Knowledge Engineering

Semester 2, 2004-05

Michael Rovatsos
mrovatso@inf.ed.ac.uk

Lecture 5 – Basics of Ontologies

25th January 2005

Where are we?

Last time . . .

- we attempted a transition from Knowledge Acquisition to Knowledge Representation

Focus of the KR&R part of the module . . .

- representation of complex domain knowledge
- ontology reasoning systems
- dealing with uncertainty

Today . . .

- basics of ontologies
- formalising certain kinds of knowledge

Ontologies

- In toy domains, easy to describe relevant objects and relationships to reason about
- In more complex domains, a principled way of structuring the domain of discourse is required
- **Ontology**
  - philosophically speaking: a theory of nature of being or existence
  - practically speaking: a formal specification of a shared conceptualisation

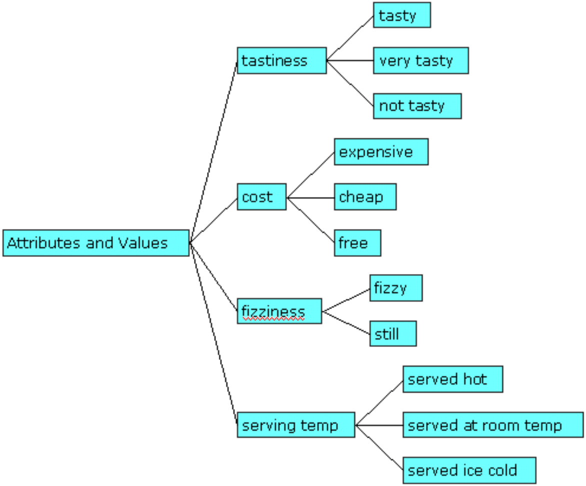
Ontologies

What are they good for?

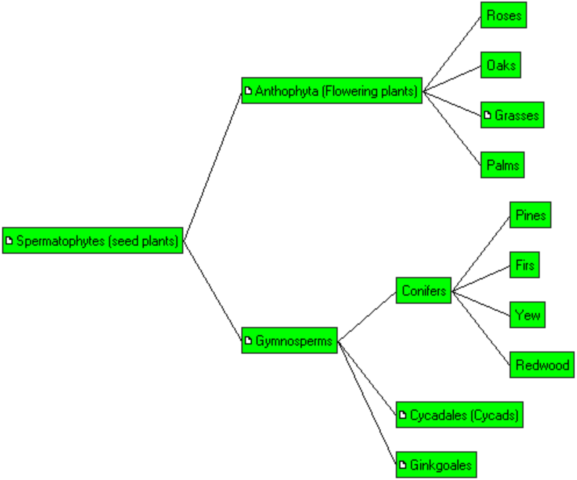
- Knowledge sharing and reuse (agreeing on a vocabulary)
- Support of use of **knowledge level** vs. symbolic level
- Make **ontological commitments** (decisions regarding conceptualisation which relfect points of view) explicit
- Interaction problem: choice of knowledge representation depends on problem to solve and inference mechanisms to be used

Many different representations, will use first-order logic (FOL) and discuss various knowledge modelling issues

