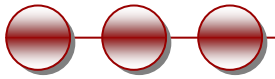
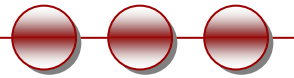


CMSC 671
Fall 2010

Thu 10/28/10

Description Logics
Ontologies in OWL
Semantic Web

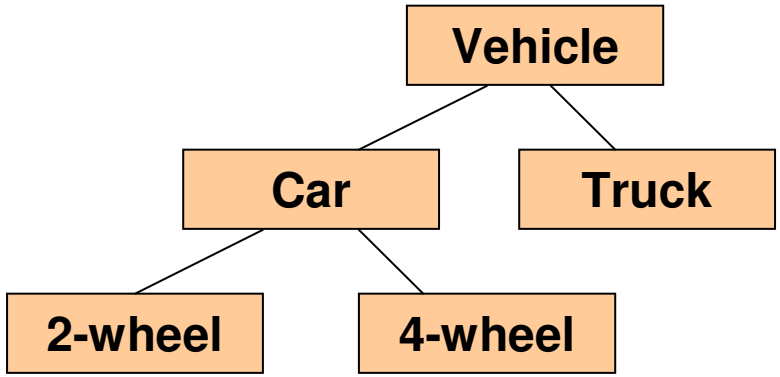
Prof. Laura Zavala, laura.zavala@umbc.edu, ITE 373, 410-455-8775



Some content on the slides has been taken from the tutorials at
http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/

Ontologies

- Specification of a conceptualization
- Representations of concepts
- Explicit formal specifications of the terms in the domain and relations among them
- Usually represented as a type hierarchy

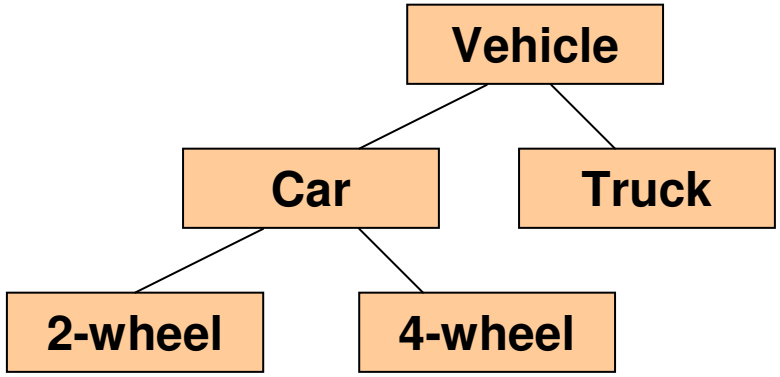


Why do we need an ontology

- To share common understanding of the structure of information among people or software agents
- To enable reuse of domain knowledge
- To make domain assumptions explicit
- To separate domain knowledge from the operational knowledge
  - We can merge, extend, and change
- To analyze domain knowledge

Description Logics

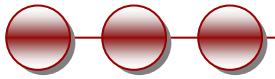
- Describe definitions and properties of categories
- Current systems take care to keep the languages simple, so that all inference can be done in polynomial time (in the number of objects)
  - ensuring tractability of inference



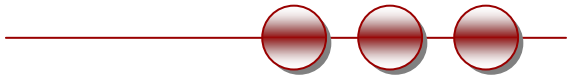
Description Logics (2)

- More expressive than propositional logic
- More efficient decision problems than first order predicate logic

- DL are of particular importance in providing a logical formalism for Ontologies and the Semantic Web



The Semantic Web: Means Many Things to Many People



What is the Semantic Web?

