CS 2740 Knowledge representation
Lecture 16

Semantic web

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Semantic web

```
<table>
  <tr>
     <td><strong>Title:</strong></td>
     <td>The Road</td>
  </tr>
  <tr>
     <td><strong>Author:</strong></td>
     <td>Cormac McCarthy</td>
  </tr>
  <tr>
     <td><strong>Price:</strong></td>
     <td>14.95</td>
  </tr>
</table>
```

html

Hypertext Markup Language (**HTML**)
a language for structuring text based Documents supplemented with interactive forms, embedded images and objects

versus

```
<book>
  <title>The Road</title>
  <author>Cormac McCarthy</author>
  <price>14.95</price>
</book>
```

xml

Extensible Markup Language (**XML**)
a general-purpose *specification* for creating custom markup languages

Semantic web

The **Semantic Web** provides a common framework that allows data and knowledge to be shared and reused across application, enterprise, and community boundaries.

The development of the Sematic web is a collaborative effort led by **W3C** with participation from a large number of researchers and industrial partners.

It defines standards for exchanging knowledge and for sharing conceptualizations.

Basic standards:

- **RDF** - Resource Description Framework, representation of information/data for the purpose of sharing
  - **Based on XML - Extensible Markup Language format -** a general-purpose *specification* for building custom markup languages
- **OWL** – a language for sharing vocabularies, sets of terms supporting web searches and other applications (a part of RDF)

Semantic web

In terms of the knowledge representation and reasoning **SW** lets us:

- Represent the knowledge
- Support search queries on knowledge and matches
- Support inference

Differences from other KR systems:

- Multiple sources of information and knowledge built for potentially different purposes
- Ambiguities may arise (the same term with two different meanings or two different terms with the same meaning)
- Dynamically changing environment – knowledge is added at fast pace so it should be robust to handle that

Semantic web

Benefits:

- knowledge integration,
- knowledge storage,
- knowledge searching,
- and knowledge inference.

Semantic web

Benefits:

- knowledge integration,
- knowledge storage,
- knowledge searching,
- and knowledge inference.

Semantic web: knowledge integration

Benefit of large amounts of information and knowledge on the web stands and falls on the data/knowledge integration

Technical challenges:

- ***Location*:** where the data/knowledge resides. The *location* of a Semantic Web resource is defined by the **Uniform Resource Identifier (URI).** A URI is simply a formatted string that identifies - via name, location, or any other characteristic - a resource. A standard web link is a form of a URI. URI allows us to label a Semantic Web source with a *findable, unique* location.
- ***Query Protocol*:** We need to interact with web resources. We need a communication language. The protocol for the Semantic Web uses standards such as ***http*** to form a flexible, easily understood, request/response exchange.
- ***Format*:** The data must be in a comprehensive and translatable format. The Semantic Web uses a standard format - the **OWL Web Ontology Language.** It is based on the **Resource Description Framework (RDF)** standard and **Extensible Markup Language (XML).**

Technical challenges are resolved by standards

Semantic web: knowledge integration

Other Challenges

- ***Timely, Authoritative*:** The data must be trusted and be up-to-date. It is possible to have multiple answers to the same question. In addition, information may get outdated. The Semantic Web lets you to deal directly with the actual source to avoid the problem. You need not maintain complex synchronization unless it is absolutely necessary due to performance or other requirements.

The key challenge:

- ***Purpose*:** We have to align the data with our purpose. This may require translation and modifications. It needs to fit your world view be it English, medical, financial to name but a few. This is about getting right the semantic. For example, we can tie a *person* in one data source with an *individual* from another data source - they represent the same meaning or a related meaning.
- Semantic web standards do enable easier and more efficient data sharing and integration but really reach their full potential by **the ability to align purpose across different data sources**.

