UNIVERSITY OF SAN FRANCISCO

CHANGE THE WORLD FROM HERE

AI – Advanced Knowledge Representation

Cindi Thompson

The problem with logic…

- Logic is useful to reason about arbitrary scenarios
- To reason about arbitrary scenarios you need to be able to express arbitrary scenarios
- Logics that can express arbitrary scenarios have very difficult (impossible) inference problems
- To reason with a logic you need to solve its decision problem...

The more useful a logic becomes, the harder to use it becomes!

Modern research in logic in AI is a constant trade-off between expressivity and complexity.

Temporal & Epistemic Logic

- One successful approach to applying logic in AI has been to tailor the logic to a fixed domain.
- Two common domains are *Time* (temporal logic) and *Knowledge* (epistemic logic).
- Temporal and epistemic logic extend **propositional logic**
- Temporal logic adds operators for “at the next moment of time”, and “at some future moment of time”.
- For each agent *a*, epistemic logic adds an operator for “agent *a* knows”

Another approach: Categories and Objects

- The organization of objects into categories is a vital part of Knowledge Representation (KR)
- Important relationships are
  - subclass relation (AKO - a kind of)

    <sub-category> AKO <category>
  - instance relation (ISA - is a)

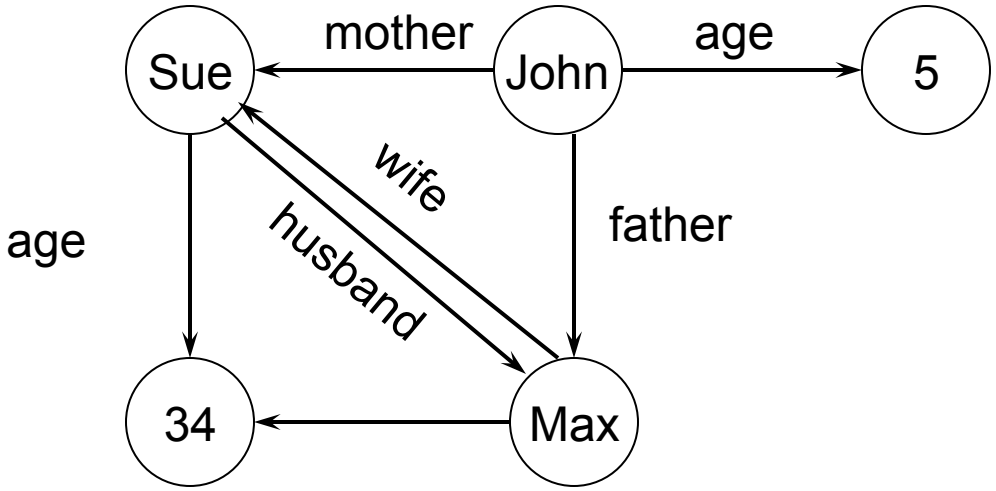
    <object> ISA <category>

Semantic Networks

- First introduced by Quillian back in the late-60s
  - M. Ross Quillian. "Semantic Memories", In M. M. Minsky, editor, Semantic Information Processing, pages 216-270. Cambridge, MA: MIT Press, 1968
- A simple representation scheme which uses a graph of labeled nodes and labeled directed arcs to encode knowledge
  - Nodes – objects, concepts, events
  - Arcs – relationships between nodes
- **Graphical depiction** associated with semantic networks is a big reason for their popularity

Nodes and Arcs

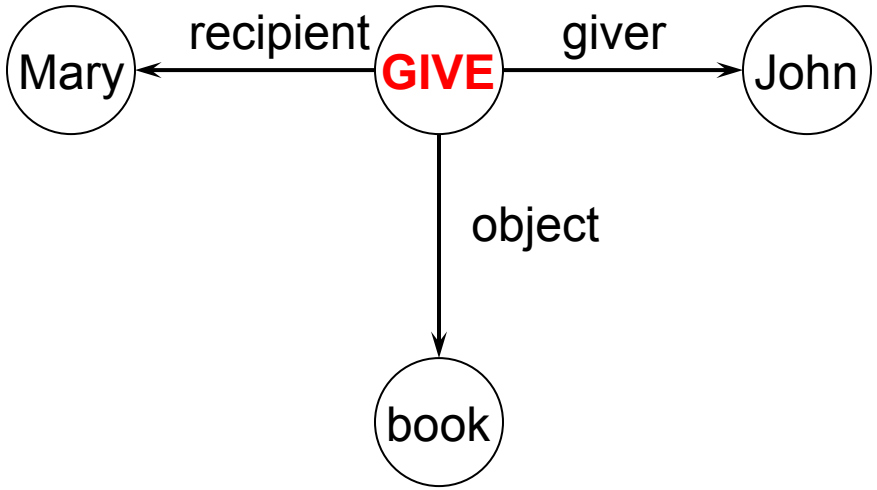
Arcs define binary relations that hold between objects denoted by the nodes