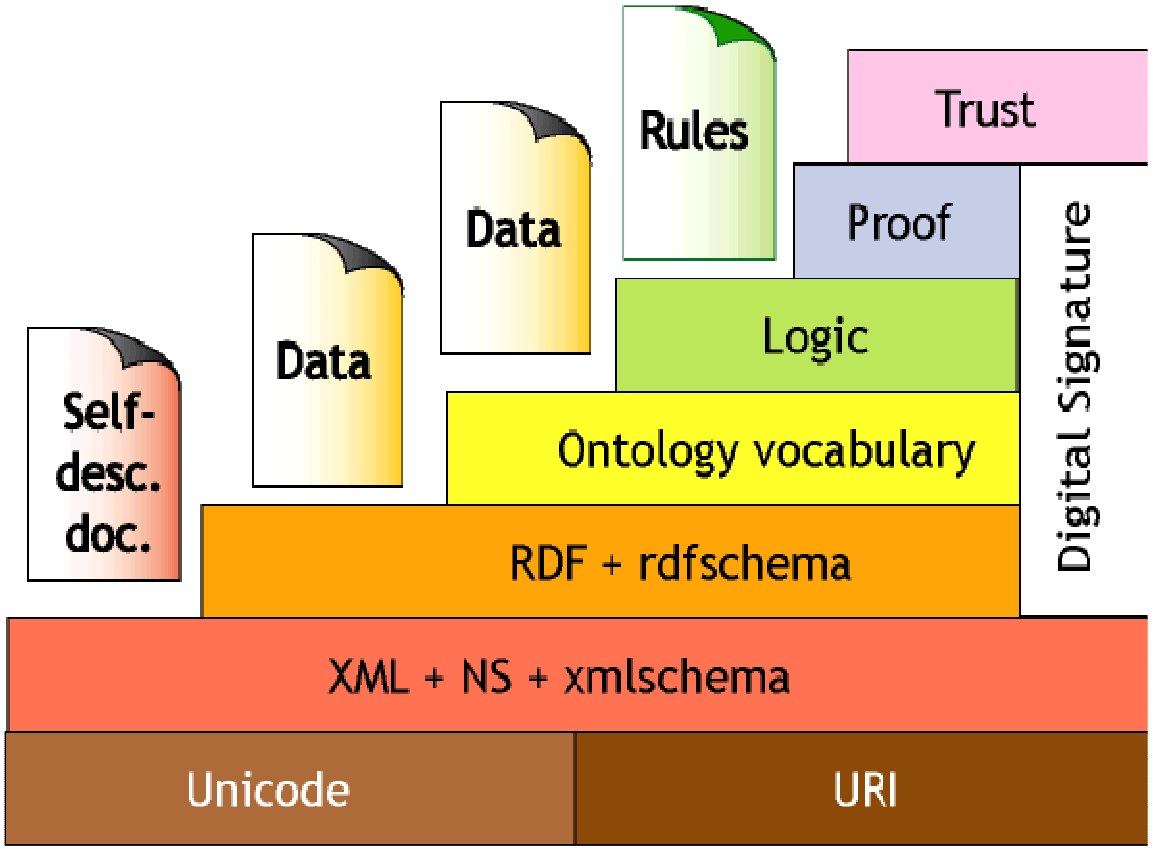
The **Semantic Web** is an evolving extension of the World Wide Web in which web content can be expressed not only in natural language, but also in a form that can be understood, interpreted and used by software agents, thus permitting them to find, share and integrate information more easily.

It derives from W3C director Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange.

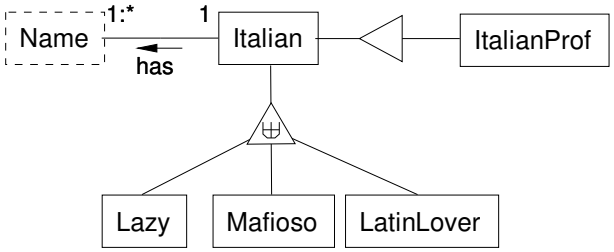
Semantic Web

- A group of methods and technologies to allow machines to understand the meaning - or "semantics" - of information on the World Wide Web
  - Resource Description Framework (RDF)
    - ABox
    - Triple Stores
      - Allegro, OntoBroker, Sesame, Virtuoso, RDFBroker, Jena, …
  - SPARQL is a protocol and query language for semantic web data sources (in RDF)
  - **Ontologies**
    - Web Ontology Language (OWL), RDFs, …
    - TBox
  - Rule Engines or Systems (Forward Chaining and Backward Chaining)
    - Pellet, Jena, Fact, Fact++, SwetRules, Jess, RuleML, …