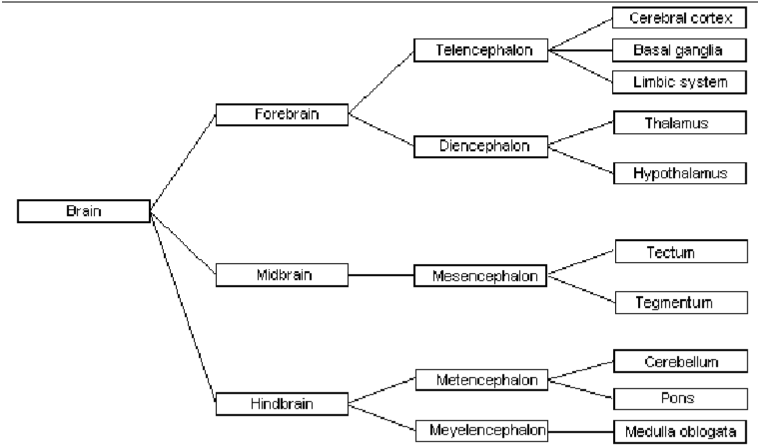
Example: Attribute Ladder



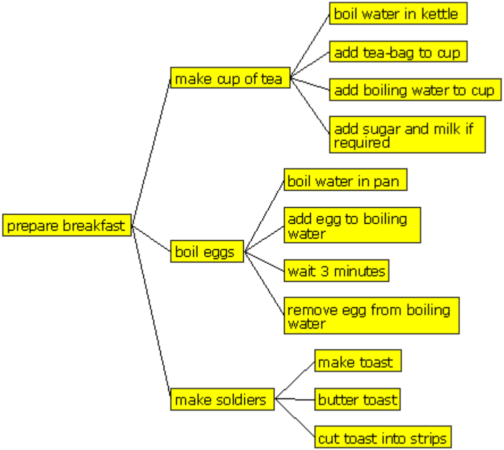
Example: Concept Tree



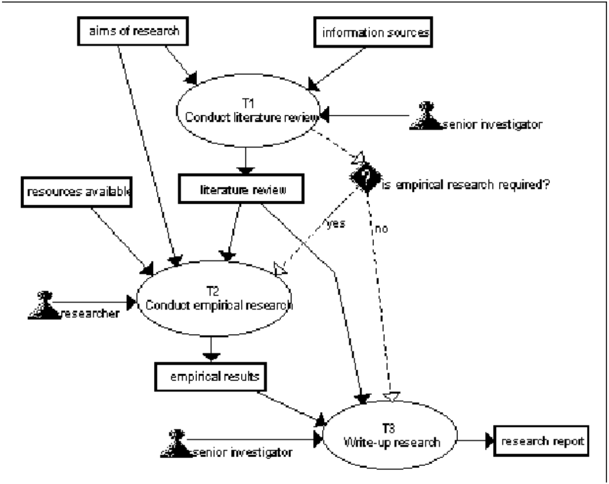
Example: Composition Ladder



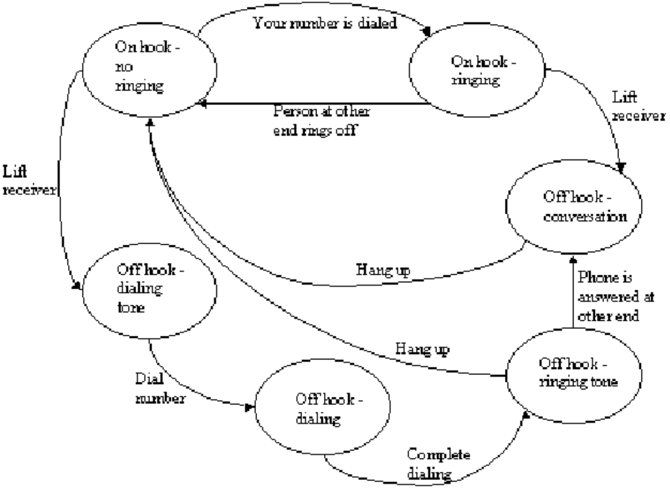
Example: Process Ladder



Example: Process Map

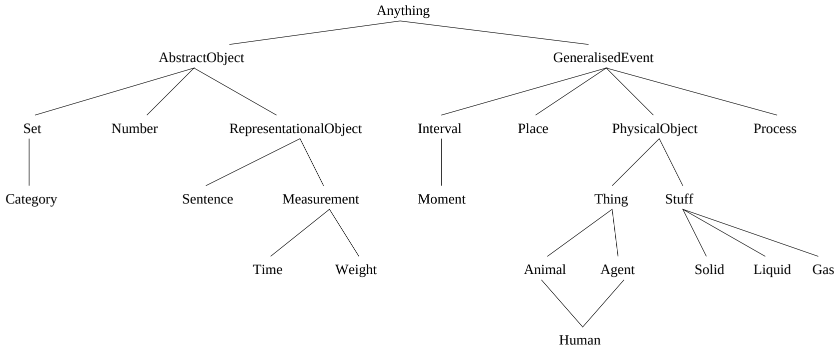


Example: State Diagram



Upper Ontologies

General framework of concepts (convention: from top to bottom more specific)

