Knowledge Representation (Part 3)

Dr. Mohammad Kazemifard

Session 5

Outline

- Ontology
  - Introduction
  - Definitions
- Ontology Languages
  - RDF
  - OWL

Where does it come from?

- ontology n.

  1692; Greek . phil. *onto-* “being” + *-logia* “study of”
- Philosophy
  - The study of what is, what has to be true for something to exist, the kinds of things that can exist
- AI and computer science
  - Something exists if it can be represented, described, defined (in a formal, hence, machine-interpretable way).

Ontologies

Vocabulary + Structure = Taxonomy

Taxonomy + Relationships, Constraints and Rules = Ontology

Ontology + Instances = Knowledge Base

Ontologies (cont.)

- Ontologies are about vocabularies and their meanings, with explicit, expressive, and well-defined semantics, possibly machine-interpretable.
- *“Ontology is a formal specification of a conceptualization.” Gruber, 1993*
- Main elements of an ontology:
  - Concepts
  - Relationships
    - Hierarchical
    - Logical
  - Properties
  - Instances (individuals)

A Definition

- Informal
  - Terms
    - from a specific domain
    - uniquely defined, usually via natural language definitions
  - May contain additional semantics in the form of informal relations
  - machine-processing is difficult
  - Examples
    - Controlled vocabulary
    - Glossary
    - Thesaurus

A Definition

- Formal
  - Domain-specific vocabulary
  - Well-defined semantic structure
    - Classes/concepts/types
      - E.g., a class { Publication } represents all publications
      - E.g., a class { Publication } can have subclasses { Newspaper }, { Journal }
    - Instances/individuals/objects
      - E.g., the newspaper *Le Monde* is an instance of the class { Newspaper }
    - Properties/roles/slots
      - Data
        - E.g., the class { Publication } and its subclasses { Newspaper }, { Journal } have a data property { numberOfPages }
      - Object
        - E.g., the class { Publication } and its subclasses { Newspaper }, { Journal } have an object property { publishes }
  - Is machine-processable

Semantic Web

- Provide shared data structures to exchange information between software agents
- Can be explicitly used as annotations in web sites
- Can be used for knowledge-based services using other web resources
- Can help to structure knowledge to build domain models (for other purposes)

Meaning is in Connections

is made from

a G

m

Grape

e

a

i

e

n

r

f

W

m i

s

p

o

Wine d e r

For machines…

The meaning of the document is not defined.
Machines cannot understand it.