The Layer Cake of the Semantic Web



Ontology Language ~ OSM/OWL

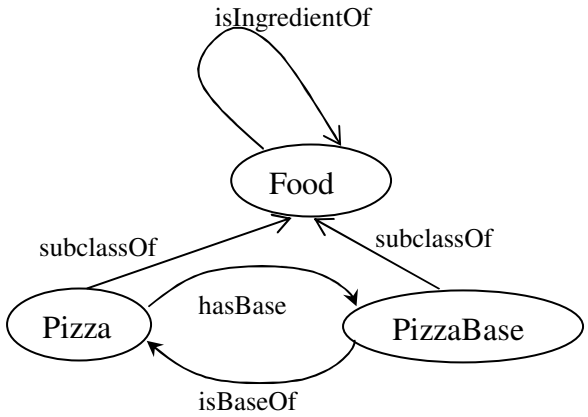


Lazy(x) => not ItalianProf(x)
ItalianProf(x) => not Lazy(x)
Mafioso(x) => not ItalianProf(x)
ItalianProf(x) => not Mafioso(x)

```
<owl:Ontology>
…
<owl:Class rdf:ID="Italian">
   <owl:unionOf rdf:parseType="owl:collection">
      <owl:Class ref:resource="#Lazy"/>
      <owl:Class ref:resource="#Mafioso"/>
      <owl:Class ref:resource="#LatinLover"/>
   …
   <owl:Restriction>
      <owl:onProperty rdf:resource="#hasName"/>
      <owl:minCardinality>
         <rdf:datatype="&xsd:#nonNegativeInteger">1</…>
      </owl:minCardinality>
      <owl:maxCardinality>
         …
<owl:Class rdf:ID="LatinLover">
   <rdfs:subClassOf rdf:resource="#Italian"/>
   <owl:disjointWith rdf:resource="#Lazy"/>
   <owl:disjointWith rdf:resource="#Mafioso"/>
…
<owl:ObjectProperty rdf:ID="hasName">
   <rdfs:domain rdf:resource="#Italian"/>
   <rdfs:range rdf:resource="#Name"/>
…
```

OWL Example: Pizza Ontology

- Ontologies in OWL are not trees, they are graphs.
  - They are not only a hierarchy of classes
  - They represent objects and relations about those objects (which includes subsumption relations, i.e. subclasses)




```
<owl:Class rdf:about="#Food"/>

<owl:Class rdf:about="#Pizza">
      <rdfs:subClassOf rdf:resource="#Food"/>
</owl:Class>

 <owl:ObjectProperty rdf:about="#hasBase">
      <rdf:type rdf:resource="&owl;FunctionalProperty"/>
      <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#Pizza"/>
      <rdfs:range rdf:resource="#PizzaBase"/>
      <rdfs:subPropertyOf rdf:resource="#hasIngredient"/>
      <owl:inverseOf rdf:resource="#isBaseOf"/>
   </owl:ObjectProperty>

...
```

