Knowledge Modelling and Management

Part A: Ontologies
Monday 16:10-17:00 and Thursday 16:10-17:00
Stuart Aitken stuart@aiai.ed.ac.uk
Room 7.09b Appleton Tower
Weeks 1-5

Part B: Conceptual Modelling Methods
Dr Jessica Chen-Burger
Weeks 5-10

KMM Website:
http://www.inf.ed.ac.uk/teaching/courses/kmm/

Knowledge Modelling and Management

Part A: Ontologies
10 Lectures
Part B: Conceptual Modelling Methods
10 Lectures
2 Assessed Exercises for KMM
accounting for 25%
Exam accounts for 75% of marks
(50% on Ontologies, 50% on Methods)

Tutorial material will be posted on the KMM website

Course outline

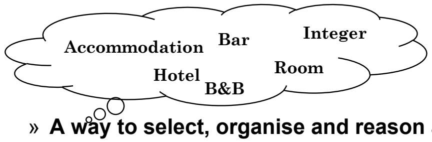
- Introduction - What is an Ontology?
  - Reading:
    - » For further reading see the course website
    - » Article: “What are ontologies?” Chandrasekaran (IEEE IS `99)
    - » Book: *Ontological Engineering* Gomez-Perez (Springer) Ch 1
      - copies in the Main Library
      - specific page references for L1: [OE p1-38]
- Methodology and the Protégé tool (conceptual approaches)
- Description logic and OWL (logical approaches)
  Parent = Person *hasChild* some Person
  Uncle = Man *hasSibling* some Parent
- Examples: CYC, Gene Ontology, Enterprise Ontology
- Parts and wholes (conceptual and logical approaches)
- Principles for organising and evaluating class hierarchies

Ontologies

- As a branch of philosophy, ontology is the science of *what is*, the kinds and structures of objects, properties, events, processes and relations in every area of reality.
  - ‘metaphysics’ (used by Aristotle) is a synonym
  - The term ‘ontologia’ was coined in 1613
  - ‘ontology’ first recorded in the OED in 1721
  - Linnaeus’ taxonomy of plants, and binomial naming scheme e.g. Homo sapiens <Genus> <species> (1753)
  - Too many schools of thought to list here…except:
- The ‘adequatist’ view of ontology as a descriptive enterprise
  - Ontology as taxonomy and description, not explanation
  - Now supported by the formal tools of logic

Ontologies

- What is an ontology?
  - A set of concepts and relations



subclass-of
part-of
no-of-rooms

    - » A way to select, organise and reason about concepts
  - Knowledge about general propositions that are true of concepts (representing the real world)
    - » Not (typically) concerned with individual hotels/rooms/ pubs
    - » Focused on definitions of concepts
    - » Usually resulting in computer implementations
  - Defined informally, or using
    - » Ideas from Set theory
    - » Logic

Ontologies

- Why are ontologies useful?
  - Communication between computers
    - » <cost> £10</cost> vs <price>£10</price>
  - Communication between people and computers
    - » friend(ann, jane) - how should this be read?
  - Communication between people
    - » Technical terms are often used with different meaning by different groups in an organisation: 'customer'
- Comparable with:
  - OO modelling/ UML
  - XML
  - Database schema
- Differentiated by:
  - Capturing knowledge at a level above the encoding

[OE p21-25]

Why would this be useful?

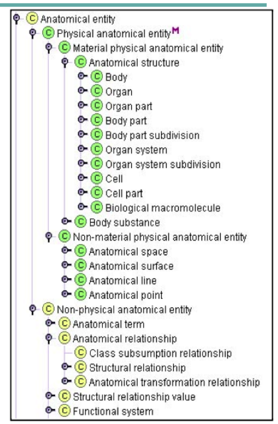
- Interoperability costs $15.8 billion annually in the US (1-2% of construction industry revenue)*
- ISO 15926 – an ontology for the construction industry **
  - Integration of life-cycle data for process plants including oil and gas production facilities
  - The value of ISO 15926 stems entirely from the benefit of being able to transfer information:
    - » directly from one software application to another
    - » without needing prior knowledge of either application
    - » automatically
    - » while maintaining the meaning of all the data values transferred

