Ontology and OWL basics

Robert Hoehndorf

Applied Ontology

- builds on philosophy, cognitive science, linguistics and logic
- with the purpose of understanding, clarifying, making explicit and communicating *people's assumptions* about the nature and structure of the world
- orientation towards helping people understand each other distinguishes applied ontology from philosophical ontology, and motivates its unavoidable interdisciplinary nature
- Ontological analysis: study of content (of these assumptions) as such (independently of their representation)

Do we know what to Represent?

- First analysis, then representation
  - Unfortunately, that's not always the case
- Computer scientists have focused on the structure of representations and the nature of reasoning more than on the content of such representations
- Essential ontological promiscuity of AI: any agent creates its own ontology based on its usefulness for the task at hand

Ontologies in CS

- Specific (theoretical or computational) artifacts expressing the intended meaning of a vocabulary in terms of primitive categories and relations describing the nature and structure of a domain of discourse
  - in order to account for the competent use of vocabulary in real situations
- Gruber: "Explicit and formal specifications of a conceptualization of a domain"

What is a conceptualization

- Formal structure of (a piece of) reality as perceived and organized by an agent, independently of:
  - the vocabulary used
  - the actual occurence of a specific situation
- Different situations involving the same objects, described by different vocabularies, may share the same conceptualization
  - Language 1: Apple
  - Language 2: tafaha

Ontologies vs classifications

- Classifications focus on:
  - access, based on pre-determined criteria (encoded by syntactic keys)
- Ontologies focus on:
  - meaning of terms
  - nature and structure of a domain

Ontologies vs Knowledge bases

Knowledge base:

- Assertional component
  - reflects specific states of affairs
  - designed for problem solving
- Terminological component (ontology)
  - independent of states of affairs
  - designed to support terminological services
- Ontological formulas are invariant, necessary information

The formal tools of ontological analysis

- Theory of Parts (Mereology)
- Theory of Unity and Plurality
- Theory of Essence and Identity
- Theory of Dependence
- Theory of Composition and Constitution
- Theory of Properties and Qualities
- Theory of Space and Time

Formal Ontology

Theory of formal distinctions and connections within

- entities of the world, as we perceive it (particulars)
- categories we use to talk about such entities (universals, categories, properties)

Formal Ontology

Theory of formal distinctions and connections within

- entities of the world, as we perceive it (particulars)
- categories we use to talk about such entities (universals, categories, properties)

We will not discuss any of this here but rather focus on how to *build*, *find* and *use* ontologies.

Ontologies

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*

Ontologies

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*
- *instances* of classes are individuals satisfying the classes' intension
  - my arm, the influenza I had last year, one ethanol molecule, etc.

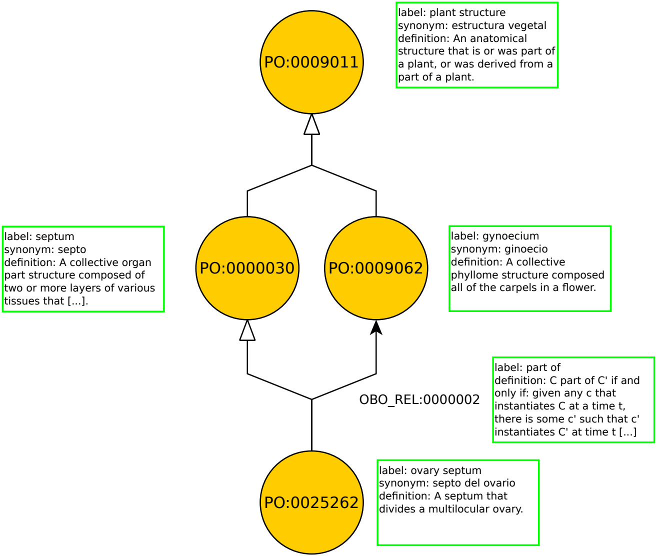
Ontologies

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*
- *instances* of classes are individuals satisfying the classes' intension
  - my arm, the influenza I had last year, one ethanol molecule, etc.
- *relations* between instances arise from interactions, configurations, etc., of individuals
  - my arm is **part of** me, the **duration of** my influenza was 10 days
  - formal and material relations

Ontologies

- *classes* represent kinds of things in the world
  - *Arm*, *Apoptosis*, *Influenza*, *Homo sapiens*, *Drinking behavior*, *Membrane*
- *instances* of classes are individuals satisfying the classes' intension
  - my arm, the influenza I had last year, one ethanol molecule, etc.
- *relations* between instances arise from interactions, configurations, etc., of individuals
  - my arm is **part of** me, the **duration of** my influenza was 10 days
  - formal and material relations
- *axioms* specify our knowledge of the domain
  - every instance of *Hand* is a **part of** an instance of *Arm*
  - $\forall x, y, z(Q(x) \wedge inheresIn(x, y) \wedge inheresIn(x, z) \rightarrow y = z)$
  - $\forall x, y(\neg po(x, y) \rightarrow \exists z(po(z, y) \wedge \neg o(z, x)))$

