OWL
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Ontology in Philosophy

- notion exists only in singular (no “ontologies”)
- denotes the “study of being”
- can be found in philosophical writings of Aristotle (Socrates), Thomas Aquinas, Descartes, Kant, Hegel, Wittgenstein, Heidegger, Quine, …
- term first mentioned in 17th century
Ontology in Computer Science

Gruber (1993):

“An Ontology is a

  formal specification \Rightarrow \text{interpretable by machines}

  of a \text{shared} \Rightarrow \text{based on consensus}

  conceptualization \Rightarrow \text{describes relevant notions}

  of a \text{domain of interest}\Rightarrow \text{referring to a “topic”}
Ontologies in Practice
Some Requirements

- instantiation of classes by individuals
- conceptual hierarchies (taxonomies, “inheritance”): classes, concepts
- binary relations between individuals: properties, roles
- characteristics of relations (z.B. range, transitive)
- datatypes (e.g. numbers): concrete domains
- logical operators
- clear semantics
RDFS – Simple Ontologies

Classes

- ex:Employee
- ex:Student
- ex:Professor
- ex:Tutor
- ex:PhDStudent
- ex:BGlimm
- ex:AWindeck
- ex:Frank

Instanzierung

- subClass

Relations

- ex:Professor
  - rdfs:domain: ex:Employee
  - rdfs:subPropertyOf: ex:responsibeFor
  - rdfs:subPropertyOf: ex:advises

- ex:Student
  - rdfs:range: rdf:Literal

- ex:email

- ex:companies: rdfs:subPropertyOf: rdfs:subPropertyOf

- ex:supervises
  - rdfs:domain: ex:Employee
  - rdfs:range: ex:Employee
RDF Schema as Ontology Language?

- appropriate for simple ontologies
- advantage: automated inferencing relatively efficient
- but: not appropriate for more complex modeling
- resort to more expressive languages, like
  - OWL
  - RIF ...
agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
OWL – General Remarks

- W3C Recommendation since 2004
- semantic fragment of FOL
- three variants:
  - OWL Lite
  - OWL DL
  - OWL Full
- no reification in OWL DL
- RDFS is fragment of OWL Full
- OWL DL is decidable
  - corresponds to description logic SHOIN(D)
- W3C documents contain details that cannot all be covered here
OWL 1 Variants

- OWL Full
  - contains OWL DL and OWL Lite
  - contains all of RDFS (as the only OWL variant)
  - semantics contains some aspects that are problematic from a logical perspective
  - undecidable
  - limited support by tools
OWL 1 Variants

- **OWL Full**
  - contains OWL DL and OWL Lite
  - contains all of RDFS (as the only OWL variant)
  - semantics contains some aspects that are problematic from a logical perspective
  - undecidable
  - limited support by tools

- **OWL DL**
  - contains OWL Lite and is sublanguage of OWL Full
  - widely supported by tools
  - complexity NExpTime (worst-case)
OWL 1 Variants

- **OWL Full**
  - contains OWL DL and OWL Lite
  - contains all of RDFS (as the only OWL variant)
  - semantics contains some aspects that are problematic from a logical perspective
  - undecidable
  - limited support by tools

- **OWL DL**
  - contains OWL Lite and is sublanguage of OWL Full
  - widely supported by tools
  - complexity NExpTime (worst-case)

- **OWL Lite**
  - sublanguage of OWL DL and OWL Full
  - low expressivity
  - complexity ExpTime (worst-case)
OWL Documents