*http://www.fire.nist.gov/bfrlpubs/build04/PDF/b04022.pdf
**https://www.posccaesar.org/wiki/ISO15926Primer_Benefits

Ontologies

Examples

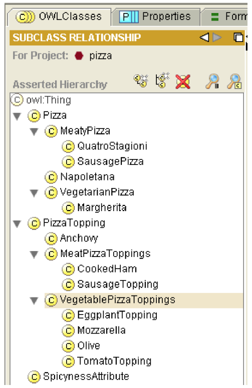
- A medical ontology
  - Hierarchical organisation of classes
  - Levels of granularity
  - Attributes distinguish the major branches of the taxonomy
    - » physical
    - » Material
- Uses
  - Indexing genetic data
  - Combining with clinical observations
    - » Drug targets and side effects
  - Recording treatments



Ontologies

Examples

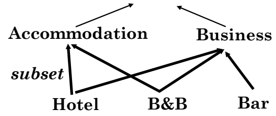
- A pizza ontology
  - Hierarchical organisation of classes
  - Automatic computation of the hierarchy based on the definitions of concepts
- Uses: tutorial example



What is an ontology?

Quick answer:

- A classification of types of entities (concepts/classes)
- Constraints on what can be said about them



$\forall x\ \ x \in \text{Hotel} \Rightarrow x \notin \text{B\&B}$ [Hotels and B&Bs are disjoint]

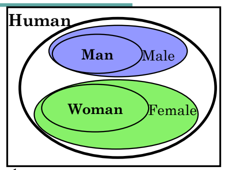
$\forall x\ \exists y\ x \in \text{Hotel} \Rightarrow \text{part-of}(y, x) \wedge y \in \text{Bar}$ [all Hotels have a Bar]

Plus:

- the domain and ranges of relations
- properties such as transitivity

Ontological modelling

- Concepts (class, category, type)
  - Can be thought of as a set of instances
  - However, identity between concepts is not established by having the same members
- Instances
  - The concrete entities (individuals) that instantiate a Concept
  - Assertions about individuals would be considered part of a knowledge base, not part of an ontology



Venn Diagram

Ontology	$\forall x\ \ x \in \text{Female} \Rightarrow x \notin \text{Male}$
Knowledge Base	$\text{ann} \in \text{Woman} \wedge \text{married(ann, joe)}$

Ontological modelling

- Relationships
  - N-ary predicates (unary, binary, ternary,…)
  - Unary predicates: man(adam) are typically modelled as Concepts:

    Man

    adam ∈ Man ; adam ∈ Male ; adam ∈ Human
  - Most often, analysis will reduce relationships to be binary: hasMother(adam, ann)
  - Exceptions: between(door, table, window)
- Functions: + - / * (meters 5.2)

Ontologies

A note about syntax -

atomic formula in first-order logic will be written:

predicate(argument1, argument2)

To denote set membership: $x \in C$

¬A	negation/not	A⇔B	if and only if
A∧B	and	A≡B	is defined
A∨B	or	∀x A	forall x
A⇒B	implies	∃x A	exists an x

Letting p and q be predicates, well-formed formulas include:

∀x p(x)

∀x p(x) ⇒ q(x)

The quantifier ∀ may be dropped for legibility - x,y,z typically denote variables. (Variables may be indicated by ?x.)