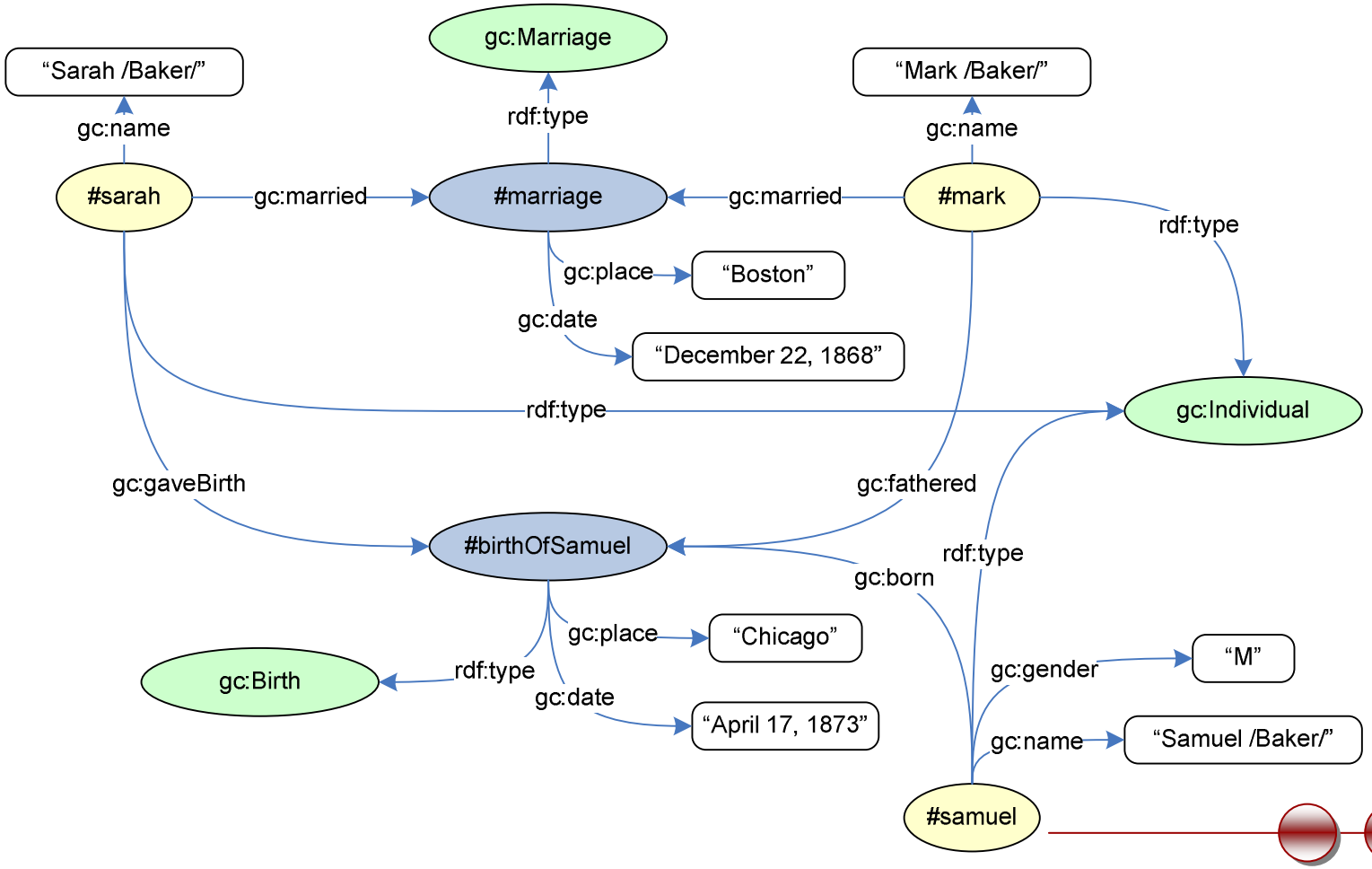
RDF & RDFS

- RDF (Resource Description Framework)
  - A triple model where every assertion is decomposed in three parts: (subject, predicate, object) for instance (tutorial.php, author, "Fabien").
  - The subject is a URI identifying a resource. The predicate is a binary relation identified by a URI. The object is either a URI identifying a resource or a literal value.
  - Each triple can be seen as a labeled arc and joining these arcs one obtains a graph that describes URI-identified resources and their relations.

- RDFS (RDF Schema)
  - RDFS is a set of primitives to describe **lightweight ontologies**
  - RDFS allows us to:
    - name and declare a vocabulary (name resource types and binary relation types called properties);
    - specify the signature of properties (type of the domain i.e. type of the subject and type of the range i.e. type of the object);
    - specify the (multiple)-inheritance links between the types of classes (subClassOf);
    - specify the (multiple)-inheritance links between the types of properties (subPropertyOf);

RDF Graph

(How to think about RDF)



SPARQL

(An RDF Query Language)

