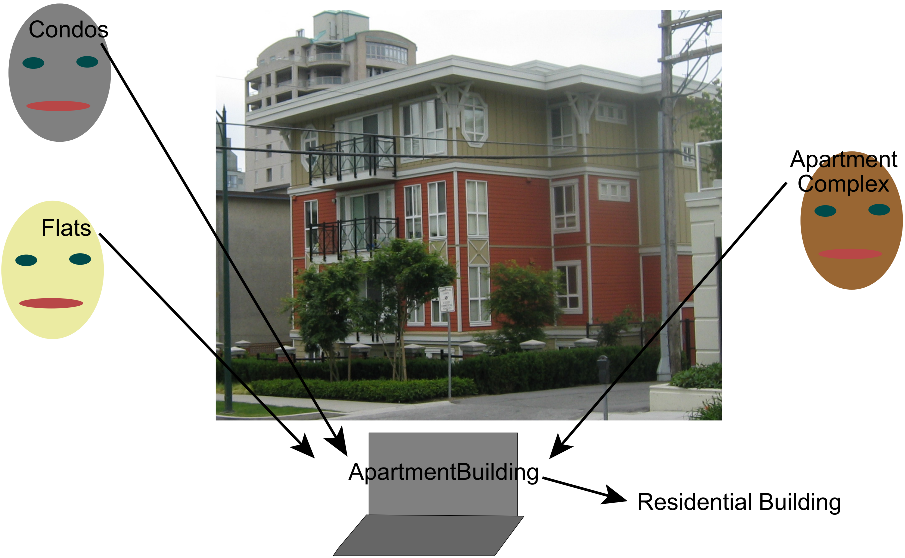
Knowledge Sharing

- If more than one person is building a knowledge base, they must be able to share the conceptualization.
- A conceptualization is a map from the problem domain into the representation. A conceptualization specifies:
  - What sorts of objects are being modeled
  - The vocabulary for specifying objects, relations and properties
  - The meaning or intention of the vocabulary
- An ontology is a specification of a conceptualization.
  An ontology specifies the meanings of the symbols in an information system.

Mapping from a conceptualization to a symbol



Semantic Web

- Ontologies are published on the web in machine readable form and are publicly readable.
- Builders of knowledge bases or web sites adhere to and refer to a published ontology:
  - the same symbol means the same thing across the various web sites that obey the ontology.
  - if someone wants to refer to some other object or relation, they publish the terminology with its intended interpretation. Others adopt the new terminology by using it and referring to its source. In this way, ontologies grow.
  - Separately developed ontologies can have mappings between them published.

Challenges of building ontologies

- They can be huge: finding the appropriate terminology for a concept may be difficult.
- How one divides the world can depend on the application. Different ontologies describe the world in different ways.
- People can fundamentally disagree about the appropriate structure.
- Different knowledge bases can use different ontologies.
- To allow KBs based on different ontologies to inter-operate, there must be mapping between different ontologies.
- It has to be in user's interests to use an ontology.
- The computer doesn't understand the meaning of the symbols. The formalism can constrain the meaning, but can't define it.

Semantic Web Technologies

- **XML** the Extensible Markup Language provides generic syntax.
  $\langle tag \ldots / \rangle$ or
  $\langle tag \ldots \rangle \ldots \langle / tag \rangle$.
- **URI** a Uniform Resource Identifier is a name of an object (resource). This name can be shared. Often in the form of a URL to ensure uniqueness.
- **RDF** the Resource Description Framework is a language of triples
- **OWL** the Web Ontology Language, defines some primitive properties that can be used to define terminology. (Doesn't define a syntax).

Main Components of an Ontology

- **Individuals** the objects in the world (not usually specified as part of the ontology)
- **Classes** sets of individuals
- **Properties** between individuals and their values

Individuals

- Individuals are things in the world that can be named. (Concrete, abstract, concepts, reified).
- Unique names assumption (UNA): different names refer to different individuals.
- The UNA is not an assumption you can universally make: "The Queen", "Elizabeth Windsor", etc.
- Without the determining equality, you can't count!
- In OWL you can specify:

  i_1 *SameIndividual* i_2.

  i_1 *DifferentIndividuals* i_3.

Classes

- A class is a set of individuals. E.g., house, building, officeBuilding
- One class can be a subclass of another

  house subClassOf building.

  officeBuilding subClassOf building.
- The most general class is *Thing*.
- Classes can be declared to be the same or to be disjoint:

  house EquivalentClasses singleFamilyDwelling.

  house DisjointClasses officeBuilding.
- Different classes are not necessarily disjoint.
  E.g., a building can be both a commercial building and a residential building.

Properties

- A property is between an individual and a value.
- A property has a domain and a range.

  livesIn domain person.

  livesIn range placeOfResidence.
- An *ObjectProperty* is a property whose range is an individual.
- A *DatatypeProperty* is one whose range isn't an object, e.g., is a number or string.
- There can also be property hierarchies:

  livesIn subPropertyOf enclosure.

  principalResidence subPropertyOf livesIn.

Properties (Cont.)

- One property can be inverse of another

  livesIn InverseObjectProperties hasResident.

- Properties can be declared to be transitive, symmetric, functional, or inverse-functional. (Which of these are only applicable to object properties?)
- You can also state the minimum and maximal cardinality of a property.

  principalResidence minCardinality 1.

  principalResidence maxCardinality 1.