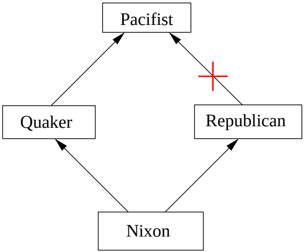


Categories

- Categories play an important role in reasoning (although individual objects are interacted with in practice)
- Representation through predicates ($Car(X)$) or through **reification** ($Member(X, Cars)$)
- One way of defining categories: category = a collection of its members
- **Inheritance** most common relationship between categories

Categories

- Subclasses inherit properties of super-classes (➡ OOP)
- **Taxonomy**: an ontology of categories induced by subclass relationships
- Problems of multiple inheritance
- Example: The Nixon diamond



Categories

- Can use FOL to express all kinds of properties of categories:
  - Subclasses: $Basset \subset Dog$, $Dog \subset Animal$
  - Describing properties/inferring class membership: $\forall x\ Basset(x) \Rightarrow GoodScent(x)$, $\forall x\ GoodScent(x) \Rightarrow Basset(x)$
  - Category properties: $Basset \in Species$
- Further common properties of categories:
  - Disjointness
  - Exhaustive decomposition
  - Partition
- Exercise: describe these in FOL

Physical Composition

- Want to express physical composition of objects
- part-of relation (reflexive,transitive), e.g. $PartOf(Leg, Body)$
- How do we express a collection of concrete objects, e.g. a bag of apples?
- Use of "set" problematic, since a set has no weight (is not an object itself)
- Define "bunch": $\forall x\; x \in s \Rightarrow PartOf(x, BunchOf(s))$
- Smallest object satisfying this condition (**logical minimisation**):

$$\forall y\; [\forall x\; x \in s \Rightarrow PartOf(x, y)] \Rightarrow PartOf(BunchOf(s), y)$$

Measurements

- **Quantitative** measurements: mass, price, weight etc.
  - $Price(MyBasset) = Pounds(500) = Euro(750)$
  - Abstract objects: $Pounds(500)$ is not a 500 pound amount of money/account balance
  - Each measurement value exists only once
- **Qualitative** measurements: focus on ordering btw. different values, not the values themselves
- Example: use of rule

  $$\forall x \forall y\, Vehicle(x) \wedge Vehicle(y) \wedge Faster(x, y) \Rightarrow Prefer(x, y)$$

  sufficient (KB contains facts $Faster(Car, Bicycle)$) rather than getting speed measurements for each type of vehicle
- Area of **qualitative physics**

Substances and Objects

- Intuition: specify objects in the world and put them together to obtain composite objects
- Problem of **individuation** (division into distinct object)
  - No problem for count nouns (cats, dogs, apples, planets)
  - But how about "stuff" (water, air, energy)?
- Example: Assume category *Water*
  - $x \in Water \wedge PartOf(x, y) \Rightarrow y \in Water$
  - $x \in Water \Rightarrow BoilingPoint(x, 100^o C)$
- But still problems: *SaltWater* subcategory of *Water* but how about *PintsOfWater*?
- Underlying problem: difference between **intrinsic** properties (properties of the substance, retained under subdivision) and **extrinsic** properties of objects

