Foundations of the Semantic Web: Ontology Engineering

Building Ontologies 1

Alan Rector & colleagues

1

Goals for this module: for you

- Be able to implement an ontology representation in OWL-DL
  - Be able to elicit a conceptualisation
  - Be able to formulate an ontology representation
  - Be able to implement the ontology representation in OWL-DL
    - Or be able to say you can't
    - To understand the limits of OWL-DL ontologies
  - Be able to test the resulting ontology implementation
  - Be ready to apply ontology representations in any of several use cases
    - In one week, we can't build the applications…
      …but to build an ontology is only a means to building applications
      - Without applications ontologies are pointless

2

Goals for this Module: For us

- Still experimental – we need your feedback
  - Feedback
    - On tools – we treat this as a User Centred Design experiment
      - Please be patient
        - The good news is they are getting better
    - On the course
      - Did the content work for you?
      - What other content would you like?
      - Balance of labs and lecture
      - Content of labs
    - For the Semantic Web Best Practice Working Group
  - New ideas

3

Mechanics - reminder

- Assessment
  - 25% lab
  - 25% Mini project
  - 50% Exam
- All labs to be handed in by number electronically – see lab handout
- Deadline – 2 weeks after end of course

4

Ontologies and Ontology Representations

- “Ontology” – a word borrowed from philosophy
  - But we are necessarily building logical systems
    - “Physical symbol systems”
      - Simon, H. A. (1969, 1981). *The Sciences of the Artificial,* MIT Press
- “Concepts” and “Ontologies”/ “conceptualisations” in their original sense are psychosocial phenomena
  - We don’t really understand them
- “Concept representations” and “Ontology representations” are engineering artefacts
  - At best approximations of our real concepts and conceptualisations (ontologies)
    - And we don’t even quite understand what we are approximating

5

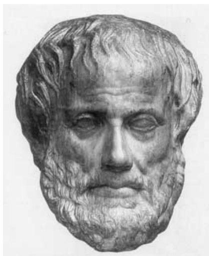
Ontologies and Ontology Representations (cont)

- Most of the time we will just say “concept” and “ontology” but whenever anybody starts getting religious, remember…
  - ***It is only a representation!***
    - We are doing engineering, not philosophy – although philosophy is an important guide
- *There is no one way!*
  - But there are consequences to different ways
    - and there are wrong ways
      - and better or worse ways for a given purposes
  - *The test of an engineering artefact is whether it is fit for purpose*
    - Ontology representations are engineering artefacts

6

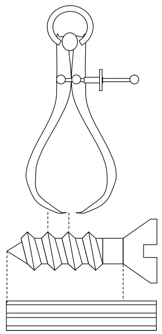
What Is An Ontology?

- Ontology (Socrates & Aristotle 400-360 BC)
- The study of being
- Word borrowed by computing for the explicit description of the conceptualisation of a domain:
  - concepts
  - properties and attributes of concepts
  - constraints on properties and attributes
  - Individuals (often, but not always)
- An ontology defines
  - a common vocabulary
  - a shared understanding



Measure the world…*quantitative models*

(not ontologies)

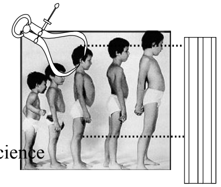


- Quantitative
  - Numerical data:
    - 2mm, 2.4V, between 4 and 5 feet
  - Unambiguous tokens
  - Main problem is accuracy at initial capture
  - Numerical analysis (e.g. statistics) well understood
- Examples:
  - How big is this breast lump?
  - What is the average age of patients with cancer ?
  - How much time elapsed between original referral and first appointment at the hospital ?

8

describe the our understanding of the world - *ontologies*

- Qualitative
  - Descriptive data
    - Cold, colder, blueish, not pink, drunk
  - Ambiguous tokens
    - What's wrong with being drunk ?
      - Ask a glass of water.
  - Accuracy poorly defined
  - Automated analysis or aggregation is a new science
- Examples
  - Which animals are dangerous ?
  - What is their coat like?
  - What do animals eat ?



Light and Heavy expressivity

A matter of rigour and representational expressivity

- Lightweight
  - Concepts, atomic types
  - Is-a hierarchy
  - Relationships between concepts
- Heavyweight
  - Metaclasses
  - Type constraints on relations
  - Cardinality constraints
  - Taxonomy of relations
  - Reified statements
  - Axioms
  - Semantic entailments
  - Expressiveness
  - Inference systems

[Deborah McGuinness, Stanford]

So what is an ontology?