⇑πηονε ισ μαδε βψ Aππλε

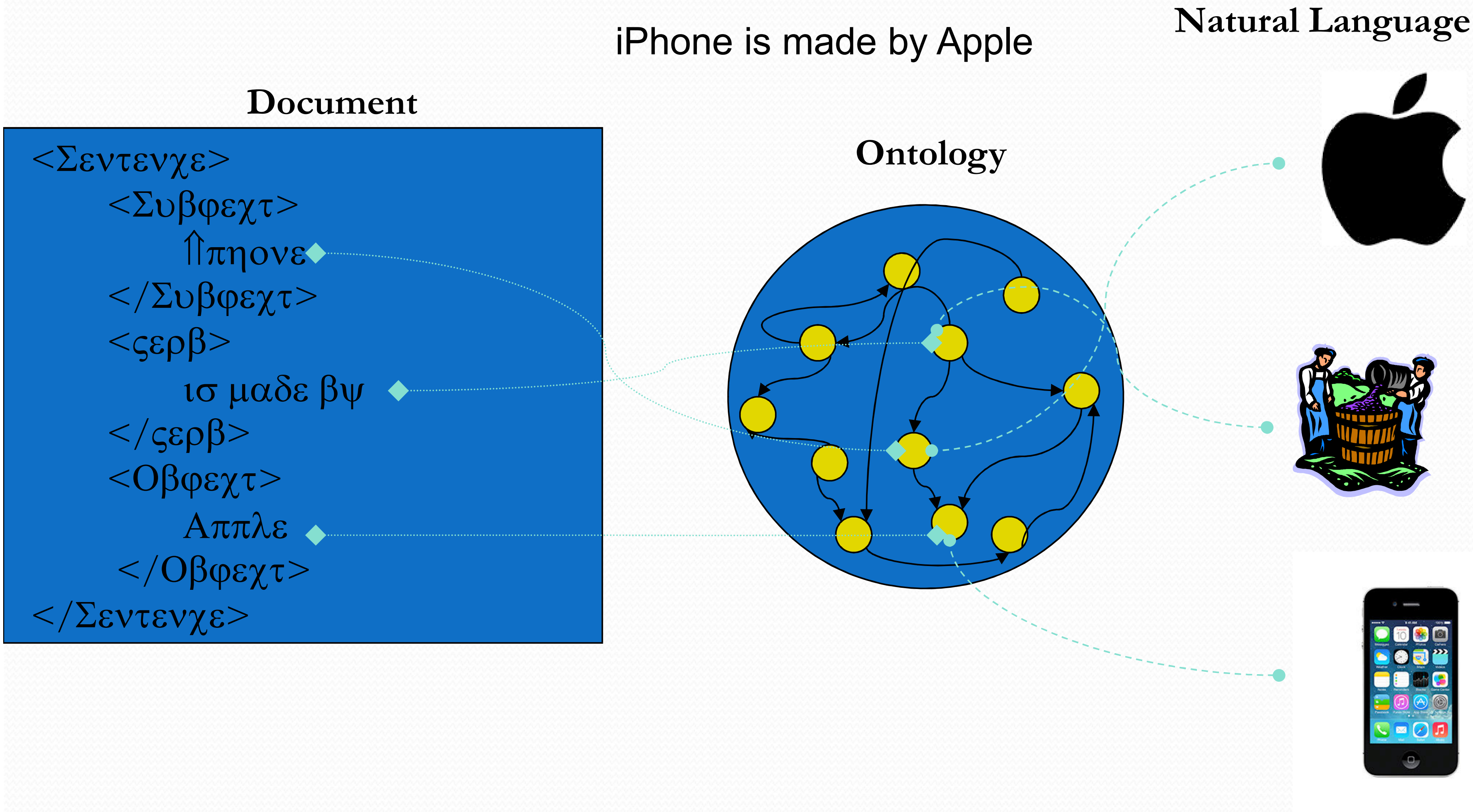
We are defining the structure of document by XML
but now the meaning of the structure is not defined.

```
<Sentence>
        <Subject>
                ⇑πηονε
        </Subject>
        <Verb>
                ισ μαδε βψ
        </Verb>
        <Object>
                Aππλε
         </Object>
</Sentence>
```

XML document

```
<Σεντενχε>
        <Συβφεχτ>
                ⇑πηονε
        </Συβφεχτ>
        <ςερβ>
                ισ μαδε βψ
        </ςερβ>
        <Οβφεχτ>
                Aππλε
         </Οβφεχτ>
</Σεντενχε>
```

Ontology gives the meaning…



iPhone is made by Apple

Natural Language

Document

```
<Σεντενχε>
    <Συβφεχτ>
        ⇑πηονε
    </Συβφεχτ>
    <ςερβ>
        ισ μαδε βψ
    </ςερβ>
    <Οβφεχτ>
        Αππλε
    </Οβφεχτ>
</Σεντενχε>
```

Ontology

Why develop ontologies?

- To share knowledge
  - E.g., using an ontology for integrating terminologies

- To reuse domain knowledge
  - E.g., geography ontology

- To make domain assumptions explicit
  - Facilitate knowledge management
  - Enable new users to learn about the domain

- To distinguish domain knowledge from operational knowledge
  - e.g., biblio metadata

What they are good for

- Search
  - Concept-based query
    - User uses own words, language
    - e.g., "*Jordan*" as a name of *Basket-ball player* and name of a *country*
  - Related terms
  - Intelligent query expansion: "fishing vessels in China" expands to "fishing vessels in Asia"
- Consistency checking
  - e.g., "Goods" has a property called "price" that has a value restriction of number
- Interoperability support
  - Terms defined in expressive ontologies allow for mapping precisely how one term relates to another