mother (john, sue)
age (john, 5)
wife (sue, max)
age (max, 34)
…

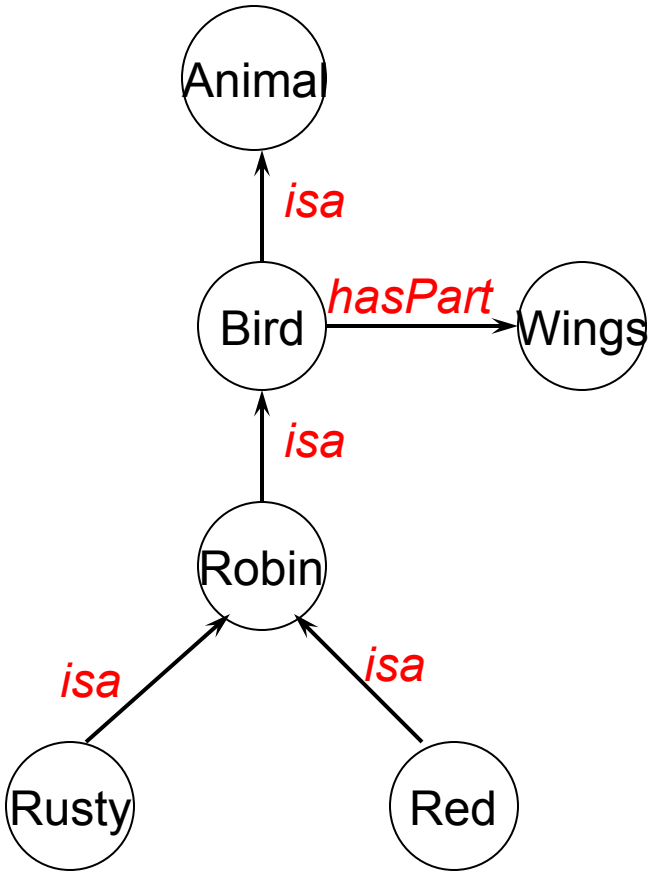
Beyond binary relations

- We can represent the generic *give* event as a relation involving three things:
  - A giver
  - A recipient
  - An object



Inheritance

- The main kind of reasoning done in semantic nets
- Recall **ISA** that can also link a class and its superclass.
- Some links (e.g. **haspart**) are inherited along **ISA** paths
- The semantics of a semantic net can be relatively informal or very formal
  - Often defined at the implementation level



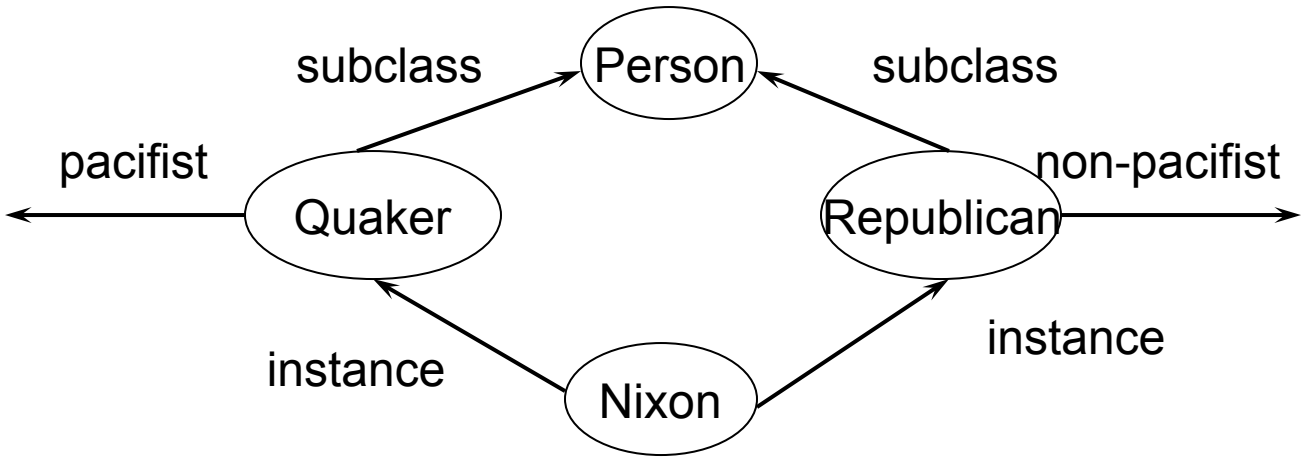
Multiple Inheritance

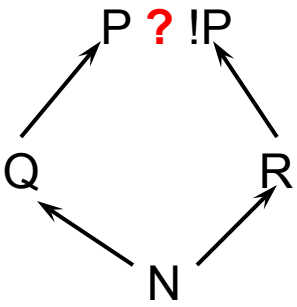
A node can have any number of superclasses that contain it

- This enables a node to inherit properties from multiple parent nodes and their ancestors in the network.
- It can cause conflicting inheritance.