Catalog/ ID

Thesauri

Formal Is-a

Frames (properties)

General Logical constraints

Terms/ glossary

Informal Is-a

Formal instance

Value restrictions

Disjointness, Inverse, partof

Gene Ontology

Mouse Anatomy

Arom

EcoCyc

TAMBIS

PharmGKB

[Mike Uschold, Boeing Corp]

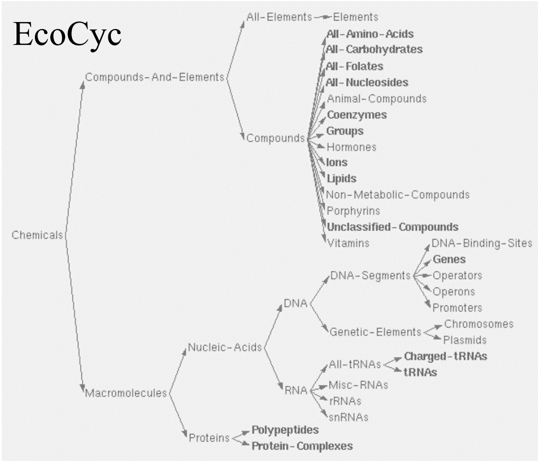
A semantic continuum

Pump: "a device for moving a gas or liquid from one place or container to another"

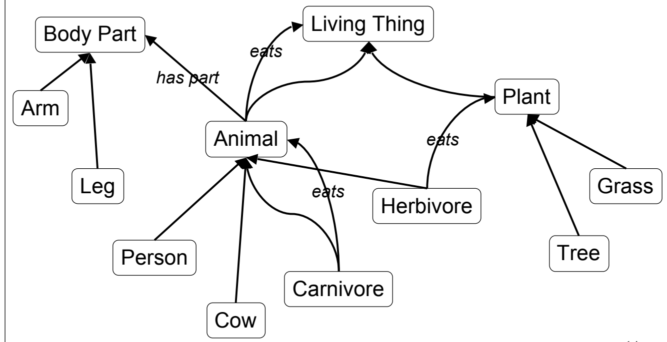
(pump has (superclasses (...))

Shared human consensus	Text descriptions	Semantics hardwired; used at runtime	Semantics processed and used at runtime
Implicit	Informal (explicit)	Formal (for humans)	Formal (for machines)

Further to the right means:
- Less ambiguity
- More likely to have correct functionality
- Better inter-operation
- Less hardwiring
- More robust to change
- More difficult



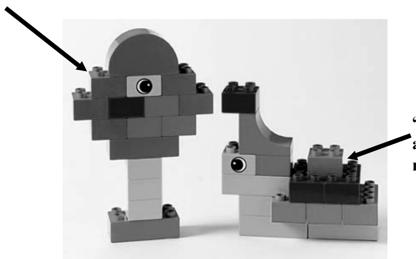
A simple ontology: Animals



14

Logic-based Ontologies: Conceptual Lego: A BioInformatics View

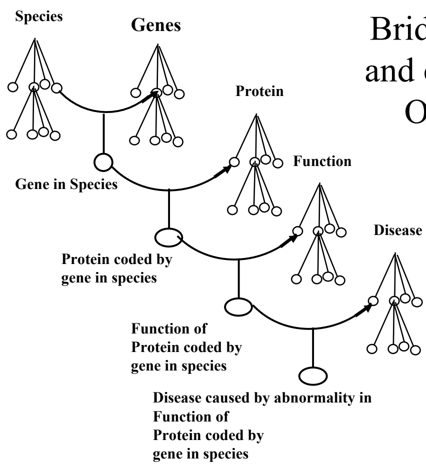
"SNPolymorphism of CFTRGene causing Defect in MembraneTransport of ChlorideIon causing Increase in Viscosity of Mucus in CysticFibrosis…"



"Hand which is anatomically normal"

15

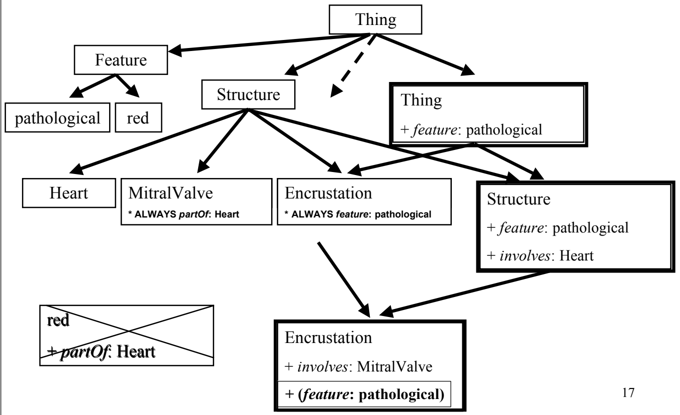
Bridging Scales and context with Ontologies



16

Logic Based Ontologies: A crash course

Primitives **Descriptions** **Definitions** **Reasoning** **Validating**



17

Why Develop an Ontology?