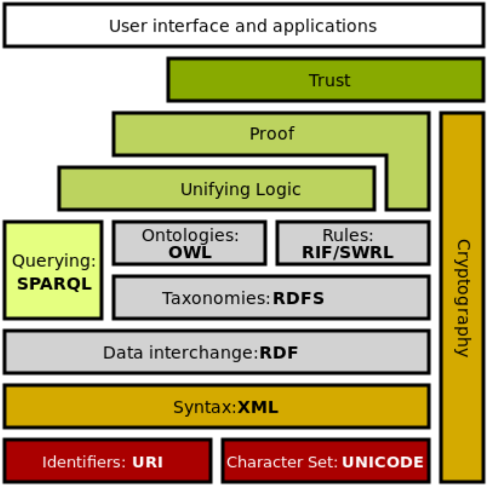
Ontologies


PO:0009011
label: plant structure
synonym: estructura vegetal
definition: An anatomical structure that is or was part of a plant, or was derived from a part of a plant.
PO:0000030
label: septum
synonym: septo
definition: A collective organ part structure composed of two or more layers of various tissues that [...].
PO:0009062
label: gynoecium
synonym: ginoecio
definition: A collective phyllome structure composed all of the carpels in a flower.
OBO_REL:0000002
label: part of
definition: C part of C' if and only if: given any c that instantiates C at a time t, there is some c' such that c' instantiates C' at time t [...]
PO:0025262
label: ovary septum
synonym: septo del ovario
definition: A septum that divides a multilocular ovary.

The Semantic Web

- A set of standards for common data formats and exchange protocols
- Extension of web standards
- Resource Description Framework (RDF)
- Web Ontology Language (OWL)
- SPARQL Protocol and RDF Query Language (SPARQL)
- ...

The Semantic Web



Web Ontology Language (OWL)

- OWL 2 is based on the Description Logic $\mathcal{SROIQ}(\mathcal{D})$
- $\mathcal{ALC}$ ($\sqcap$, $\sqcup$, $\neg$, $\equiv$, $\sqsubseteq$) with
  - complex role inclusions: $r \circ s \subseteq r$
  - role hierarchy: $r \subseteq s$
  - role transitivity $r \circ r \subseteq r$
  - nominals: $\{a_1, ..., a_n\}$ as concept constructor
  - qualified number restrictions: $(\leq nr.Q)$
  - datatype properties: $\exists r.[\geq n(Integer)]$

Terminology

- Instances
- Properties
  - Object properties
  - Datatype properties
- Classes
- Meta-classes
  - OWL Full
  - Punning
- Axiom
  - Class axioms: Subclass, Equivalent class, Disjoint class
  - Property axioms
- Ontology
- OWL: Web Ontology Language

Syntax

- originally an extension of RDF and RDF Schema
- several different syntaxes

Consider the axiom $\mathit{Parent} \equiv \mathit{Human} \sqcap \exists \mathit{hasChild}.\top$

Functional Syntax

```
EquivalentClasses(:Parent
  ObjectSomeValuesFrom(:hasChild owl:Thing))
```

RDF/XML Syntax

```
<owl:Class rdf:about="http://example.com/demo-ontology.owl#Parent">
    <owl:equivalentClass>
        <owl:Restriction>
            <owl:onProperty rdf:resource="http://example.com/demo-ontology.owl#hasChild"/>
            <owl:someValuesFrom rdf:resource="&owl;Thing"/>
        </owl:Restriction>
    </owl:equivalentClass>
</owl:Class>
```

RDF Turtle Syntax

```
:Parent rdf:type owl:Class ;

        owl:equivalentClass [ rdf:type owl:Restriction ;
                              owl:onProperty :hasChild ;
                              owl:someValuesFrom owl:Thing
                            ] .
```

OWL/XML Syntax

```
<EquivalentClasses>
    <Class IRI="#Parent"/>
    <ObjectSomeValuesFrom>
        <ObjectProperty IRI="#hasChild"/>
        <Class abbreviatedIRI="owl:Thing"/>
    </ObjectSomeValuesFrom>
</EquivalentClasses>
```

Manchester OWL Syntax

```
Class: Parent
    EquivalentTo:
        hasChild some owl:Thing
```