Nixon Diamond

(two contradictory inferences from the same data)





Advantages of Semantic nets

- Easy to visualize
- Formal definitions have been developed
- Related knowledge is easily clustered
- Efficient space requirements
  - Objects represented only once
  - Relationships handled by pointers

Disadvantages of Semantic nets

- Inheritance (particularly from multiple sources and when exceptions in inheritance are wanted) can cause problems.
- Facts placed inappropriately cause problems.
- No standards for node and arc values

Frames

- Semantic net with properties
- Represents an entity as a set of slots (attributes) and associated values
- Can represent a specific entry, or a general concept
- Frames are implicitly associated with one another because the value of a slot can be another frame

3 components of a frame

- frame name
- attributes (slots)
- values (fillers: list of values, range, string, etc.)

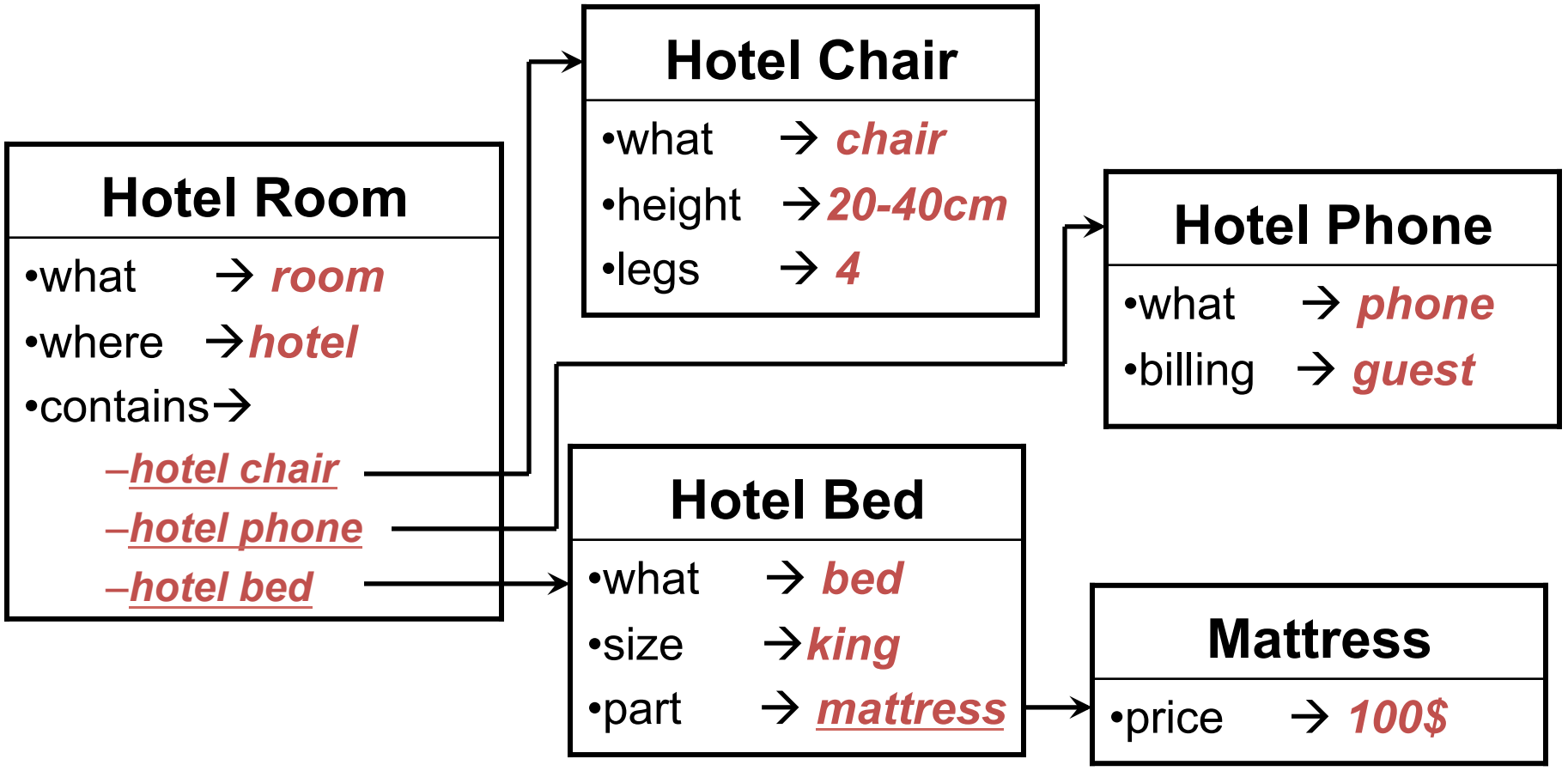
Book Frame
Slot → ***Filler***
• Title → ***AI A modern Approach***
• Author → ***Russell & Norvig***
• Year → ***2003***

Features of Frame Representation

- More natural support of values then semantic nets (each slot has constraints describing legal values that a slot can take)
- Can be easily implemented using object-oriented programming techniques
- Inheritance is easily controlled

Inheritance

Similar to OOP paradigm



Advantages of Frames

- Makes programming easier by grouping related knowledge
- Easily understood by non-developers
- Expressive power
- Easy to set up slots for new properties and relations
- Easy to include default information and detect missing values

Disadvantages of Frames

- No standards (slot-filler values)
- More of a general methodology than a specific representation:
  - Frame for a classroom will be different for a professor and for a maintenance worker
- No associated reasoning/inference mechanisms

Description Logic (DL)

- There is a family of frame-like KR systems with a formal semantics
  - Examples: KL-ONE, Classic
- A subset of FOL designed to focus on categories and their definitions in terms of existing relations.
- More expressive than frames and semantic networks
- Major inference tasks:
  - Subsumption: Is category C1 a subset of C2?
  - Classification: Does Object O belong to C?

