OWL term)

- **<u>Abstract roles</u> connect individuals with individuals**
- **<u>Concrete roles</u> connect individuals with typed data**

Abstract roles are defined similar as classes:

```
<owl:ObjectProperty rdf:ID="BelongingTo"/>
```

Domain and range of abstract roles:

```
<owl:ObjectProperty rdf:ID="BelongingTo">
    <rdfs:domain rdf:resource="#Person"/>
    <rdfs:range rdf:resource="#Organisation"/>
</owl:ObjectProperty>
```

Domain and range of concrete roles:

```
<owl:DatatypeProperty rdf:ID="FirstName">
    <rdfs:domain rdf:resource="#Person" />
    <rdfs:range rdf:resource="&xsd;string" />
</ owl:DataTypeProperty>
```

← **"FirstName" is a role with range string (defined datatype)**

12

Relationships between Classes and Individuals

Relationships between classes:

rdfs:subClassOf	**generates class hierarchies**
owl:disjointWith	**disjoint classes do not have common members**
owl:equivalentClass	**equivalent classes have the same members**

Relationships between individuals:

owl:sameAs **identical idividuals**

owl:differentFrom **distinct individuals**

For abbreviated notation:

owl:AllDifferent **and** owl:distinctMembers

Closed classes:

owl:oneOf **a class may be defined in terms of a collection of individuals**

13

Constructing Complex Classes

Logical constructors:

owl:intersectionOf	**logical AND**
owl:unionOf	**logical OR**
owl:complementOf	**logical negation**

Role restrictions using

owl:allValuesFrom	**all**
owl:someValuesFrom	**at least one**
owl:maxCardinality	**at most**
owl:minCardinality	**at least**
owl:cardinality	**exactly**
owl:hasValue	**refers to concrete individual**

Compare with cardinality restrictions in description logics!

14

Example for Role Restriction

```
<owl:Class rdf:ID="Examen">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#hasExaminer"/>
            <owl:allValuesFrom rdf:resource="#Professor"/>
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

15

Properties and Relations of Roles

In OWL several <u>role properties</u> can be specified:

- **domain**
- **range**
- **transitivity** — **r(a, b) and r(b, c) implies r(a, c)**
- **symmetry** — **r(a, b) implies r(b, a)**
- **functionality** — **r(a, b) and r(a, c) implies b = c**
- **inverse functionality** — **r(a, b) and r(c, b) implies a = c**

<u>Role relations</u>:

- **subPropertyOf** — **r $\subseteq$ r´and r(a, b) implies r´(a, b)**
- **equivalentProperty**
- **inverseOf** — **r inverse of r´ and r(a, b) implies r´(b, a)**

```
<owl:ObjectProperty rdf:ID="hasExaminer">
    <owl:inverseOf rdf:resource="#isExaminedBy"/>
</owl:ObjectProperty>
```

16

OWL Inference Services

- **No specific recommendations of W3C regarding inference services**
- **General agreement about important services:**

Determine

- **class equivalence**
- **subclass relationship**
- **disjunctiveness**
- **global consistency (satisfiability)**
- **class consistency**

A class is inconsistent, if it is equivalent to owl:Nothing

Example for obviously inconsistent class:

```
<owl:Class rdf:about="#book">
    <owl:subClassOf rdf:resource="#publication"/>
    <owl:disjointWith rdf:resource="#publication"/>
</owl:Class>
```

17

OWL Tools

Editors

- **Protegé, http://protege.stanford.edu**
  Best-known ontology editor
- **SWOOP, http://www.mindswap.org/2004/SWOOP/**
- **OWL Tools, http://owltools.ontoware.org/**

Inference engines

- **Pellet, http://www.mindswap.org/2003/pellet/index.shtml**
  Open Source, complete support of OWL DL
- **KAON2, http://kaon2.semanticweb.org**
- **FACT++, http://owl.man.ac.uk/factplusplus/**
- **RacerPro, http://www.racer-systems.com/**
  Originally developed at KOGS, Department of Informatics, Hamburg University

18

Ontology Definitions with OWL (1)

Class definitions

```
<owl:Class rdf:ID="Animal">
       <rdfs:label>Animal</rdfs:label>
       <rdfs:comment>
              This class of animals is illustrative of a number of
              ontological idioms.
       </rdfs:comment>
</<owl:Class>
```

Subclasses

```
<owl:Class rdf:ID="Male">
             <rdfs:subClassOf rdf:resource="#Animal"/>
</owl:Class>

<owl:Class rdf:ID="Female">
             <rdfs:subClassOf rdf:resource="#Animal"/>
             <owl:disjointWith rdf:resource="#Male"/>
</owl:Class>
```

Multiple parent classes

```
<owl:Class rdf:ID="Man">
    <rdfs:subClassOf rdf:resource="#Person"/>
    <rdfs:subClassOf rdf:resource="#Male"/>
</owl:Class>
```

Ontology Definitions with OWL (2)

Value restrictions on property ranges

```
<<owl:Class rdf:ID="Person">
    <rdfs:subClassOf rdf:resource="#Animal"/>
    <rdfs:subClassOf>
        <owl:Restriction>
                <<owl:onProperty rdf:resource="#hasParent"/>
                <<owl:toClass rdf:resource="#Person"/>
        </<owl:Restriction>
    </rdfs:subClassOf>
```

Number restrictions on property ranges

```
<rdfs:subClassOf>
        <<owl:Restriction <owl:cardinality="1">
                <<owl:onProperty rdf:resource="#hasFather"/>
        </<owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
        <<owl:Restriction>
                <<owl:onProperty rdf:resource="#shoesize"/>
                <<owl:minCardinality>1</<owl:minCardinality>
        </<owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Ontology Definitions with OWL (3)

Object property definitions

```
<owl:ObjectProperty rdf:ID="hasParent">
  <rdfs:domain rdf:resource="#Animal"/>
  <rdfs:range rdf:resource="#Animal"/>
</owl:ObjectProperty>
```

Datatyp property definitions

Use of URIs

```
<owl:DatatypeProperty rdf:ID="age">
  <rdfs:comment>
    age is a DatatypeProperty whose range is xsd:decimal.
    age is also a UniqueProperty (can only have one age)
  </rdfs:comment>
  <rdf:type rdf:resource=
    "http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource=
"http://www.w3.org/2000/10/XMLSchema#nonNegativeInteger"/>
</owl:DatatypeProperty>
```

Ontology Definitions with OWL (4)

Subclass restrictions

```
<owl:Class rdf:about="#Person">
  <rdfs:subClassOf>
    <owl:Restriction owl:maxCardinalityQ="1">
        <owl:onProperty rdf:resource="#hasOccupation"/>
        <owl:hasClassQ rdf:resource="#FullTimeOccupation"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Unique properties

```
<owl:UniqueProperty rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
  <rdfs:range rdf:resource="#Female"/>
</owl:UniqueProperty>
```

Inverse properties

```
<owl:ObjectProperty rdf:ID="hasChild">
 <owl:inverseOf rdf:resource="#hasParent"/>
</owl:ObjectProperty>
```

Summary Ontology Language OWL

- Ontology language definitions and tools are standardized by W3C in an ongoing development
- Long-term goal is to make Web information machine understandable
- So far, RDF and OWL are mainly used for semantic annotation and exploitation of data in a well-defined application context
- OWL offers 3 sublanguages of varying expressivity, yet not all interesting queries can be formulated

23