- are RDF documents (at least in the standard syntax; there are others)
- consist of
  - head with general information
  - rest with actual ontology
Head of an OWL Document

definition of name spaces in the root

<rdf:RDF
    xmlns="http://example.org/exampleontology#"
    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>
Head of an OWL Document

general information

<owl:Ontology rdf:about="">
  <rdfs:comment>
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      SWRC ontology, version of June 2007
    </rdfs:comment>
  <owl:versionInfo>v0.7.1</owl:versionInfo>
  <owl:imports rdf:resource="http://www.example.org/foo" />
  <owl:priorVersion>
    rdf:resource="http://ontoware.org/projects/swrc" />
</owl:Ontology>
Head of an OWL Document

taken from RDFS
rdfs:comment
rdfs:label
rdfs:seeAlso
rdfs:isDefinedBy

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

in addition
owl:imports
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Classes, Roles and Individuals

three building blocks of ontology axioms

- classes
  - comparable with classes in RDFS
- individuals
  - comparable with “proper” instances in RDFS
- roles
  - comparable with properties in RDFS
classes

definition

- `<owl:Class rdf:about =¨Professor¨/>
- equivalent to

```xml
<rdf:Description rdf:about="#Professor">
  <rdf:type
    rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
```

pre-defined

- owl:Thing
- owl:Nothing
Individuals

definition via class membership

<rdf:Description rdf:about="rudiStuder">
  <rdf:type rdf:resource="#Professor"/>
</rdf:Description>

equivalent:

<Professor rdf:about="rudiStuder"/>
Abstrakte Rollen

abstract roles are defined in a way similar to classes

<owl:ObjectProperty rdf:about="hasAffiliation" />

domain and range of abstract roles

<owl:ObjectProperty rdf:about="hasAffiliation">
  <rdfs:domain rdf:resource="Person" />
  <rdfs:range rdf:resource="Organization" />
</owl:ObjectProperty>
Concrete Roles

concrete roles have datatypes as range

```xml
<owl:DatatypeProperty rdf:about="firstName" />
```

domain and range of concrete roles

```xml
<owl:DatatypeProperty rdf:about="firstName">
  <rdfs:domain rdf:resource="Person" />
  <rdfs:range rdf:resource="&xsd;string" />
</owl:DatatypeProperty>
```

many XML datatypes can be used
Individuals and Roles

<Person rdf:about="rudiStuder">
  <hasAffiliation rdf:resource="aifb" />
  <hasAffiliation rdf:resource="fzi" />
  <firstName rdf:datatype="&xsd:string">Rudi</firstName>
</Person>

in general roles are not functional
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
  - Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Simple Class Relationships

<owl:Class rdf:about="Professor">
  <rdfs:subClassOf rdf:resource="FacultyMember" />
</owl:Class>

<owl:Class rdf:about="FacultyMember">
  <rdfs:subClassOf rdf:resource="Person" />
</owl:Class>

It follows by inference that Professor is a subclass of Person
Simple Class Relationships

<owl:Class rdf:about="Professor">
  <rdfs:subClassOf rdf:resource="FacultyMember" />
</owl:Class>
<owl:Class rdf:about="Book">
  <rdfs:subClassOf rdf:resource="Publication" />
</owl:Class>
<owl:Class rdf:about="FacultyMember">
  <owl:disjointWith rdf:resource="Publication" />
</owl:Class>

it follows by inference that Professor and Book are also disjoint classes
Simple Class Relationships

<owl:Class rdf:about="Man">
  <rdfs:subClassOf rdf:resource="Person" />
</owl:Class>
<owl:Class rdf:about="Person">
  <owl:equivalentClass rdf:resource="Human" />
</owl:Class>

it follows by inference that Man is a subclass of Human
it follows by inference that Foundations of Semantic Web Technologies is a Publication.
Relationships between Individuals

<Professor rdf:about="rudiStuder" />
 rdf:Description rdf:about="rudiStuder">
 owl:sameAs rdf:resource="professorStuder" />
</rdf:Description>

it follows by inference that rudiStuder is a Professor
distinctness of individuals expressed via owl:differentFrom.
Relationships between Individuals

<owl:AllDifferent>
<owl:distinctMembers rdf:parseType="Collection">
<Person rdf:about="rudiStuder" />  
<Person rdf:about="dennyVrandecic" />  
<Person rdf:about="peterHaase" />  
</owl:distinctMembers>
</owl:AllDifferent>

abbreviated notation instead of using several owl:differentFrom
usage of owl:AllDifferent and owl:distinctMembers exclusively for this purpose
Closed Classes