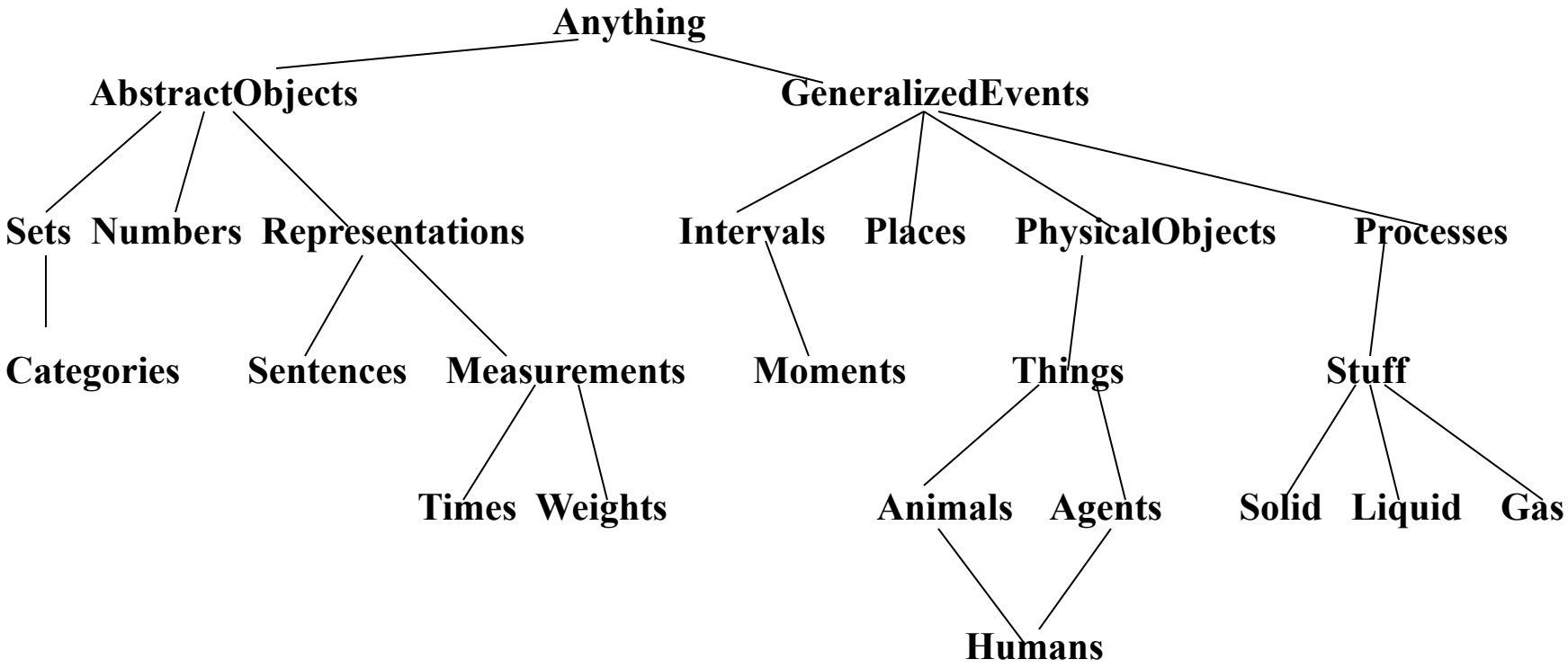
Ontologies

- Description Logics can represent Ontologies very effectively

Ontology means the study of existence, the things that are. In the field of philosophy this is a branch of metaphysics, but in information sciences the study of ontologies has come to mean the way we can describe concepts.

- Representing abstract concepts is referred to as ontological engineering or knowledge engineering.

Example Ontology



Applications of DL & Ontologies

- DL is a very natural representation for Knowledge bases, and
- Also conducive to reasoning techniques such as resolution and backwards chaining.

Applications of DL

- Information extraction
- Semantic Web
- Medical

Example of engineering an ontology

- Protégé ontology construction tool

Information Extraction and Ontologies

- Ontologies are increasingly being used in industry to support information extraction
- Reminder: creating structured information (KB) from unstructured text
  - Often focused on who did what to whom