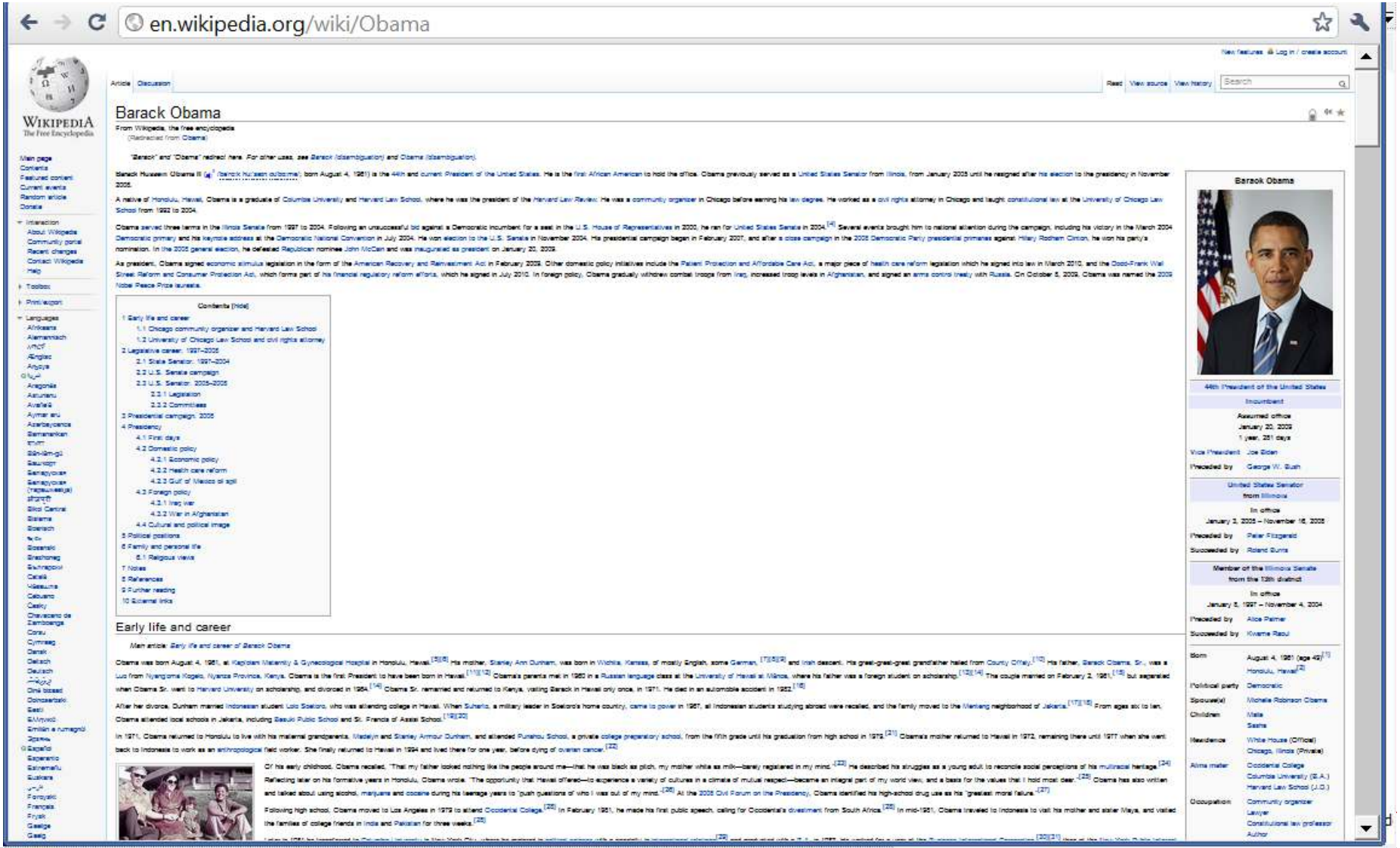
- Specifies queries over an RDF triple store
- Schema for the triple store (e.g. the Genealogy Core, gc)
- Example: get names and, if available, gender and birthdate of people born in the 1880's.

```
SELECT ?Name ?Gender ?BirthDate
WHERE { ?IndividualURI gc:name ?Name .
        OPTIONAL { ?IndividualURI gc:gender ?Gender ;
                                  gc:born ?Birth .
                   ?Birth gc:date ?BirthDate } .
        FILTER REGEX(?BirthDate, "188\\d")   }
```

The dbpedia knowledge base

- **DBpedia** is a community effort to extract structured information from Wikipedia and to make this information available on the Web. DBpedia allows you to ask sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data.