Ontological modelling - Classes and subClassOf

- The basis of an ontology is the categorisation of entities into classes, i.e. the class hierarchy
  - subClassOf relates the subclass to the super class: subClassOf(SUBCLASS, SUPERCLASS)
  - subClassOf is defined in RDFS (a W3C standard)
  - also known as *is-a*
- subClassOf is defined in terms of instance-of (type in RDF)

  type(a, C) ≈ a ∈ C

  subClassOf(S, T) ⇔ (∀x type(x, S) ⇒ type(x, T))

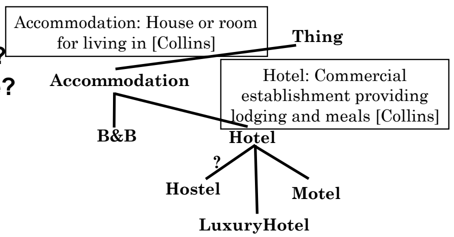
Note that S and T are the *names* of concepts - an alternative is to make S and T unary predicates:

∀x S(x) ⇒ T(x)

- convince yourself that this is equivalent to subClassOf(S, T)

Ontological modelling -Classes and subClassOf

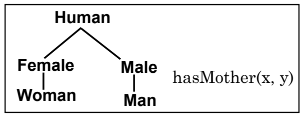
- How are concepts named?
- What constitutes a definition?
- How is a categorisation made?
  - Can it be computed?



- Methodology
  - Empirical
  - Differentia
  - Psychological
    - » Natural kinds
    - » tables and chairs vs furniture vs objects
  - Reuse
    - » Can concepts defined elsewhere be reused?
- Subject of Lecture 2

Ontological modelling - Relations

Relations and their argument types



- Domain domain(P, C):

  P(x, y) ⇒ type(x, C)

  E.g. domain(hasMother, Human)
  - Multiple domain statements imply that the domain is the intersection
    - » Adding domain(P, D) means all x lie in C ∩ D
- Range range(P, C):

  P(x, y) ⇒ type(y, C)

  E.g. range(hasMother, Woman)
  - Multiple range statements imply that the range is the intersection
- Higher-arity relations also have specific argument types
- Cardinality: constrain the number of instances of the <u>Range</u> type: # hasMother == 1

Ontological modelling - Relations

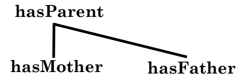
Relations and their properties

- **Transitive**
  $P(x, y) \wedge P(y, z) \Rightarrow P(x, z)$
  part-of(a, b) ∧ part-of(b, c) ⇒ part-of(a, c)
- **Symmetric**
  $P(x, y) \Leftrightarrow P(y, x)$
  connected-to(a, b) ⇔ connected-to(b, a)
- **Reflexive**
  $P(x, x)$
  part-of(a, a)
- **Irreflexive**
  $\neg P(x, x)$
  ¬hasMother(a, a)
- **Functional**
  $P(x, y) \wedge P(x, z) \Rightarrow y=z$
  hasMother
- **Inverse**
  $P(x, y) \Leftrightarrow Q(y, x)$
  hasMother(a, b) ⇔ motherOf(b, a)

Ontological modelling - Relations

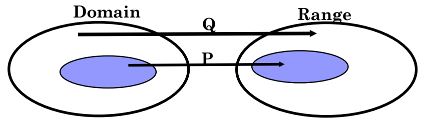
Relations and sub-properties

- Relations can be 'hierarchically organised'



subPropertyOf(hasMother, hasParent)

subPropertyOf(P, Q): $P(x, y) \Rightarrow Q(x, y)$



Ontological modelling -Attribute Relations

Attributes - a type of Relation distinguished by its range

- Attributes and values
  - Attributes are binary relations
  - The range is a specified set of instances (these instances are part of the ontology)
    SizeValue = {small, medium, large}
  - Attributes may be ordered/partially ordered
  - small<medium ∧ medium<large
    [and < is transitive]
  - hasSize attribute would have the range SizeValue
  - hasSize might be functional
    Write: hasSize(joe, large)

Ontological modelling -Attribute Relations

Attributes and Classes

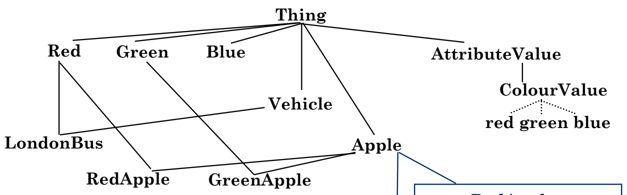
- Attributes can be drawn from a set of discrete attribute values or from a continuous range (real numbers)
  - Nominal (mutually exclusive and exhaustive)
  - Ordinal (identify relative magnitudes)
- Attribute relations: binary; functional; hierarchically organised
- Red - a class or an attribute ?