<owl:Class rdf:about="SecretariesOfStuder">
  <owl:oneOf rdf:parseType="Collection">
    <Person rdf:about="giselaSchillinger" />
    <Person rdf:about="anneEberhardt" />
  </owl:oneOf>
</owl:Class>

tells that there are only exactly these two SecretariesOfStuder
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Logical Class Constructors

- **logical and (conjunction):**
  owl:intersectionOf

- **logical or (disjunction):**
  owl:unionOf

- **logical not (negation):**
  owl:complementOf

- used to construct complex classes from simple classes
Conjunction

<owl:Class rdf:about="SecretariesOfStuder">
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class rdf:about="Secretaries" />
    <owl:Class rdf:about="MembersOfStudersGroup" />
  </owl:intersectionOf>
</owl:Class>

It follows by inference that all *SecretariesOfStuder* are also *Secretaries*
Disjunction

<owl:Class rdf:about="Professor">
  <rdfs:subClassOf>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="ActivelyTeaching" />
        <owl:Class rdf:about="Retired" />
      </owl:unionOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>
Negation

\[
<\text{owl:Class} \text{ rdf:about="FacultyMember">}
\quad <\text{rdfs:subClassOf}>
\quad \quad <\text{owl:Class}>
\quad \quad \quad <\text{owl:complementOf rdf:resource="Publication"} />
\quad \quad </\text{owl:Class}>
\quad </\text{rdfs:subClassOf}>
</\text{owl:Class}>
\]

semantically equivalent:

\[
<\text{owl:Class} \text{ rdf:about="FacultyMember">}
\quad <\text{owl:disjointWith rdf:resource="Publication"} />
</\text{owl:Class}>
\]
Role Restrictions (allValuesFrom)

used to define complex classes via roles

```xml
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer"/>
      <owl:allValuesFrom rdf:resource="Professor"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

i.e., all examiners of an exam have to be professors
Role Restrictions (someValuesFrom)

<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer" />
      <owl:someValuesFrom rdf:resource="Person" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

i.e., every exam must have at least one examiner
Role Restrictions (Cardinalities)

<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer"/>
      <owl:maxCardinality rdf:datatype="&xsd;nonNegativeInteger">
        2
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

an exam may have at most two examiners
Role Restrictions (Cardinalities)

```xml
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasTopic"/>
      <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">3</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

an exam must cover at least three topics
```
Role Restrictions (Cardinalities)

```
<owl:Class rdf:about="Exam">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasTopic"/>
      <owl:cardinality
        rdf:datatype="&xsd;nonNegativeInteger">3
    </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

an exam must cover exactly three topics
```
Role Restrictions (hasValue)

<owl:Class rdf:about="ExamStuder">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer" />
      <owl:hasValue rdf:resource="rudiStuder" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>

owl:hasValue always refers to one singular individual
the above is equivalent to the example on the next slide
Role Restrictions (hasValue)

```xml
<owl:Class rdf:about="ExamStuder">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasExaminer"/>
      <owl:someValuesFrom>
        <owl:oneOf rdf:parseType="Collection">
          <owl:Thing rdf:about="rudiStuder"/>
        </owl:oneOf>
        <owl:oneOf>
          <owl:someValuesFrom/>
        </owl:oneOf>
      </owl:someValuesFrom>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Role Relationships

<owl:ObjectProperty rdf:about="hasExaminer">
  <rdfs:subPropertyOf rdf:resource="hasParticipant" />
</owl:ObjectProperty>

**likewise:** owl:equivalentProperty
roles can be inverses of each other:

<owl:ObjectProperty rdf:about="hasExaminer">
  <owl:inverseOf rdf:resource="examinerOf"/>
</owl:ObjectProperty>
Role Characteristics