Ontology Languages

- Graphical notations
  - Semantic networks
  - UML
  - RDF (Resource Description Framework)
- Logic based
  - OWL (Ontology Web Language)
  - Rules
- Object oriented model
  - Class
  - Object
  - properties

Conclusion

- Ontology is a formal specification of a conceptualization
- Ontology usage
- Ontology languages

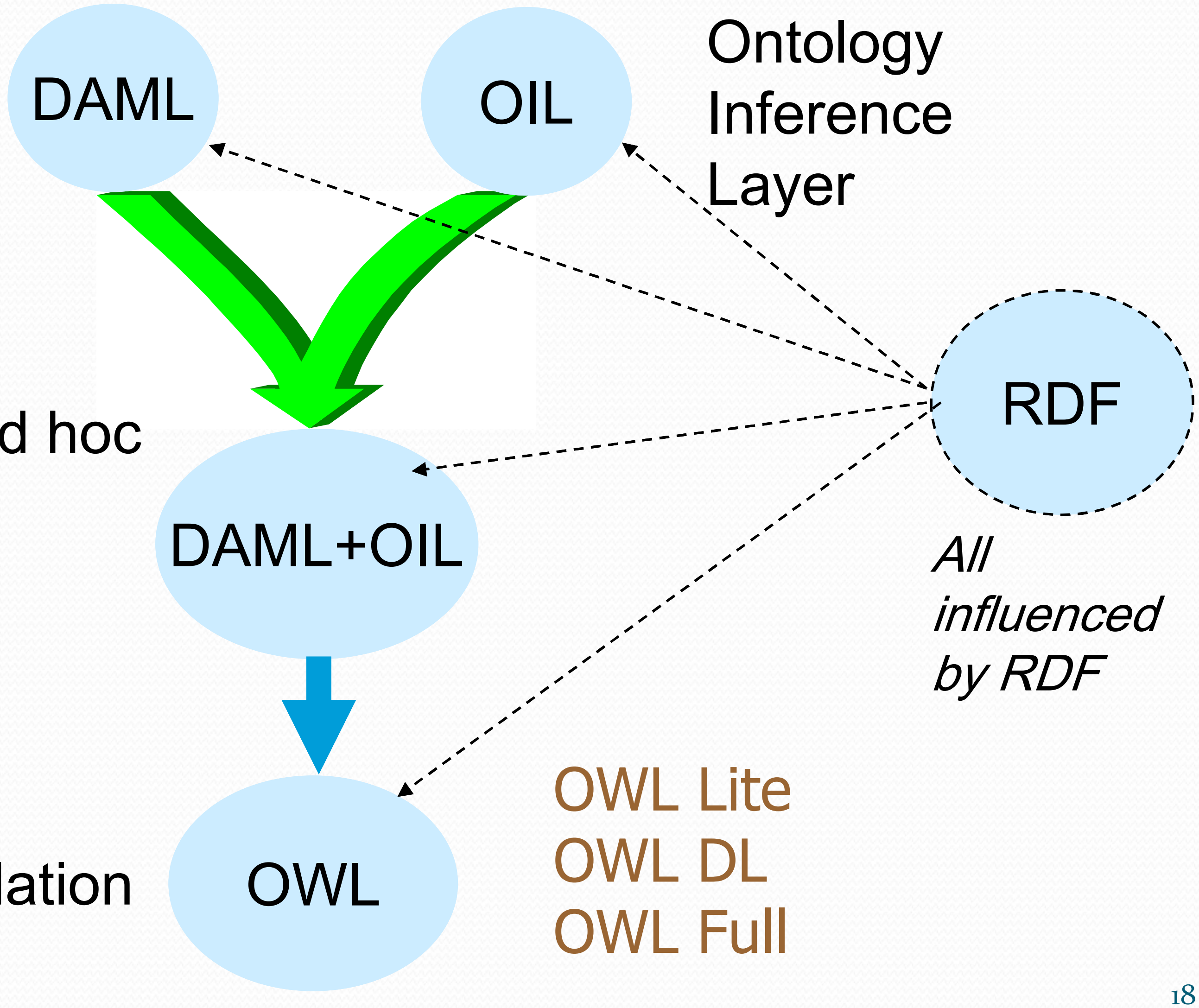
OWL (Ontology Web Language)

- OWL is now a W3C Recommendation
- The **purpose** of OWL is identical to RDFS i.e. to provide an XML vocabulary to define classes, properties and their relationships.
  - RDFS enables us to express very rudimentary relationships and has limited inferencing capability.
  - OWL enables us to express **much richer relationships**, thus yielding a much enhanced inferencing capability.
- The **benefit** of OWL is that it facilitates a much greater degree of inference than you get with RDF Schema.