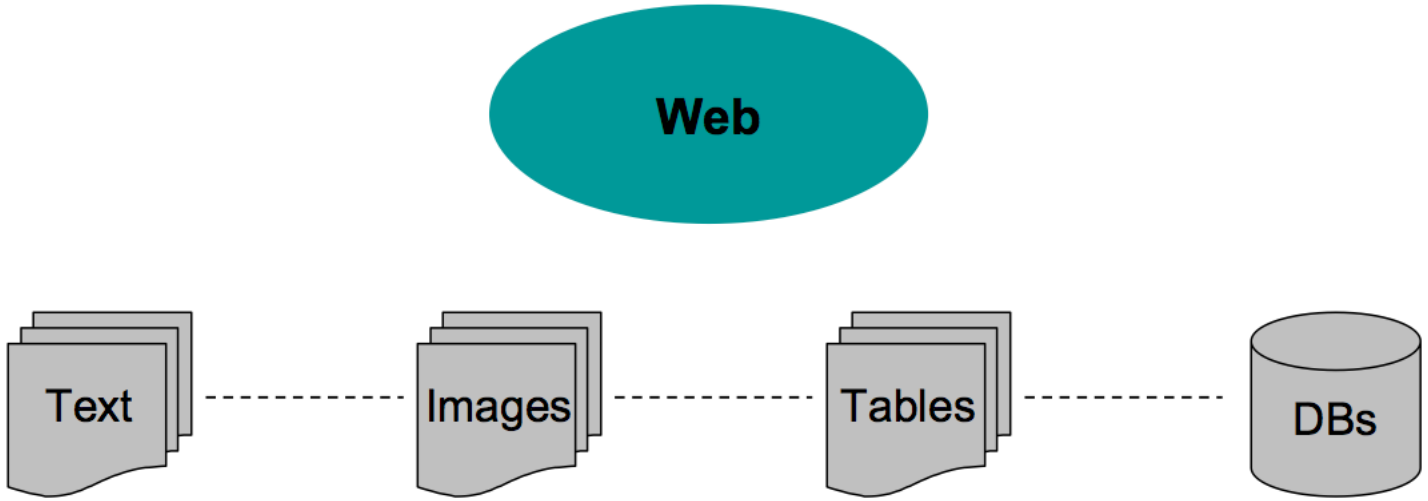
The entities defined in an ontology can be used to guide the IE process

Semantic Web

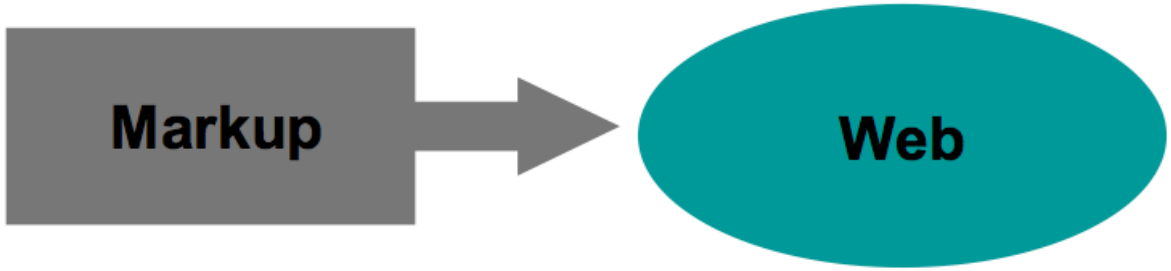
- Ambitious ongoing project attempting to associate *meaning* to concepts in webpages.
- Extension of “standard” html.
- Uses XML to incorporate meta-data into web-pages
- Allows for web-search and information synthesis based on more than just word frequencies and PageRank

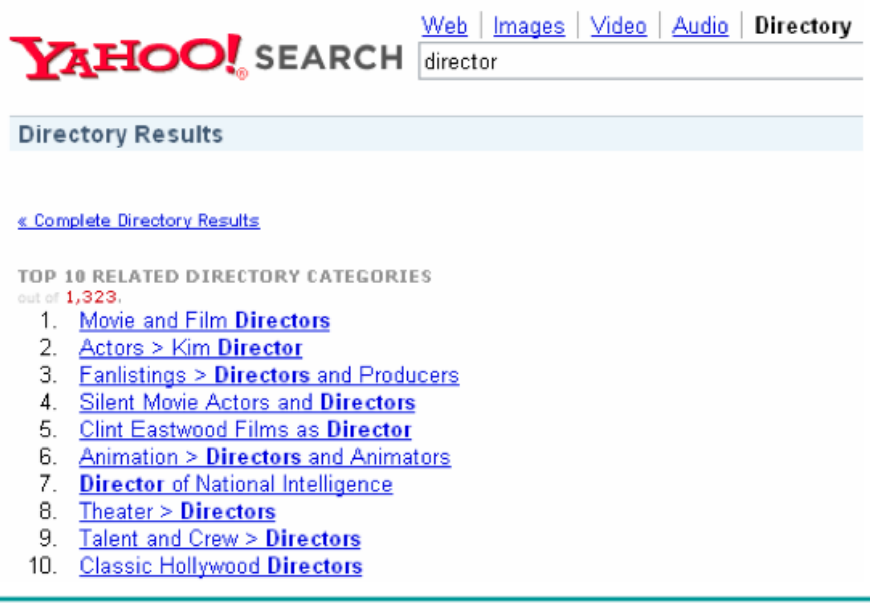
Semantic Web – how to get there….