- domain
- range
- transitivity, i.e.
  \[ r(a, b) \text{ and } r(b, c) \text{ imply } r(a, c) \]
- symmetry, i.e.
  \[ r(a, b) \text{ implies } r(b, a) \]
- functionality
  \[ r(a, b) \text{ and } r(a, c) \text{ imply } b = c \]
- inverse functionality
  \[ r(a, b) \text{ and } r(c, b) \text{ imply } a = c \]
Domain and Range

<owl:ObjectProperty rdf:about="isMemberOf">
    <rdfs:range rdf:resource="Organization" />
</owl:ObjectProperty>

equivalent to:

<owl:Class rdf:about="&owl;Thing">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="isMemberOf" />
            <owl:allValuesFrom rdf:resource="Organization" />
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
Domain and Range: Caution!

<owl:ObjectProperty rdf:about="isMemberOf">
  <rdfs:range rdf:resource="Organization" />
</owl:ObjectProperty>
<number rdf:about="five">
  <isMemberOf rdf:resource="PrimeNumbers" />
</number>

it follows that PrimeNumbers are an Organization!
Role Characteristics

```xml
<owl:ObjectProperty rdf:about="hasColleague">
    <rdf:type rdf:resource="&owl;TransitiveProperty" />
    <rdf:type rdf:resource="&owl;SymmetricProperty" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="hasProjectLeader">
    <rdf:type rdf:resource="&owl;FunctionalProperty" />
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="isProjectLeaderFor">
    <rdf:type rdf:resource="&owl;InverseFunctionalProperty" />
</owl:ObjectProperty>
<Person rdf:about="peterHaase">
    <hasColleague rdf:resource="philippCimiano" />
    <hasColleague rdf:resource="steffenLamparter" />
    <isProjectLeaderFor rdf:resource="neOn" />
</Person>
<Project rdf:about="x-Media">
    <hasProjectLeader rdf:resource="philippCimiano" />
    <hasProjectLeader rdf:resource="cimianoPhilipp" />
</Project>
```
Consequences from the Example

- steffenLamparter hasColleague peterHaase
- steffenLamparter hasColleague philippCimiano
- philippCimiano owl:sameAs cimianoPhilipp
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
OWL 1 Variants

- **OWL Full**
  - contains OWL DL and OWL Lite
  - contains all of RDFS (as the only OWL variant)
  - semantics contains some aspects that are problematic from a logical perspective
  - undecidable
  - limited support by tools
OWL 1 Variants

- **OWL Full**
  - contains OWL DL and OWL Lite
  - contains all of RDFS (as the only OWL variant)
  - semantics contains some aspects that are problematic from a logical perspective
  - undecidable
  - limited support by tools

- **OWL DL**
  - contains OWL Lite and is sublanguage of OWL Full
  - widely supported by tools
  - complexity NExpTime (worst-case)
OWL 1 Variants

- **OWL Full**
  - contains OWL DL and OWL Lite
  - contains all of RDFS (as the only OWL variant)
  - semantics contains some aspects that are problematic from a logical perspective
  - undecidable
  - limited support by tools

- **OWL DL**
  - contains OWL Lite and is sublanguage of OWL Full
  - widely supported by tools
  - complexity NExpTime (worst-case)

- **OWL Lite**
  - sublanguage of OWL DL and OWL Full
  - low expressivity
  - complexity ExpTime (worst-case)
OWL Full

- unrestricted use of all OWL and RDFS language elements (has to be valid RDFS)
- difficult e.g.: non-existent type separation (classes, roles, individuals), thus:
  - owl:Thing becomes the same as rdfs:resource
  - owl:Class becomes the same as rdfs:Class
  - owl:DatatypeProperty becomes a subclass of owl:ObjectProperty
  - owl:ObjectProperty becomes the same as rdf:Property
Example for Confusion of Types in OWL Full

<owl:Class rdf:about="Book">
  <germanName rdf:datatype="&xsd;string">Buch</germanName>
  <frenchName rdf:datatype="&xsd;string">livre</frenchName>
</owl:Class>

inferences about such constructs are rarely needed in practice
OWL DL

- only usage of RDFS language elements that are explicitly allowed (like those in our examples)
  not allowed: `rdfs:Class`, `rdfs:Property`
- type separation: classes and roles have to be explicitly declared
- concrete roles must not be specified as transitive, symmetric, inverse or inverse functional
- number restrictions must not be used with transitive roles, their subroles, or inverses thereof
OWL Lite