Ontology Languages

- **RDF(S)** (Resource Description Framework (Schema))
- **OIL** (Ontology Interchange Language)
- **DAML+OIL** (DARPA Agent Markup Language + OIL)
- **OWL** (Ontology Web Language)
- **XOL** (XML-based Ontology Exchange Language)
- **SHOE** (Simple HTML Ontology Extension)
- **OML** (Ontology Markup Language)

Origins of OWL



DARPA
Agent Markup
Language

DAML

OIL

Ontology
Inference
Layer

RDF

EU/NSF Joint Ad hoc
Committee

DAML+OIL

*All
influenced
by RDF*

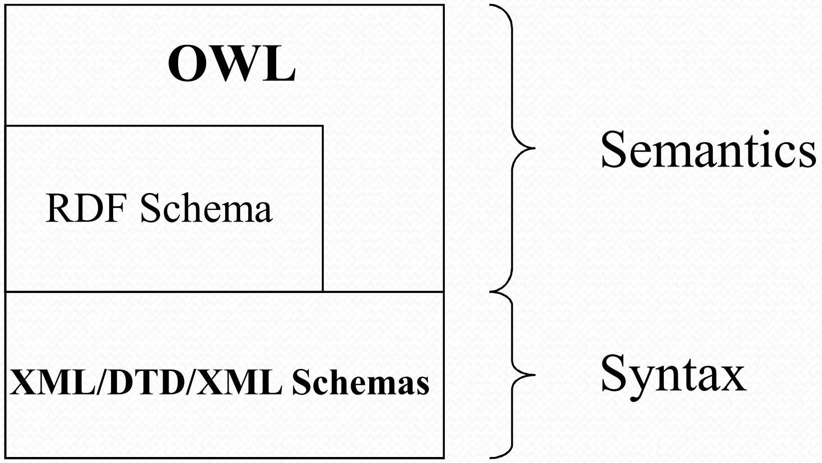
A W3C
Recommendation

OWL

OWL Lite
OWL DL
OWL Full

OWL

- OWL and RDF Schema enable rich *machine-processable semantics*



RDFS

```
<rdfs:Class rdf:ID="River">
        <rdfs:subClassOf rdf:resource="#Stream"/>
</rdfs:Class>
```

OWL

```
<owl:Class rdf:ID="River">
        <rdfs:subClassOf rdf:resource="#Stream"/>
</owl:Class>
```

Why Build on RDF

- Provides basic ontological primitives
  - Classes and relations (properties)
  - Class (and property) hierarchy

- Can exploit existing RDF infrastructure

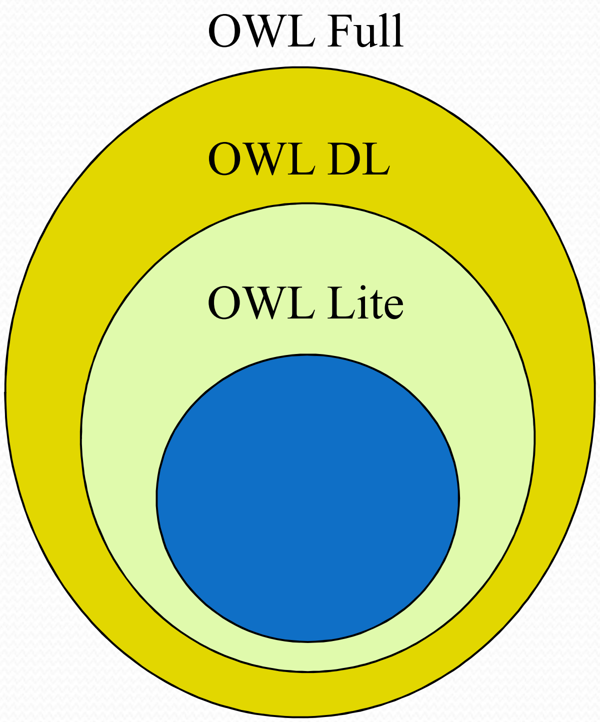
- Provides mechanism for using ontologies
  - RDF triples assert facts about resources
  - Use vocabulary from DAML+OIL ontologies