Web is (mostly) non-interpreted data

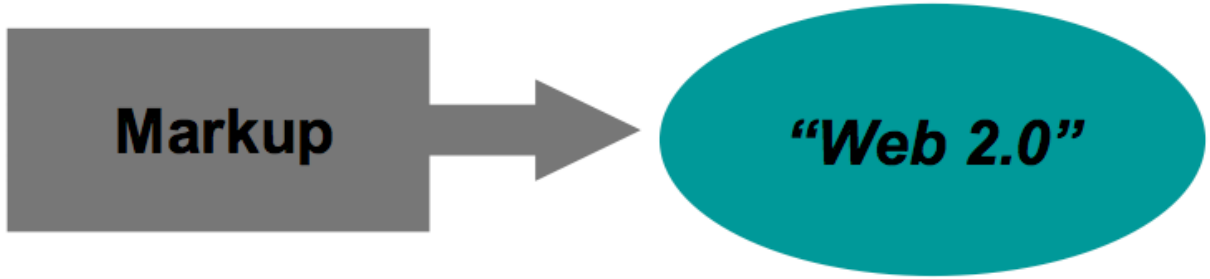


Interpretation through Markup – Categories





Interpretation through Markup – User Tags



Interpretation through Markup – User Tags

Formal Interpretation – Knowledge Markup

Formal Interpretation – Knowledge Markup

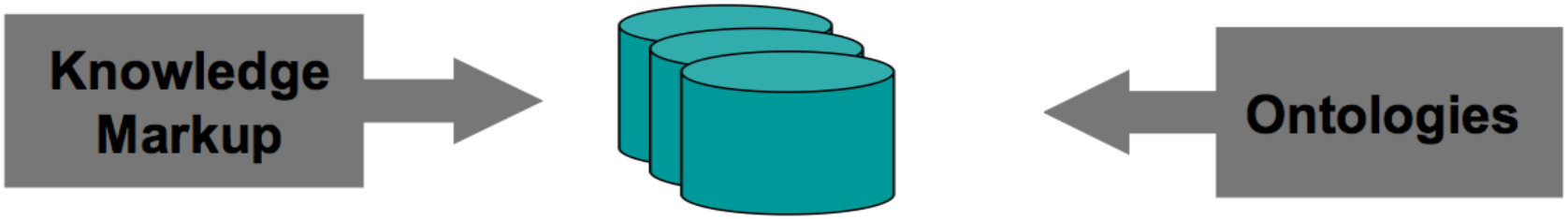

```
- <daml:Class rdf:ID="Director">
    <rdfs:label>Director</rdfs:label>
    <rdfs:comment />
  - <rdfs:subClassOf>
      <daml:Class rdf:about="#Role" />
    </rdfs:subClassOf>
  </daml:Class>
```

Formal Interpretation – Knowledge Markup


```
- <daml:Class rdf:ID="Director">
    <rdfs:label>Director</rdfs:label>
    <rdfs:comment />
  - <rdfs:subClassOf>
      <daml:Class rdf:about="#Role" />
    </rdfs:subClassOf>
  </daml:Class>
```

```
- <owl:Class rdf:ID="Director">
    <rdfs:label>director</rdfs:label>
  - <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Person" />
    - <owl:Restriction>
        <owl:onProperty rdf:resource="#headOf" />
      - <owl:someValuesFrom>
          <owl:Class rdf:about="#Program" />
        </owl:someValuesFrom>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
```

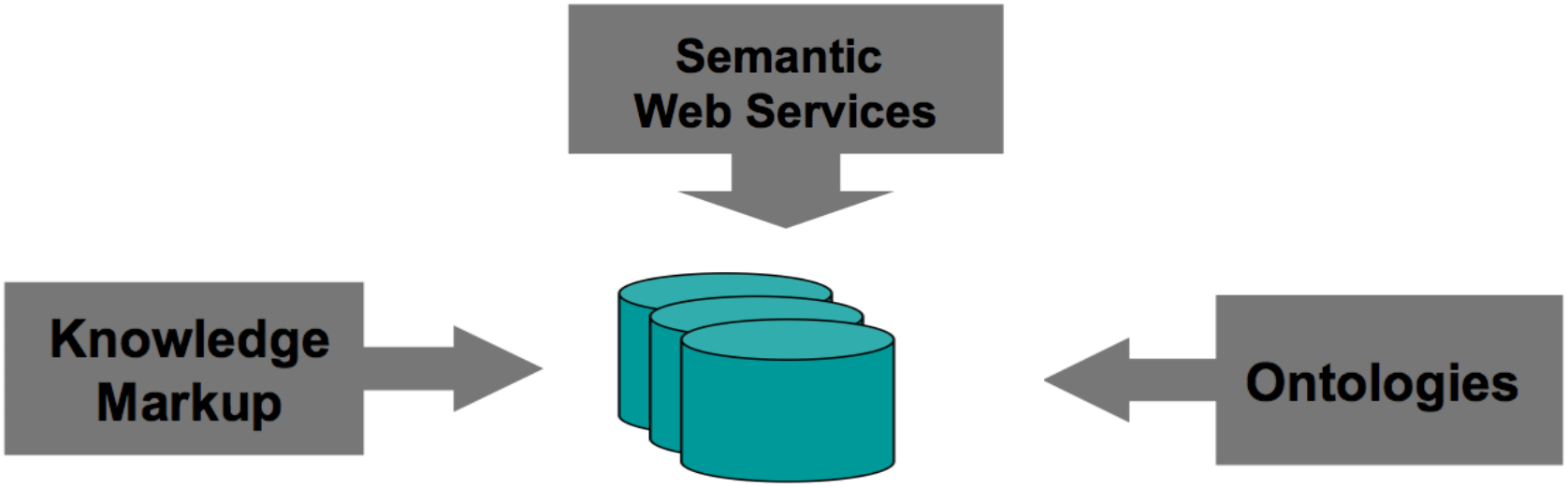
Turns the Web into a Knowledge Base




```
- <daml:Class rdf:ID="Director">
    <rdfs:label>Director</rdfs:label>
    <rdfs:comment />
  - <rdfs:subClassOf>
      <daml:Class rdf:about="#Role" />
    </rdfs:subClassOf>
  </daml:Class>
```