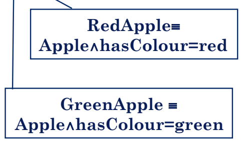
A. The set of red things: type(a, Red) [or red(a)]
Or
B. define ColourValue = {red, green, blue}
range(hasColour, ColourValue)
and state: hasColour(a, red)

Ontological modelling -Attribute Relations

Attributes and Classes

- Is Red a justifiable class ?

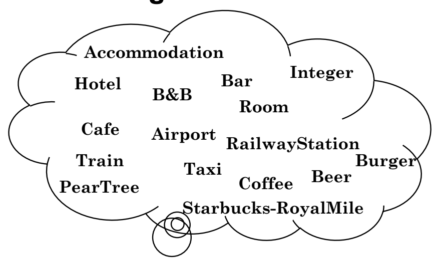




'Apple' supplies the means to count/distinguish particulars, while 'Red' can only do so for particulars (individuals) that have been distinguished already (Strawson, Guarino)
- Sortals/nonsortals
- Common nouns/adjectives

Example of Ontology Building

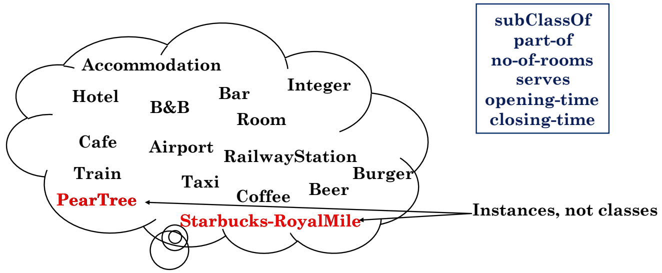
- Task: Organise some concepts related to travel, food and accommodation
  - Represent things that exist
  - Not design an OO model



subClassOf
part-of
no-of-rooms
serves
opening-time
closing-time

Example

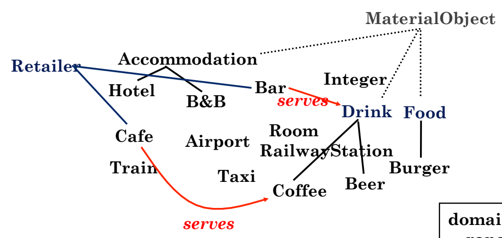
- For the moment: ignore the purpose of the ontology, and the method for generating concepts



subClassOf
part-of
no-of-rooms
serves
opening-time
closing-time

Example

- Identify subclasses and general concepts that group more specific types of thing
- Natural language usage is a guide



subClassOf
part-of
no-of-rooms
serves
opening-time
closing-time

domain of *serves* is Retailer
range of *serves* is Drink

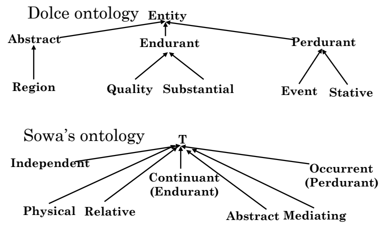
Types of ontology

- We have seen:
  - Simple ontologies
    - » Human, Male, Man, Female, Woman,…
    - » Accommodation, Hotel, B&B,…
  - Relations used to specify ontologies
    - » subClassOf, type, domain, range,…
- These are examples of Application Ontologies and Knowledge Representation Ontologies
- Ontology Types:
  - Knowledge Representation
  - Upper-level
  - Generic
  - Application (Domain) Ontologies