The dbpedia knowledge base



Dbpedia: knowledge extracted from Wikipedia Infoboxes

Barack Obama

44th President of the United States

Incumbent

Assumed office
January 20, 2009
1 year, 281 days

Vice President Joe Biden

Preceded by George W. Bush

United States Senator from Illinois

In office
January 3, 2005 – November 16, 2008

Preceded by Peter Fitzgerald

Succeeded by Roland Burris

Member of the Illinois Senate from the 13th district

In office
January 8, 1997 – November 4, 2004

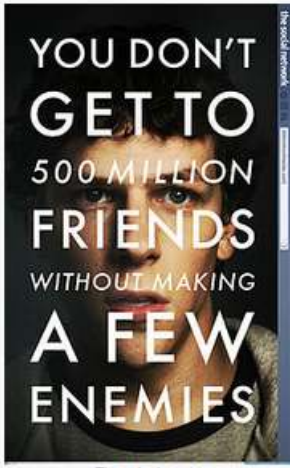
Preceded by Alice Palmer

Jon Bon Jovi

Jon Bon Jovi, 2007

Background information

Birth name	John Francis Bongiovi, Jr.
Also known as	Jon Bon Jovi
Born	March 2, 1962 (age 48) Perth Amboy, New Jersey,
Genres	Hard rock, glam metal, country rock, Rock
Occupations	Musician, songwriter, actor, philanthropist, producer
Instruments	Vocals, guitar, piano, harmonica, drums, percussion instruments, harp, bass guitar, banjo, french horn, trumpet, trombone, clarinet, cello
Years active	1978–present
Labels	Island, Mercury
Associated acts	Bon Jovi, Southside Johnny & The Asbury Jukes, Skid Row
Website	http://www.bonjovi.com



Theatrical poster

Directed by	David Fincher
Produced by	David Fincher Scott Rudin Kevin Spacey Dana Brunetti Michael De Luca Ceán Chaffin
Screenplay by	Aaron Sorkin
Based on	*The Accidental Billionaires* by Ben Mezrich
Starring	Jesse Eisenberg Andrew Garfield Justin Timberlake Brenda Song Rooney Mara Armie Hammer Max Minghella
Music by	Trent Reznor Atticus Ross
Cinematography	Jeff Cronenweth

The dbpedia knowledge base

- The DBpedia data set uses a large multi-domain ontology which has been derived from Wikipedia.
- The DBpedia data set currently describes 3.4 million “things” with over 1 billion “facts” (March 2010).

Querying dbpedia

- SPARQL
- Example queries
  - All soccer players, who played as goalkeeper for a club that has a stadium with more than 40.000 seats and who are born in a country with more than 10 million inhabitants
  - People who were born in Berlin before 1900
  - German musicians with German and English descriptions
  - German musicians who were born in Berlin
  - French films
  - First-person shooter computer games
  - Luxury cars
- http://wiki.dbpedia.org/OnlineAccess

Querying dbpedia

- You write your queries and try them at some public endpoint
  - http://DBpedia.org/sparql
  - http://DBpedia.org/snorql
- You can use some SPARQL query builder, e.g. http://querybuilder.dbpedia.org/

Querying dbpedia

- French Films

```
SELECT ?film
WHERE
    { ?film skos:subject
      <http://dbpedia.org/resource/Category:French_films>
    }
```

Other semantic data initiatives

- Freebase
- GeoNames
- Data.gov
  - *An Official Web Site of the United States Government*
  - Is now hosting one of the largest open collections of RDF datasets in the world!

OWL: Web Ontology Language

- Part of the Semantic Web Framework
- Explicit semantics for reasoning about systems of categories.
- Open World
- Different Flavours
  - Lite, DL, Full
  - EL, QL, RL
- Several Common Representations (syntax)
  - XML, N3, Abstract Syntax

OWL and RDF

- Resource Description Framework (RDF)
  - Describes a graph of nodes and arcs, each normally identified by a URI
  - RDF statements are triples
    - subject → predicate → object
    - Myhouse → isLocatedIn → Maryland
    - Maryland → isA → State
  - Semantics are limited and use is unconstrained compared to OWL

OWL and RDFS

- RDF Schema (RDFS)
  - Adds the notion of classes to RDF
  - Allows hierarchies of classes and properties
  - OWL has the same interpretation of some RDFS statements (subsumption, domain, and range)

OWL RDF/XML Syntax

- Several Common Representations (syntax)
  - XML, N3, Abstract Syntax

```
<owl:Class rdf:ID="SpicyPizza">
   <rdfs:label xml:lang="pt">PizzaTemperada</rdfs:label>
   <rdfs:comment xml:lang="en">Any pizza that has a spicy topping is a SpicyPizza</
  rdfs:comment>
   <owl:equivalentClass>
     <owl:Class>
       <owl:intersectionOf rdf:parseType="Collection">
         <owl:Class rdf:about="#Pizza"/>
         <owl:Restriction>
           <owl:onProperty>
             <owl:ObjectProperty rdf:about="#hasTopping"/>
           </owl:onProperty>
           <owl:someValuesFrom rdf:resource="#SpicyTopping"/>
         </owl:Restriction>
       </owl:intersectionOf>
     </owl:Class>
   </owl:equivalentClass>
 </owl:Class>
```