Semantic web: knowledge integration

Three steps of integration:

- ***Aggregation*:**
  - Combines the Semantic Web data sources into one unified, virtual data source.
- ***Mapping*/Binding:**
  - Associates similar references with each other and builds upon data in existing references. For example synonyms are identified.
- ***Rules*:**
  - Enables more sophisticated alignment and enrichment such as conditional logic that adds information based on the condition of other data

Semantic web: knowledge integration

Example: from Ivan Herman

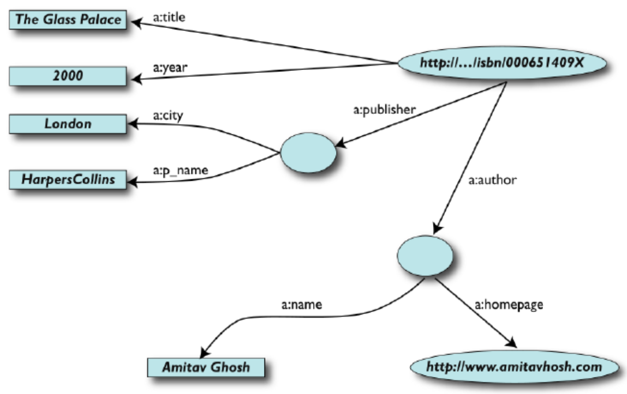
ID	Author	Title	Publisher	Year
ISBN 0-00-651409-X	id_xyz	The Glass Palace	id_qpr	2000

ID	Name	Home page
id_xyz	Amitav Ghosh	http://www.amitavghosh.com/

ID	Publisher Name	City
id_qpr	Harper Collins	London

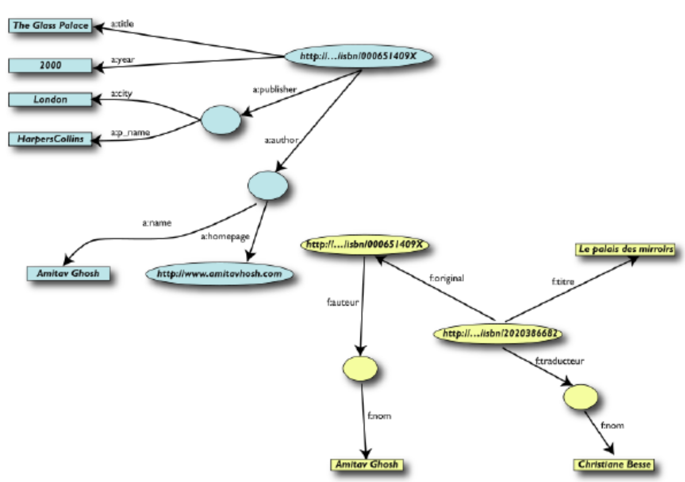
Semantic web: knowledge integration

Example:



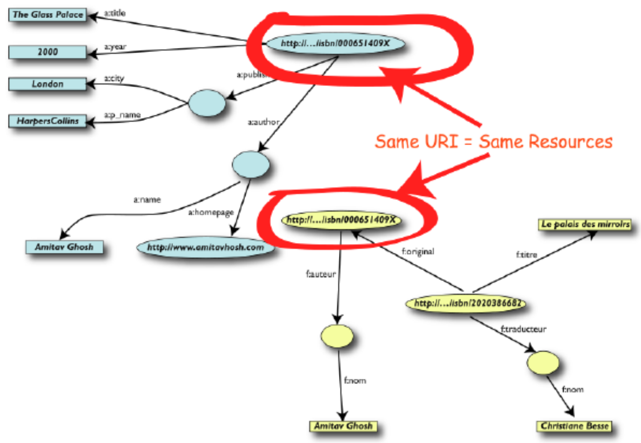
Semantic web: knowledge integration

Example: add data from another publisher



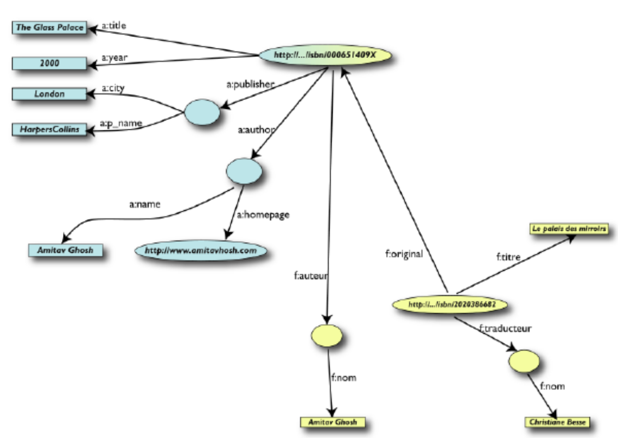
Semantic web: knowledge integration

Example:



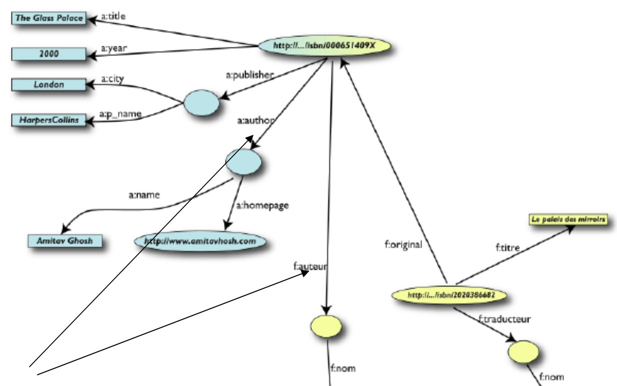
Semantic web: knowledge integration

Example: integration



Semantic web: knowledge integration

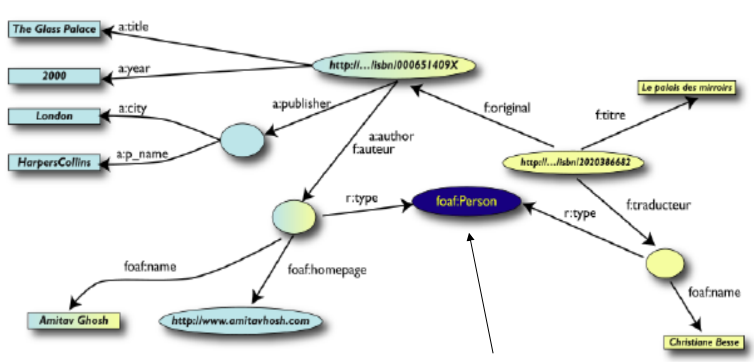
Example:



The same

Semantic web: knowledge integration

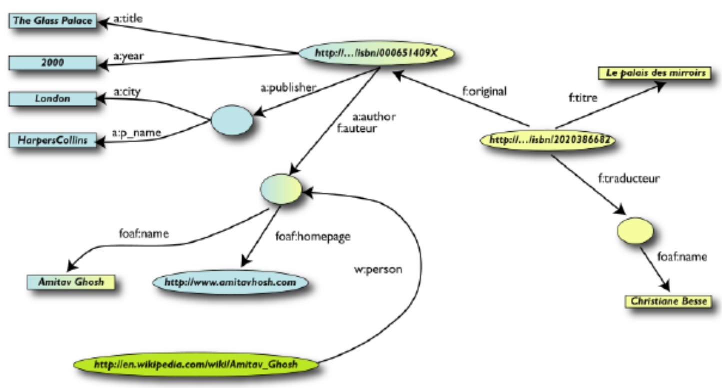
Example:



Add ontology

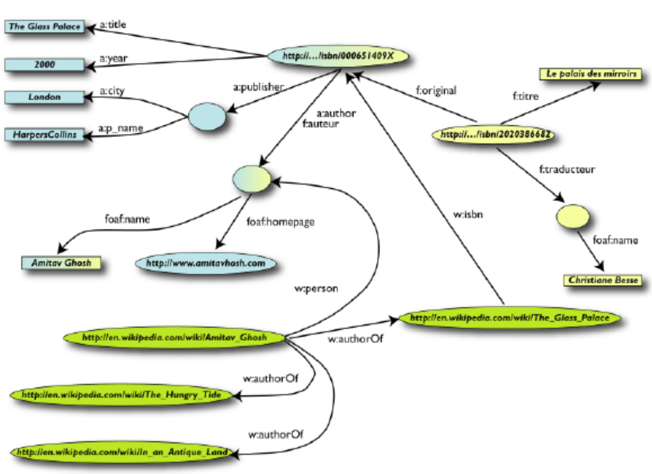
Semantic web: knowledge integration

Example: adding more knowledge



Semantic web: knowledge integration

Example: and go on



Semantic web: OWL

The Semantic Web is always written in the same language:

- **The OWL Web Ontology Language (http://www.w3.org/TR/owl-ref/)**

The Web Ontology Language OWL is:

- **a semantic markup language** for publishing and sharing **ontologies** on the World Wide Web.
- a **vocabulary extension of RDF** (the Resource Description Framework)

OWL contains all the reference information to **define any term contained within**

- it maintains its own definition of each and every term (it is self-referential).

RDF

Resource Description Framework (RDF)

- a data model that lets us make statements about Web resources in the form of subject-predicate-object sentences, called *triples:*
- The subject denotes the resource,
- the predicate expresses a subject-object relationship

Example: "The sky has the color blue"

- “a sky” is a subject
- "has the color“ is a predicate
- "blue“ is an object

RDF is an abstract model with several serialization formats (i.e., file formats): **typically XML**

Ontology

If more than one person is building a knowledge base, they must be able to share the **conceptualization**.

- A conceptualization is a mapping from the problem domain into the representation.
- A conceptualization specifies:
  – What types of objects are being modeled
  – The vocabulary for specifying objects, relations and properties
  – The meaning or intention of the relations or properties
- **An ontology is a specification of a conceptualization.**

Semantic web: OWL

```
- <owl:Ontology rdf:about="">
  <rdfs:comment>This is a weather forecast ontology.</rdfs:comment>
  <rdfs:label>Weather Site Ontology</rdfs:label>
  </owl:Ontology>
<!-- Weather Observation Class   -->
- <owl:Class rdf:ID="WeatherObservation">
  <rdfs:label>Weather Observation</rdfs:label>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasLocation" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasTime" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
...
```

Ontology definition

Class definition

Class properties

Semantic web: OWL

```
…
<rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasTemperature" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasHumidity" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasWindSpeed" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Class properties

Semantic web: OWL

```
<!-- Location Class  -->
- <owl:Class rdf:ID="Location">
  <rdfs:label>Location: City, State</rdfs:label>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasState" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasCity" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
  </owl:Class>
- <owl:DatatypeProperty rdf:ID="hasState">
  <rdfs:label>The State that this location is in. Abbreviated.</rdfs:label>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
```

Another class definition

Semantic web: OWL

```
<!-- Location Class -->
- <owl:Class rdf:ID="Location">
  <rdfs:label>Location: City, State</rdfs:label>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasState" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasCity" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
  </owl:Class>
- <owl:DatatypeProperty rdf:ID="hasState">
  <rdfs:label>The State that this location is in. Abbreviated.</rdfs:label>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
```

Another class definition

hasState property datatype

Semantic web: OWL

```
<!-- Precipitation Class -->
- <owl:Class rdf:ID="Precipitation">
  <rdfs:label>Precipitation Condition</rdfs:label>
- <owl:oneOf rdf:parseType="Collection">
  <Precipitation rdf:about="#Snow" />
  <Precipitation rdf:about="#Rain" />
  <Precipitation rdf:about="#Thunderstorm" />
  <Precipitation rdf:about="#None" />
  </owl:oneOf>
  </owl:Class>
```

Semantic web: OWL

Properties for Weather Observation class

```
<!-- Properties  -->
- <owl:ObjectProperty rdf:ID="hasLocation">
  <rdfs:label>Location of observation.</rdfs:label>
  <rdfs:range rdf:resource="#Location" />
  </owl:ObjectProperty>
- <owl:DatatypeProperty rdf:ID="hasTime">
  <rdfs:label>Date and time of observation.</rdfs:label>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="hasTemperature">
  <rdfs:label>Temperature, farenheit</rdfs:label>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
  </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="hasHumidity">
  <rdfs:label>Relative humidity, percent.</rdfs:label>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float" />
  </owl:DatatypeProperty>
```


Semantic web: OWL

Example: Aggregating knowledge from multiple ontologies