Property and Class Restrictions

- You can define complex descriptions of classes in terms of restrictions of other classes and properties.
  E.g., A homeowner is a person who owns a house.

  $$homeOwner \;\; subClassOf \;\; person.$$

  $$homeOwner \;\; subClassOf$$
  $$ObjectSomeValuesFrom(owns, house).$$

OWL Class Constructors

owl:Thing $\equiv$ all individuals
owl:Nothing $\equiv$ no individuals
owl:ObjectIntersectionOf$(C_1, \ldots, C_k) \equiv C_1 \cap \cdots \cap C_k$
owl:ObjectUnionOf$(C_1, \ldots, C_k) \equiv C_1 \cup \cdots \cup C_k$
owl:ObjectComplementOf$(C) \equiv \mathit{Thing} \setminus C$
owl:ObjectOneOf$(I_1, \ldots, I_k) \equiv \{I_1, \ldots, I_k\}$
owl:ObjectHasValue$(P, I) \equiv \{x : x\ P\ I\}$
owl:ObjectAllValuesFrom$(P, C) \equiv \{x : x\ P\ y \rightarrow y \in C\}$
owl:ObjectSomeValuesFrom$(P, C) \equiv \{x : \exists y \in C \text{ such that } x\ P\ y\}$
owl:ObjectMinCardinality$(n, P, C) \equiv \{x : \#\{y | xPy \text{ and } y \in C\} \geq n\}$
owl:ObjectMaxCardinality$(n, P, C) \equiv \{x : \#\{y | xPy \text{ and } y \in C\} \leq n\}$

OWL Predicates

rdf:type$(I, C) \equiv I \in C$
rdfs:subClassOf$(C_1, C_2) \equiv C_1 \subseteq C_2$
owl:EquivalentClasses$(C_1, C_2) \equiv C_1 \equiv C_2$
owl:DisjointClasses$(C_1, C_2) \equiv C_1 \cap C_2 = \{\}$
rdfs:domain$(P, C) \equiv$ if xPy then $x \in C$
rdfs:range$(P, C) \equiv$ if xPy then $y \in C$
rdfs:subPropertyOf$(P_1, P_2) \equiv xP_1y$ implies xP_2y
owl:EquivalentObjectProperties$(P_1, P_2) \equiv xP_1y$ if and only if xP_2y
owl:DisjointObjectProperties$(P_1, P_2) \equiv xP_1y$ implies not xP_2y
owl:InverseObjectProperties$(P_1, P_2) \equiv xP_1y$ if and only if yP_2x
owl:SameIndividual$(I_1, \ldots, I_n) \equiv \forall j \forall k\ I_j = I_k$
owl:DifferentIndividuals$(I_1, \ldots, I_n) \equiv \forall j \forall k\ j \neq k$ implies $I_j \neq I_k$
owl:FunctionalObjectProperty$(P) \equiv$ if xPy_1 and xPy_2 then $y_1 = y_2$
owl:InverseFunctionalObjectProperty$(P) \equiv$
if x_1Py and x_2Py then $x_1 = x_2$
owl:TransitiveObjectProperty$(P) \equiv$ if xPy and yPz then $y = z$
owl:SymmetricObjectProperty $\equiv$ if xPy then yPz

Knowledge Sharing

- One ontology typically imports and builds on other ontologies.
- You need facilities for version control.
- Tools for mapping one ontology to another to allow inter-operation of different knowledge bases.
- The semantic web promises to allow you to find the right concept in a query if
  - the information adheres to some ontology
  - the query adheres to some ontology
  - these are the same ontology or there is a mapping between them.

Example: Apartment Building

An apartment building is a residential building with more than two units and they are rented.

Example: Apartment Building

An apartment building is a residential building with more than two units and they are rented.

```
:numberOfUnits rdf:type    owl:FunctionalObjectProperty;
         rdfs:domain :ResidentialBuilding;
         rdfs:range  owl:OneOf(:one :two :moreThanTwo).
```

Example: Apartment Building

An apartment building is a residential building with more than two units and they are rented.

```
:numberOfUnits rdf:type    owl:FunctionalObjectProperty;
         rdfs:domain :ResidentialBuilding;
         rdfs:range  owl:OneOf(:one :two :moreThanTwo).

:ApartmentBuilding
   owl:EquivalentClasses
        owl:ObjectIntersectionOf (
             owl:ObjectHasValue(:numberOfUnits
                                :moreThanTwo)
             owl:ObjectHasValue(:onwership
                                :rental)
            :ResidentialBuilding).
```

Aristotelian definitions

Aristotle [350 B.C.] suggested the definition if a class C in terms of:

- Genus: the super-class
- Differentia: the attributes that make members of the class C different from other members of the super-class

"If genera are different and co-ordinate, their differentiae are themselves different in kind. Take as an instance the genus 'animal' and the genus 'knowledge'. 'With feet', 'two-footed', 'winged', 'aquatic', are differentiae of 'animal'; the species of knowledge are not distinguished by the same differentiae. One species of knowledge does not differ from another in being 'two-footed'."

Aristotle, *Categories*, 350 B.C.

Basic Formal Ontology (BFO)

- entity
  - continuant
    - independent continuant
      - site
      - object aggregate
      - object
      - fiat part of object
      - boundary of object
    - dependent continuant
      - realizable entity
        - function
        - role
        - disposition
      - quality
    - spatial region
      - volume / surface / line / point

BFO (cont.)

- occurrent
  - temporal region
    - connected temporal region
      - temporal interval
      - temporal instant
    - scattered temporal region
  - spatio-temporal region
    - connected spatio-temporal region
      - spatio-temporal interval / spatio-temporal instant
    - scattered spatio-temporal region
  - processual entity
    - process
    - process aggregate
    - processual context
    - fiat part of process
    - boundary of process