OWL2 EL

- class inclusion (**SubClassOf**)
- class equivalence (**EquivalentClasses**)
- class disjointness (**DisjointClasses**)
- object property inclusion (**SubObjectPropertyOf**) with or without property chains, and data property inclusion (**SubDataPropertyOf**)
- property equivalence (**EquivalentObjectProperties** and **EquivalentDataProperties**),
- transitive object properties (**TransitiveObjectProperty**)
- reflexive object properties (**ReflexiveObjectProperty**)
- domain restrictions (**ObjectPropertyDomain** and **DataPropertyDomain**)
- range restrictions (**ObjectPropertyRange** and **DataPropertyRange**)
- assertions (**SameIndividual**, **DifferentIndividuals**, **ClassAssertion**, **ObjectPropertyAssertion**, **DataPropertyAssertion**,**NegativeObjectPropertyAssertion**, and **NegativeDataPropertyAssertion**)
- functional data properties (**FunctionalDataProperty**)
- keys (**HasKey**)

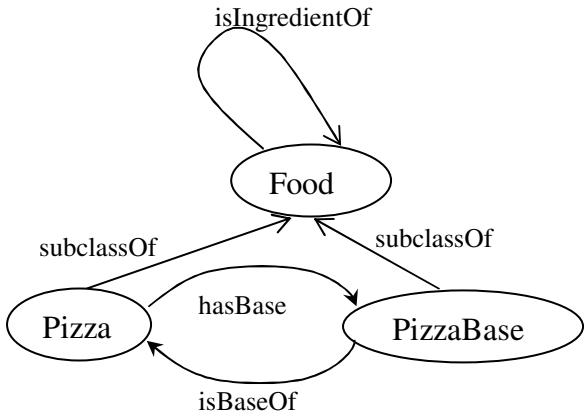
OWL Reasoning

- Reasoners are used to infer information that is not explicitly contained in the ontology
  - Consistency checking
  - Subsumption checking
  - Equivalence checking
  - Instantiation Checking

- Protégé
  - Fact++, Pellet, Racer, Kaon2

OWL Example: Pizza Ontology

- Ontologies in OWL are not trees, they are graphs.
  - They are not only a hierarchy of classes
  - They represent objects and relations about those objects (which includes subsumption relations, i.e. subclasses)