Types of ontology

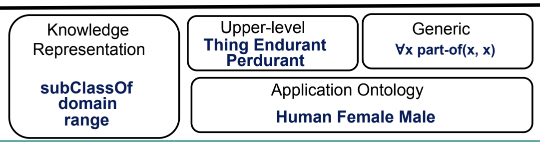
- **Upper-level Ontologies**

Abstract concepts used for organising the ontology:



- **Generic Ontologies: Mereology, Topology**

1. $\forall$x y z part-of(x, y)$\wedge$ part-of(y, z) $\Rightarrow$ part-of(x, z) [transitive]
2. $\forall$x part-of(x, x) [reflexive]
3. $\forall$x y part-of(x, y)$\wedge$ part-of(y, x) $\Rightarrow$ x=y [constraint on identity]



Types of ontology

Increasing formality

- Degree of formalisation
  - Controlled vocabulary: a finite list of terms, e.g. a catalogue
  - Glossary: meaning of terms defined in natural language
  - Thesauri: adds semantics e.g. synonyms

  Natural language resources e.g. WordNet

  - Informal is-a (subClassOf): e.g. subject hierarchies in Yahoo e.g. Computers>Software>AI
  - Formal is-a (subClassOf): subsumption properly defined so that transitivity holds

  Medical/ Bioinformatics

  - Frames: ontology defines classes and properties
  - Value restriction: properties are defined
  - General logical constraints: arbitrary constraints in first-order logic between terms

  AI Ontologies

Ontologies in use

- Gene Ontology www.geneontology.org ~19,000 terms
  - Molecular function; Biological process; Cellular location
  - Use for annotation of experimental data in databases
  - Visualisation and knowledge discovery across the database
  - Even used for the prediction of biological function
- Unified Medical Language System (UMLS) ~100K terms - MESH used for indexing Medline articles
- Cyc www.cyc.com ~millions of axioms and facts. Cyc Knowledge Based System & Web search trialed by beta.hotbot.com (Lycos)
- Galen and FMA - medical ontologies ~100K terms
- DAML - many OWL ontologies at www.daml.org

Ontologies

Ontology is concerned with standards - but there are so many to choose from…

- Language standards
  - OWL (Web Ontology Language) and the XML/RDF syntax
  - KIF (Knowledge Interchange Format) a previous proposed standard - will not be covered here
  - and many other languages….
- Modelling standards, however, there are
  - Diverse upper-level ontologies
  - Diverse approaches to formality
- Applications
  - Motivate design choices from an engineering perspective
- Philosophy
  - Motivate design choices from a wider perspective
- No overall coherent theory, rather, elements to be understood individually, and applied in combination

What is an ontology?

- An ontology is a specification of a conceptualization. (Thomas Gruber) That is, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents.
- the metaphysical study of the nature of being and existence (WordNet)
- In knowledge-based systems, an ontology is that part of the system which specifies what things exist and what is true about them. Cyc's ontology is essentially its whole knowledge base. You may hear people refer to their "ontology of devices" or their "temporal ontology". What they are talking about is those parts of their knowledge base (the constants and assertions) that concern devices or time. (Cycorp)

 [OE p6-11]

Ontologies

From this lecture, note:

- the <u>vocabulary</u> used to construct the KR ontology:
  - type, subClassOf, domain, range, subPropertyOf
  - The meaning and definition of these relations
- the <u>definitions</u> of properties that relations might have:
  - transitive, symmetric, reflexive etc + cardinality
- types of ontology
- some examples of classes and relations about a particular domain:
  - Concepts/Classes: Man, Woman, etc
  - Assertions: type(ann, Woman) hasMother(adam, ann)

What's Next

- A more explicit Methodology for ontology development
- Description Logic
  - Logical system ALC
    - » Concept definitions
    - » Reasoning (the tableaux algorithm)
  - Web Ontology Language (OWL and OWL 1.1)
    - » Semantics
    - » RDF/XML syntax
    - » Manchester syntax