Expressing Change

- Straightforward way of capturing change: use time-steps t in all predicates, and express change by reasoning about subsequent time-steps:

$$\forall t \; Rains(t) \Rightarrow WetGround(t+1)$$

- Alternatively, concentrate on **situations** brought about by different actions ➡ **situation calculus**
- Situations are logical terms S_0, S_1, etc.
- Function $Result(a, s)$ used to name situation that results from executing action a in s
- Sometimes useful to extend this to sequences of actions

$$Result([a|rest], s) = Result(rest, Result(a, s))$$

Expressing Change

- **Fluents** = functions/predicates that vary from situation to situation (opposite: **atemporal**/**eternal** functions/predicates)
- Describe actions by possibility and effect axioms:
  - Possibility axiom: $Preconditions \Rightarrow Poss(a, s)$
  - Effect axiom:
    $Poss(a, s) \Rightarrow Changes\ that\ result\ from\ the\ action$
- Example (blocks world):
  - Possibility axiom:
    $\forall s\ Clear(A, s) \wedge Clear(B, s) \Rightarrow Poss(Stack(A, B), s)$
  - Effect axiom :$\forall s\ Poss(Stack(A, B), s) \Rightarrow$
    $On(A, B, Result(Stack(A, B), s)) \wedge$
    $\neg Clear(B, Result(Stack(A, B), s)$

Frame Problem

- Problem: Effect axioms say what changes, but not what stays the same!
- In the above example: How can we infer $Clear(A, Result(Stack(A, B), s)$?
- **Frame problem**: Problem of representing all things that stay the same
- Expressing what does stay the same through **frame axioms** is one possibility

Frame Problem

- Costly, would require $O(AF)$ frame axioms for A actions and F fluents
- **Representational frame problem**: If any action has at most E effects, would like to make do with $O(AE)$ rules instead
- **Inferential frame prolem**: Would like to project results of t-long action sequence in time $O(Et)$ rather than $O(Ft)$ or $O(AEt)$
- **Qualification problems**: Capturing *all* conditions for successful action (no solution)

Representational Frame Problem

- Solution: Use **successor-state axioms**
  Action is possible $\Rightarrow$
  (Fluent is true in result state $\Leftrightarrow$ *Action's effect made it true* $\vee$ *It was true before and action left it alone)*
- Example:

$$
\begin{aligned}
Poss(a, s) \Rightarrow (Clear(A, Result(a, s)) \Leftrightarrow \\
(On(B, A, s) \wedge a = UnStack(B, A)) \\
\vee (Clear(A, s) \wedge a \neq Stack(B, A)))
\end{aligned}
$$

- Solves problem, because each effect of an action is only mentioned once (note use of "$\Leftrightarrow$")
- **Ramification problem**: dealing with implicit effects

Inferential Frame Problem

- In projecting consequences, we still need $O(AEt)$ inferences for t time steps
- Mostly involves copying unchanged fluents
- But if only one action is executed at a time, why consider all of them?
- Reconsider format of frame axiom for fluent F_i:

$$\begin{aligned} Poss(a, s) \Rightarrow & \\ F_i(Result(a, s)) \Leftrightarrow & \quad (a = A_1 \vee a = A_2 \ldots) \\ & \vee F_i(s) \wedge (a \neq A_3) \wedge (a \neq A_4) \ldots \end{aligned}$$

Inferential Frame Problem

- We can rewrite this using positive and negative effects (that make fluent true or false):

$$Poss(a,s) \Rightarrow$$
$$F_i(Result(a,s)) \Leftrightarrow PosEffect(a,F_i) \vee [F_i(s) \wedge \neg NegEffect(a,F_i)]$$

$PosEffect(A_1, F_i)$ $PosEffect(A_2, F_i)$
$NegEffect(A_3, F_i)$ $NegEffect(A_4, F_i)$

- Appropriate indexing ➡ retrieve effects of a given action A and corresponding axioms for F_i in $O(1)$
- Represent new situation by the old situation and "delta" (if nothing happens, nothing needs to be done)
- Achieves prediction in $O(Et)$

Summary

- Notion of ontology
- Discussed modelling of interesting types of knowledge
  - Categories
  - Physical Composition, Measurements, Substances/Objects
  - Actions and Change, frame problem
- Other interesting stuff we did not deal with:
  - Time, intervals, continuous processes, etc.
  - Multiple overlapping actions, multiple agents
- Next time: **category reasoning systems**