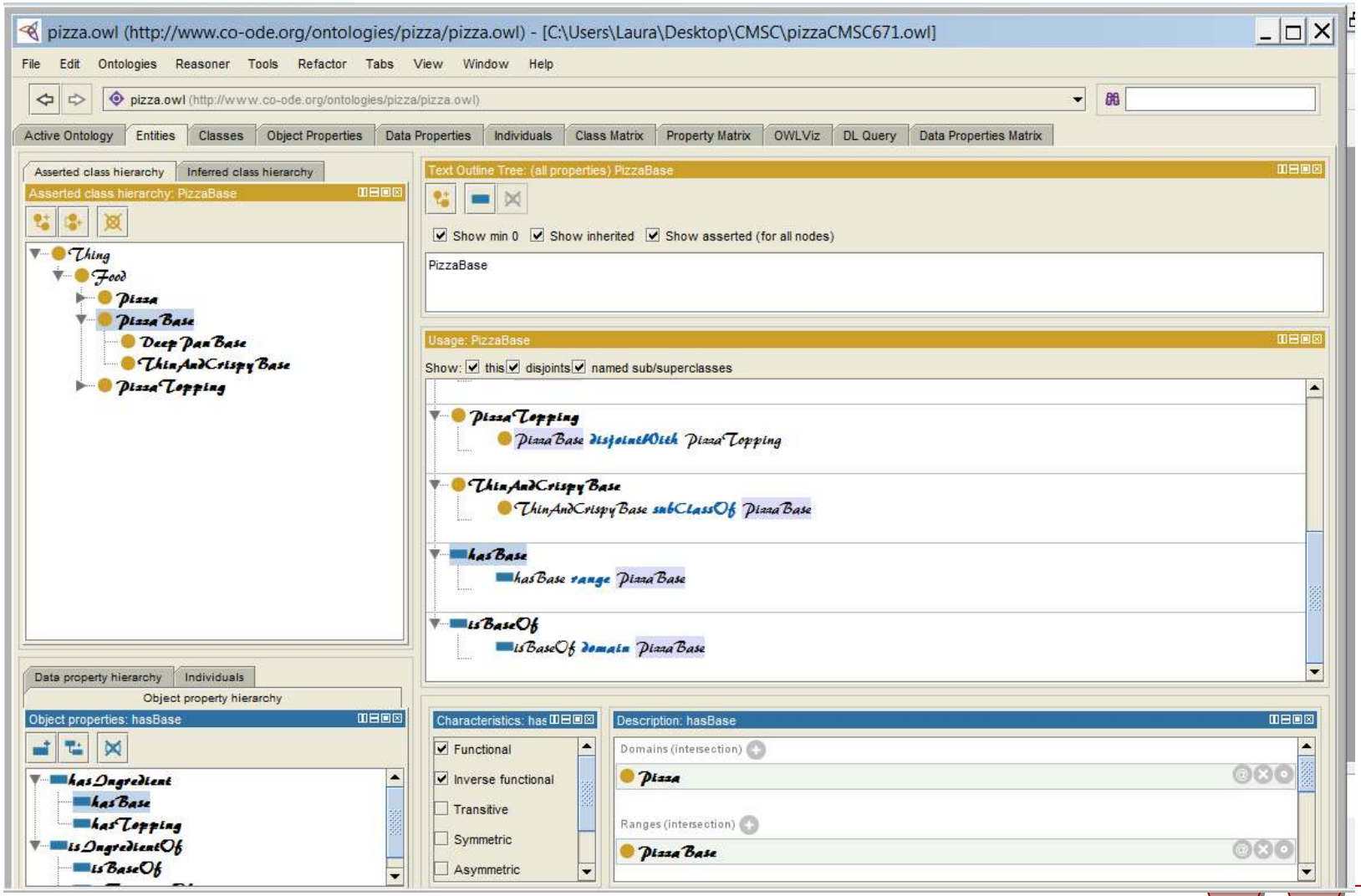

```
<owl:Class rdf:about="#Food"/>

<owl:Class rdf:about="#Pizza">
      <rdfs:subClassOf rdf:resource="#Food"/>
</owl:Class>

 <owl:ObjectProperty rdf:about="#hasBase">
      <rdf:type rdf:resource="&owl;FunctionalProperty"/>
      <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
      <rdfs:domain rdf:resource="#Pizza"/>
      <rdfs:range rdf:resource="#PizzaBase"/>
      <rdfs:subPropertyOf rdf:resource="#hasIngredient"/>
      <owl:inverseOf rdf:resource="#isBaseOf"/>
   </owl:ObjectProperty>

...
```

Protégé + OWL


pizza.owl (http://www.co-ode.org/ontologies/pizza/pizza.owl) - [C:\Users\Laura\Desktop\CMSC\pizzaCMSC671.owl]
File Edit Ontologies Reasoner Tools Refactor Tabs View Window Help
Active Ontology Entities Classes Object Properties Data Properties Individuals Class Matrix Property Matrix OWLViz DL Query Data Properties Matrix
Asserted class hierarchy Inferred class hierarchy
Asserted class hierarchy: PizzaBase
Thing
Food
Pizza
PizzaBase
DeepPanBase
ThinAndCrispyBase
PizzaTopping
Text Outline Tree: (all properties) PizzaBase
Show min 0 Show inherited Show asserted (for all nodes)
PizzaBase
Usage: PizzaBase
Show: this disjoints named sub/superclasses
PizzaTopping
PizzaBase disjointWith PizzaTopping
ThinAndCrispyBase
ThinAndCrispyBase subClassOf PizzaBase
hasBase
hasBase range PizzaBase
isBaseOf
isBaseOf domain PizzaBase
Data property hierarchy Individuals
Object property hierarchy
Object properties: hasBase
hasIngredient
hasBase
hasTopping
isIngredientOf
isBaseOf
Characteristics: has
Functional
Inverse functional
Transitive
Symmetric
Asymmetric
Description: hasBase
Domains (intersection)
Pizza
Ranges (intersection)
PizzaBase