OWL *Design Goals*

- Shared ontologies
- Ontology evolution
- Ontology interoperability
- Inconsistency detection
- Expressivity vs. scalability
- Ease of use
- Compatibility with other standards
- Internationalization

Versions of OWL

- Depending on the intended usage, OWL provides three increasingly expressive sublanguages



Full: Very expressive,
no computation guarantees

DL (Description Logic): Maximum
expressiveness, computationally complete

Lite: Simple classification hierarchy
with simple constraints.

Comparison of versions

- **Full:**
  - ☑ We get the full power of the OWL language.
  - ☒ It is very difficult to build a tool for this version.
  - ☒ The user of a fully-compliant tool may not get a quick and complete answer.

- **DL/Lite:**
  - ☑ Tools can be built more quickly and easily
  - ☑ Users can expect responses from such tools to come quicker and be more complete.
  - ☒ We don't have access to the full power of the language.

Describing classes in OWL

OWL vs. RDFS

- OWL allows greater expressiveness
  - Abstraction mechanism to group resources with similar characteristics
  - Much more powerful in describing constraints on relations between classes
  - Property transitivity, equivalence, symmetry, etc.
  - …
- Extensive support for reasoning

OWL Ontologies

- What’s inside an OWL ontology
  - Classes + class-hierarchy
  - Properties (Slots) / values
  - Relations between classes (inheritance, disjoints, equivalents)
  - Restrictions on properties (type, cardinality)
  - Characteristics of properties (transitive, …)
  - Annotations
  - Individuals

- Reasoning tasks: classification, consistency checking

Classes

- What is a Class?
  - e.g., person, pet, old
  - a collection of individuals (object, things, . . .)
  - a way of describing part of the world
  - an object in the world (OWL Full)

Properties

- What is a Property?
  - e.g., has_father, has_pet, service_number
  - a collection of relationships between individuals (and data)
  - a way of describing a kind of relationship between individuals
  - an object in the world (OWL Full)

OWL Properties

Object

Properties

Ana → owns → Cuba

Is range a

literal / typed value ?

then ERROR

Data type

Properties

Ana → age → 25

- XML Schema data types supported
  - DB people happy

What’s in OWL, but not in RDF

- Ability to be distributed across many systems
- Scalable to Web needs
- Compatible with Web standards for:
  - accessibility, and
  - Internationalization
- Open and extensible

Inverse Functional Properties

Y → p_1 → A

Z → p_1 → A

imply Z is the same as Y

(they describe the same)

What if Y, Z

where explicitly defined as “*different*” ?

An Example OWL ontology

```
<owl:Class rdf:ID=“Person” />
<owl:Class rdf:ID=“Man”>
    <rdfs:subClassOf rdf:resource=“#Person” />
    <owl:disjointWith rdf:resource=“#Woman” />
</owl:Class>
<owl:Class rdf:ID=“Woman”>
   <rdfs:subClassOf rdf:resource=“#Person” />
   <owl:disjointWith rdf:resource=“#Man” />
</owl:Class>
<owl:Class rdf:ID=“Father”>
   <rdfs:subClassOf rdf:resource=“Man” />
   <owl:Restriction owl:minCardinality="1">
          <owl:onProperty rdf:resource="#hasChild" />
   </owl:Restriction>
</owl:Class>
<owl:ObjectProperty rdf:ID=“hasChild">
          <rdfs:domain rdf:resource="#Parent" />
          <rdfs:range rdf:resource="#Person" />
</owl:ObjectProperty>
```

Proteqe