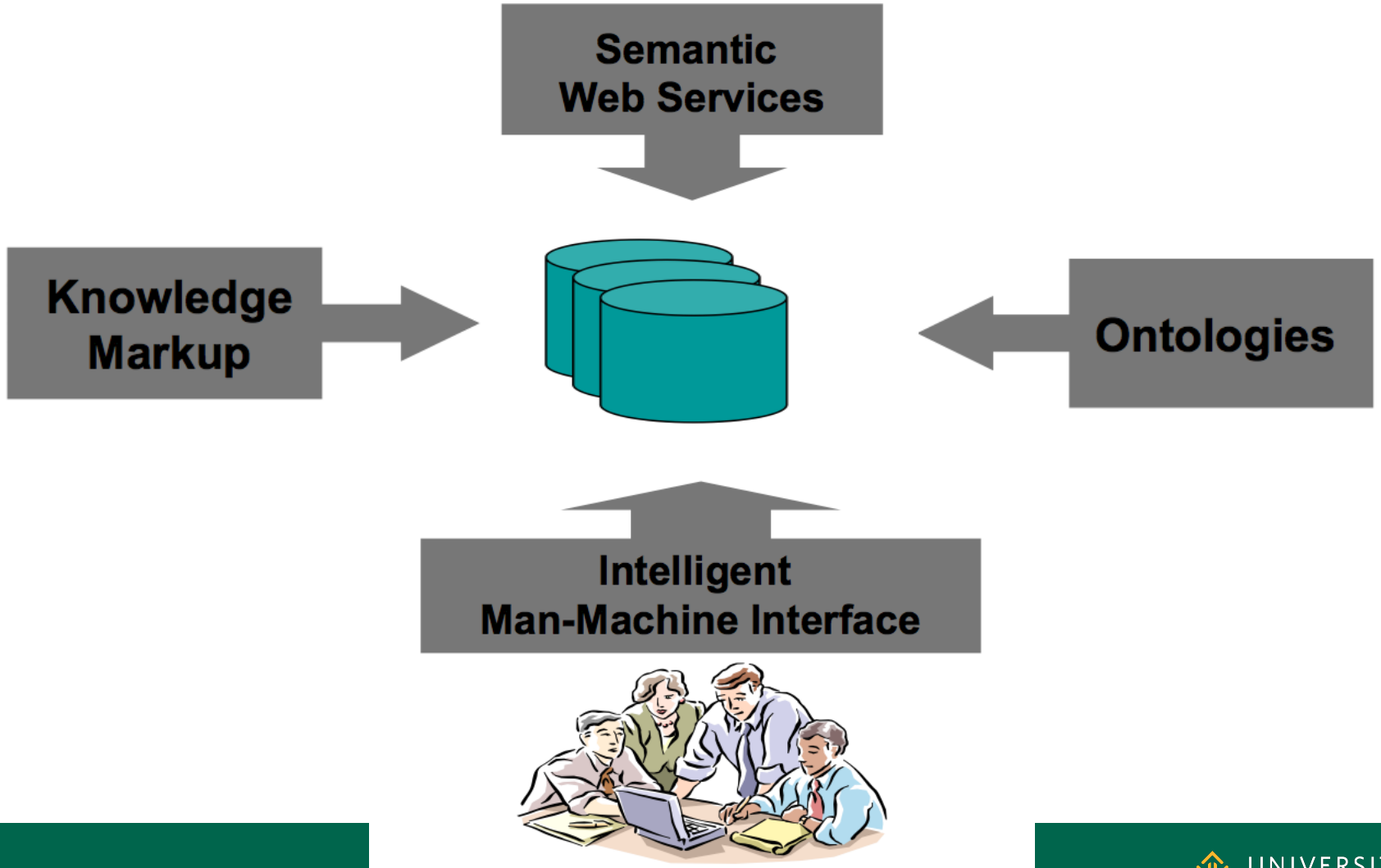
```
- <owl:Class rdf:ID="Director">
    <rdfs:label>director</rdfs:label>
  - <owl:intersectionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Person" />
    - <owl:Restriction>
        <owl:onProperty rdf:resource="#headOf" />
      - <owl:someValuesFrom>
          <owl:Class rdf:about="#Program" />
        </owl:someValuesFrom>
      </owl:Restriction>
    </owl:intersectionOf>
  </owl:Class>
```