```
<owl:Ontology rdf:about="">
 <rdfs:comment>This is the project assignment client ontology</rdfs:comment>
 <rdfs:label>Project Assignment Client Ontology</rdfs:label>
 <owl:imports rdf:resource="http://localhost/contractors/ont/contractor-ont.owl"
   />
 <owl:imports rdf:resource="http://localhost/weather/ont/weather-ont.owl" />
 <owl:imports rdf:resource="http://localhost/projectsite/ont/project-ont.owl" />
 </owl:Ontology>
```

New ‘Project assignment’ ontology
Uses 3 ontologies: weather, project, contractor

Semantic web: OWL

```
<!-- Weather   -->
- <owl:Class rdf:ID="CurrentWeather">
  <rdfs:subClassOf rdf:resource="http://localhost/weather/ont/weather-
     ont.owl#WeatherObservation" />
  <owl:equivalentClass rdf:resource="http://localhost/weather/ont/weather-
     ont.owl#WeatherObservation" />
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasTemperature" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#forCity" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
…
</owl:Class>
```

Class Current Weather in the new Ontology: equivalent class Weather Observation

Semantic web: OWL

```
<!-- Weather   -->
- <owl:Class rdf:ID="CurrentWeather">
  <rdfs:subClassOf rdf:resource="http://localhost/weather/ont/weather-
     ont.owl#WeatherObservation" />
  <owl:equivalentClass rdf:resource="http://localhost/weather/ont/weather-
     ont.owl#WeatherObservation" />
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#hasTemperature" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
- <rdfs:subClassOf>
- <owl:Restriction>
  <owl:onProperty rdf:resource="#forCity" />
  <owl:cardinality>1</owl:cardinality>
  </owl:Restriction>
  </rdfs:subClassOf>
…
</owl:Class>
```

Different properties as used for the weather observation class before. Temperature, for state, for city, is hot, is dry etc.

Semantic web: OWL

```
<owl:DatatypeProperty rdf:ID="hasTemperature">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  <owl:equivalentProperty rdf:resource="http://localhost/weather/ont/weather-
     ont.owl#hasTemperature" />
  </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="forCity">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="forState">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="isWarm">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean" />
  </owl:DatatypeProperty>
- <owl:DatatypeProperty rdf:ID="isDry">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#boolean" />
  </owl:DatatypeProperty>
```


Semantic web: aggregation of sources

- **OWL** contains all the reference information to **define any term contained within** - it maintains its own definition of each and every term (it is self-referential).
- **Consequence:** ***Aggregation of multiple sources is easy***.
  – *We can simply add any OWL data to each other - in any combination or order* . Unlike relational databases, the structure (i.e. schema) or ontology is just another set of statements within a Semantic Web data source. You can simply combine multiple OWL sources together. You cannot just pour relational database data into another database without significant work behind the scenes with the databases schemas to clean up conflicts and the like.

With OWL, you can simply query the knowledge structure the same way you query any instance data. An OWL query doesn't differentiate between the structure and the instance data.

Semantic web: mapping/binding

- More than one ontology may exist
- The same term may have multiple entries but it really can mean the same thing at the end
- Mapping allows us to accomplish two unifying actions;
  - declaring synonyms and
  - establishing relationships.

Synonyms: we can declare the two terms used in two different resources to be the same.

Relations: inheritance relations among terms can be defined

Semantic web: integration with rules

Rules enables more complex knowledge aggregation methods.

We can use if/then constructs to establish relationships and groupings.

Rules can also add flexibility to your integration by handling special cases.

- **Example:** the *weather ontology* contains temperatures whereas the *project ontology* contains broader classifications such as hot and cold. We can establish a rule to convert certain temperatures to the correct hot or cold classification.

Rules can exist alongside with OWL as part of a the ***knowledge base***, or reside in the programs that manipulate the Semantic Web.