Manchester OWL Syntax

DL Syntax	Manchester Syntax	Example
$C \sqcap D$	C and D	Human and Male
$C \sqcup D$	C or D	Male or Female
$\neg C$	not C	not Male
$\exists R.C$	R some C	hasChild some Human
$\forall R.C$	R only C	hasChild only Human
$(\geq nR.C)$	R min n C	hasChild min 1 Human
$(\leq nR.C)$	R max n C	hasChild max 1 Human
$(= nR.C)$	R exactly n C	hasChild exactly 1 Human
$\{a\} \sqcup \{b\} \sqcup ...$	{a b ...}	{John Robert Mary}

OWL classes and namespaces

- $\bot$ is `owl:Nothing`
- $\top$ is `owl:Thing`
- `owl:` is a *namespace* (`http://www.w3.org/2002/07/owl#`)
- `owl:Thing` expands to `http://www.w3.org/2002/07/owl#Thing` (a class IRI)
- all OWL entities (ontologies, classes, properties, instances) are referred to by an IRI
- namespaces define a common (IRI-)prefix, e.g.,
  - `rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#`
  - `rdfs: http://www.w3.org/2000/01/rdf-schema#`
- can define own namespaces:

```
Namespace: mynamespace <http://www.kaust.edu.sa#>
Class: mynamespace:Student # http://www.kaust.edu.sa#Student
```

Object properties

- Object property characteristics:
  - transitive
  - symmetric, asymmetric
  - reflexive, irreflexive
  - functional, inverse functional
  - inverse of
- Domain and range

Annotation properties

- OWL entities (classes, properties, axioms, ontologies, etc.) can have *annotations*
- outside of OWL semantics (unless for OWL Full)
- useful to add labels, synonyms, explanation, (textual) definitions, authoring information, versions, etc.
- predefined: `rdfs:label`, `owl:versionInfo`, `rdfs:comment`, `rdfs:seeAlso`, `rdfs:isDefinedBy`
- Dublin Core

OWL Reasoning

- Classification: compute the most specific sub- and super-classes for each named class in an OWL ontology
- Subsumption: find all sub-, super- or equivalent classes of an OWL class description
- Consistency: find contradictions in OWL knowledge base
- Instantiation: is *a* and instance of *C*?

Complexity of reasoning in OWL

- OWL 2 ($\mathcal{SROIQ}$) is 2NEXPTIME-complete
- OWL (1) ($\mathcal{SHOIN}$) is NEXPTIME-complete
- OWL Lite ($\mathcal{SHIF}$) is EXPTIME-complete

OWL profiles

- OWL 2 EL: PTIME-complete
- OWL 2 RL: PTIME-complete
- OWL 2 QL: AC^0 w.r.t. data size

OWL 2 EL

- Class axioms:
  - subclass, equivalent class, disjoint class
- Object property axioms:
  - domain and range restrictions, property inclusion, property chains, property equivalence, transitive and reflexive properties
- Class descriptions:
  - intersection, existential quantification, enumerations to a single individual
- Assertions: all

Why OWL?

- OWL exploits 20+ years of research on Description Logic
- well-defined semantics
- complexity and decidability well understood
- known algorithms
- scalability demonstrated in practise

Why OWL?

Major benefit is the large number of tools and infrastructure:

- Editors: Protege, WebProtege
- Reasoners: HermiT, Pellet, FaCT++, ELK, KAON2, RACER,...
- Explanation, justification
- Modularization
- APIs (esp. the OWL API)

OWL vs Databases

Database	OWL Ontology
Closed World Assumption	Open World Assumption
Unique Name Assumption	No UNA
Schema constraints data structure	Axioms behave like inference rules

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: Is Draco a friend of Harry Potter?

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: Is Draco a friend of Harry Potter?

- DB: No
- OWL: Don't know

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: How many friends has Harry Potter?

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig

Query: How many friends has Harry Potter?

- DB: 2
- OWL: At least 1

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig
- RonWeasley $\neq$ HermioneGranger
- HarryPotter: hasFriend only {HerminoeGranger RonWeasley}

Query: How many friends has Harry Potter?

Examples: OWL vs Databases

- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard
- HarryPotter: Wizard
- DracoMalfoy: Wizard
- HarryPotter hasFriend RonWeasley
- HarryPotter hasFriend HermioneGranger
- HarryPotter hasPet Hedwig
- RonWeasley $\neq$ HermioneGranger
- HarryPotter: hasFriend only {HerminoeGranger RonWeasley}

Query: How many friends has Harry Potter?