Enables Semantic Web Services



Sometimes called Software as a Service (to simplify)

And Intelligent Human-Machine Interfaces



Components of the Semantic Web

- XML: the base layer as a standard for passing information
- RDF: a simple language for expressing data models
- **OWL**: an ontology language for describing relationships between classes, based on description logic
- SPARQL: a protocol and query language

Semantic Web Vision

- It is hoped that semantic web resources will enable intelligent agents to synthesize information and execute plans in the web domain.
- For example, imagine being able to deploy an agent to book you the cheapest tickets to New York for two weeks during July or August.

Medical Ontologies – SNoMed-CT

- *Systemized Nomenclature of Medical-Clinical Terms.*
- Definition of over 1 million medical terms and concepts
- Organized for automated searching and deduction.
- Concepts are either
  - primitive (eg. virus) or
  - defined using Description Logic (e.g. juvenile diabetes is the intersection of the disease diabetes and state of being a child).
- also contains information on drugs, biology, topography, occupations etc.

SNoMed-CT Example

Enables the following set of associations for tuberculosis pneumonia:

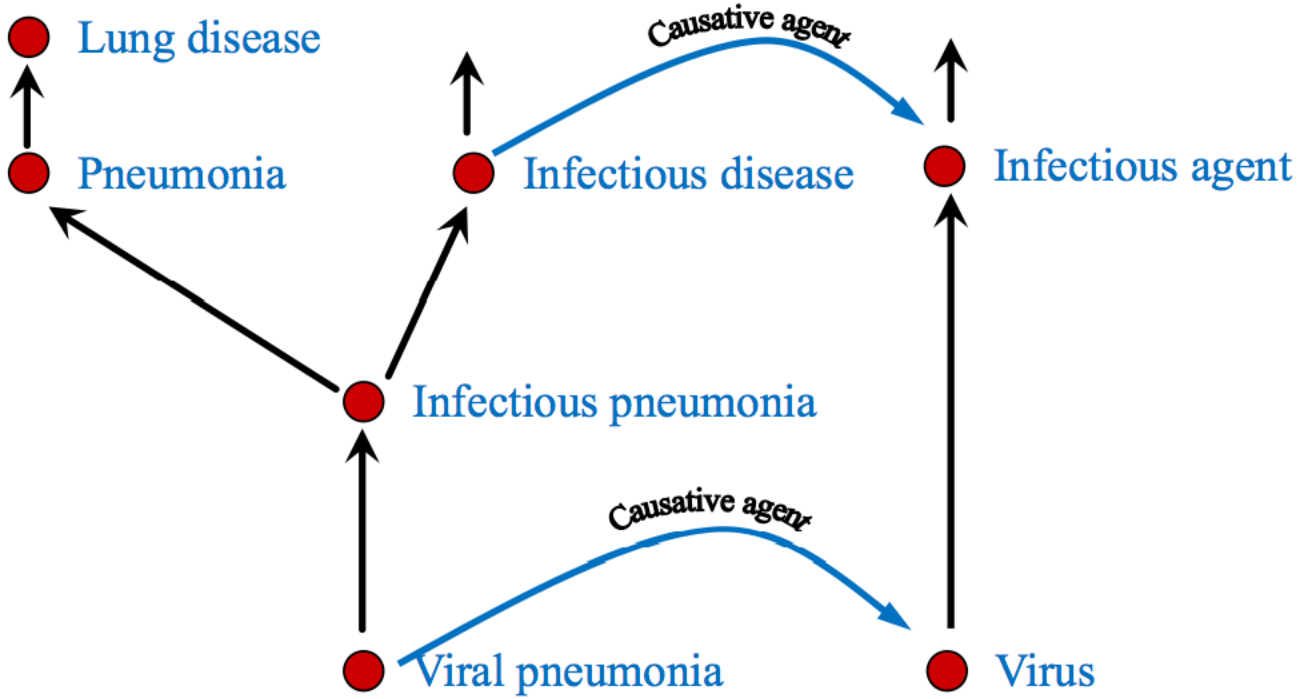
- A kind of lung infection
- A kind of pneumonia
- Caused by mycobacterium tuberculosis
- Site of infection: the lung

SNoMed-CT Example

Er s

pr

-
-
-
-



CYC

- A knowledge engineering effort
- Encoding of large amounts of knowledge about the everyday world
- 1984-present
- A person century of effort
- 10^6 general concepts and axioms

CYC – Example assertions

- You have to be awake to eat.
- You can usually see people's noses but not their hearts.
- Given two professions, either one is a specialization of the other or they are likely to be independent.
- You cannot remember events that have not happened yet.
- If you cut a lump of peanut butter in half, each half is also a lump of peanut butter; but if you cut a table in half, neither half is a table.

CYC – Example Contexts

Heart surgery

Total darkness

Fiction

Default context