- all restrictions of OWL DL
- moreover:
  - **not allowed**: `oneOf`, `unionOf`, `complementOf`, `hasValue`, `disjointWith`
  - number restrictions only allowed with 0 and 1
  - some constraints referring to anonymous (complex) classes, e.g., only in the subject of `rdfs:subClassOf`
Agenda

- Motivation
- OWL – General Remarks
- Classes, Roles and Individuals
- Class Relationships
- Complex Classes
- Role Characteristics
- OWL Variants
- OWL Ontologies: Reasoning Tasks
Terminological Queries to OWL Ontologies

- class equivalence
- subclass relationships
- disjointness of classes
- global consistency (aka satisfiability)
- class consistency: a class is inconsistent if it is equivalent to `owl:Nothing` – this hints to a modeling error:

```xml
<owl:Class rdf:about="Book">
  <owl:subClassOf rdf:resource="Publication"/>
  <owl:disjointWith rdf:resource="Publication"/>
</owl:Class>
```
Assertional Queries to OWL Ontologies

- instance checking: does a given individual belong to a given class?
- search for all individuals that are members of a given class
- are two given individuals linked by a role?
- search for all individual pairs that are linked by a certain role
- ... caution: only “provable” answers will be given!
OWL 1 Language Elements

**head**

- rdfs:comment
- rdfs:label
- rdfs:seeAlso
- rdfs:isDefinedBy
- owl:versionInfo
- owl:priorVersion
- owl:backwardCompatibleWith
- owl:incompatibleWith
- owl:DeprecatedClass
- owl:DeprecatedProperty
- owl:imports

**relationships between individuals**

- owl:sameAs
- owl:differentFrom
- owl:AllDifferent
- owl:distinctMembers

**pre-defined datatypes (OWL 1)**

- xsd:strong
- xsd:integer
<table>
<thead>
<tr>
<th>Class Constructors and Relationships</th>
<th>Role Restrictions</th>
</tr>
</thead>
<tbody>
<tr>
<td>owl:Class</td>
<td>owl:allValuesFrom</td>
</tr>
<tr>
<td>owl:Thing</td>
<td>owl:someValuesFrom</td>
</tr>
<tr>
<td>owl:Nothing</td>
<td>owl:hasValue</td>
</tr>
<tr>
<td>rdfs:subClassOf</td>
<td>owl:cardinality</td>
</tr>
<tr>
<td>owl:disjointWith</td>
<td>owl:minCardinality</td>
</tr>
<tr>
<td>owl:equivalentClass</td>
<td>owl:maxCardinality</td>
</tr>
<tr>
<td>owl:intersectionOf</td>
<td></td>
</tr>
<tr>
<td>owl:unionOf</td>
<td></td>
</tr>
<tr>
<td>owl:complementOf</td>
<td></td>
</tr>
</tbody>
</table>
OWL Language Elements

role constructors, relationships and characteristics

- owl:ObjectProperty
- owl:DatatypeProperty
- rdfs:subPropertyOf
- owl:equivalentProperty
- owl:inverseOf
- rdfs:domain
- rdfs:range
- owl:TransitiveProperty
- owl:SymmetricProperty
- owl:FunctionalProperty
- owl:InverseFunctionalProperty
Further Literature

- http://www.w3.org/2004/OWL/
  central W3C web page for OWL
- http://www.w3.org/TR/owl-features/
  overview over OWL
- http://www.w3.org/TR/owl-ref/
  comprehensive description of the OWL language components
- http://www.w3.org/TR/owl-guide/
  introduction into OWL knowledge modeling
- http://www.w3.org/TR/owl-semantics/
  describes the semantics of OWL and the abstract syntax for OWL DL (later lecture)