- DB: 2
- OWL: 2

Examples: OWL vs Databases

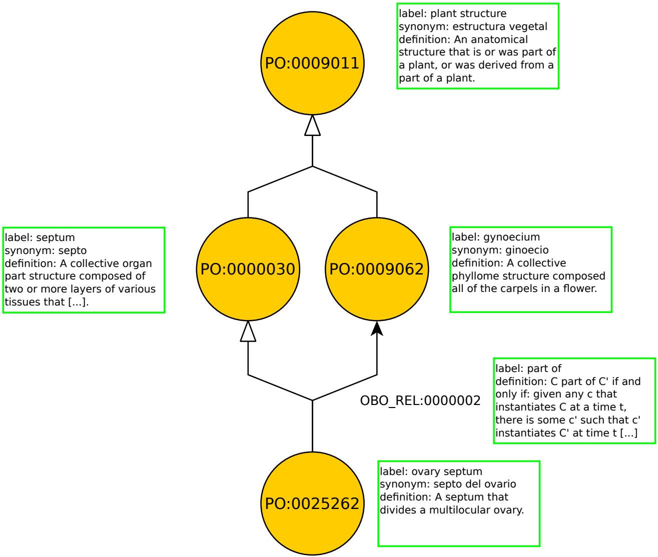
- hasPet some owl:Thing SubclassOf: Human
- Phoenix SubclassOf: petOf only Wizard

Adding new facts:

- Dumbledore: Human
- Fawkes: Phoenix
- Fawkes petOf Dumbledore

- DB: Update rejects, constrain violation
- OWL: infer that Dumbledore is a Wizard

Ontologies


PO:0009011
label: plant structure
synonym: estructura vegetal
definition: An anatomical structure that is or was part of a plant, or was derived from a part of a plant.
PO:0000030
label: septum
synonym: septo
definition: A collective organ part structure composed of two or more layers of various tissues that [...].
PO:0009062
label: gynoecium
synonym: ginoecio
definition: A collective phyllome structure composed all of the carpels in a flower.
OBO_REL:0000002
label: part of
definition: C part of C' if and only if: given any c that instantiates C at a time t, there is some c' such that c' instantiates C' at time t [...]
PO:0025262
label: ovary septum
synonym: septo del ovario
definition: A septum that divides a multilocular ovary.

The axiomatic method

Using OWL, how do we “define” the class *ovary septum*?

The axiomatic method

Using OWL, how do we "define" the class *ovary septum*?

- `'ovary septum' equivalentTo: septum and divides some 'multilocular ovary'`

The axiomatic method

Using OWL, how do we “define” the class *ovary septum*?

- `'ovary septum' equivalentTo: septum and divides some 'multilocular ovary'`
    - but what is a *septum* and a *multilocular ovary*?

The axiomatic method

Using OWL, how do we "define" the class *ovary septum*?

- `'ovary septum' equivalentTo: septum and divides some 'multilocular ovary'`
  - but what is a *septum* and a *multilocular ovary*?
  - `'multilocular ovary' equivalentTo: ovary and has-quality some multilocular`

The axiomatic method

Using OWL, how do we "define" the class *ovary septum*?

- `'ovary septum' equivalentTo: septum and divides some 'multilocular ovary'`
  - but what is a *septum* and a *multilocular ovary*?
  - `'multilocular ovary' equivalentTo: ovary and has-quality some multilocular`
- `'ovary septum' equivalentTo: septum and divides some (ovary and has-quality some multilocular)`
  - and what's an *ovary*, *multilocular*, and *has-quality*?
- We need axioms to give meaning to our definitions!

The axiomatic method

Using OWL, how do we “define” the class *ovary septum*?

- `'ovary septum' equivalentTo: septum and divides some 'multilocular ovary'`
  - but what is a *septum* and a *multilocular ovary*?
  - `'multilocular ovary' equivalentTo: ovary and has-quality some multilocular`
- `'ovary septum' equivalentTo: septum and divides some (ovary and has-quality some multilocular)`
  - and what's an *ovary*, *multilocular*, and *has-quality*?
- We need axioms to give meaning to our definitions!
- has-quality is InverseFunctional
- `'ovary septum' subClassOf: 'part of' some gynoecium`
- ...

Ontologies and graphs

- Ontology $\mathcal{O}$
- Graph G = (V, E)
- V := classes(O)
- If class C_1 SubClassOf C_2, create edge $isa(C_1, C_2)$
- How about other edges?

Ontologies and graphs

- Ontology $\mathcal{O}$
- Graph G = (V, E)
- V := classes(O)
- If class C_1 SubClassOf C_2, create edge $isa(C_1, C_2)$
- How about other edges?
- OBO Relation Ontology: relations are patterns
  - PartOf: $PartOf(X, Y) \iff$ `X SubClassOf: partOf some Y`
- Generate an edge labeled R between X and Y iff $O \models R(X, Y)$, i.e., if the statement defined by the relational pattern R is made true for classes X and Y.
  - needs an OWL reasoner