- To share common understanding of the structure of descriptive information
  - among people
  - among software agents
  - between people and software
- To enable reuse of domain knowledge
  - to avoid "re-inventing the wheel"
  - to introduce standards to allow interoperability

18

Why build an ontology

- Interworking and information sharing
  - Providing a well organised controlled vocabulary
- Indexing complex information
  - "Knowledge is fractal"
    - Ontologies are fractal
      - Self similar structure at every level of granularity (detail)
- Combat combinatorial explosions
  - The exploding bicycle
    - "Conceptual Lego"
      - A "dictionary and grammar" instead of a "phrasebook"

19

Ontology Examples

- Taxonomies on the Web
  - Yahoo! categories
- Catalogs for on-line shopping
  - Amazon.com product catalog
- Dublin Core and other standards for the Web
- Domain independent examples
  - Ontoclean
  - Sumo

20

Upper Ontologies

- Ontology Schemas
  - High level abstractions to constrain construction
    - e.g. There are “Objects” & “Processes”
  - Highly controversial
    - Sumo, Dolce, Onions, GALEN, SBU,…
  - Needed when you work with many people together
  - NOT in this tutorial – a different tutorial

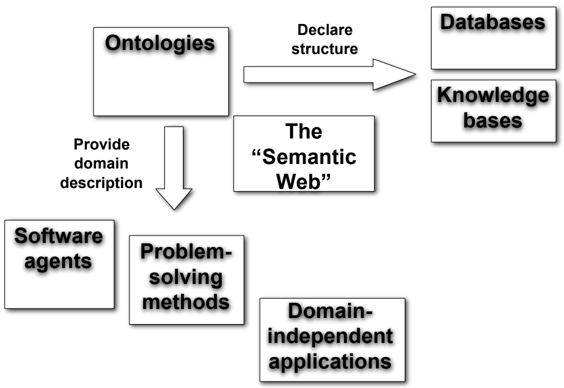
21

Domain Ontologies

- Concepts specific to a field
  - Diseases, animals, food, art work, languages, …
  - The place to start
    - Understand ontologies from the bottom up
      - Or middle out
- Levels
  - Top domain ontologies – the starting points for the field
    - Living Things, Geographic Region, Geographic_feature
  - Domain ontologies – the concepts in the field
    - Cat, Country, Mountain
  - Instances – the things in the world
    - Felix the cat, Japan, Mt Fuji

22

An Ontology should be just the Beginning



23

Ontology Technology

- “Ontology” covers a range of things
  - Controlled vocabularies – e.g. MeSH
    - Linguistic structures – e.g. WordNet
  - Hierarchies (with bells and whistles) – e.g. Gene Ontology
  - Frame representations – e.g. FMA
  - Description logic formalisms – Snomed-CT, GALEN, OWL-DL based ontologies
  - Philosophically inspired e.g. Ontoclean and SUMO

24

OWL
The Web Ontology Language

- W3C standard
- Collision of DAML (frames) and Oil (DLs in Frame clothing)
- Three ‘flavours’
  - OWL-Lite –simple but limited
  - OWL-DL – complex but deliverable (real soon now)
  - OWL-Full – fully expressive but serious logical/computational problems
    - Russel Paradox etc etc
  - All layered (awkwardly) on RDF Schema
- Still work in progress – see Semantic Web Best Practices & Deployment Working Group (SWBP)

25

Description Logics

- What the logicians made of Frames
  - Greater expressivity and semantic precision
    - Compositional definitions
      - “Conceptual Lego” – define new concepts from old
- To allow automatic classification & consistency checking
  - The mathematics of classification is tricky
    - Some seriously counter-intuitive results
      - The basics are simple – devil in the detail

26

Description Logics

- Underneath:
  - computationally tractable subsets of first order logic
- Describes relations between Concepts/Classes
  - Individuals secondary
    - ***DL Ontologies are NOT databases!***

27

Description Logics:
A brief history

- Informal Semantic Networks and Frames (pre 1980)
  - Wood: *What’s in a Link*; Brachman *What IS-A is and IS-A isn’t*.
- First Formalisation (1980)
  - Bobrow *KRL*, Brachman: *KL-ONE*
- All useful systems are intractable (1983)
  - Brachman & Levesque: *A fundamental tradeoff*
    - Hybrid systems: T-Box and A-Box
- All tractable systems are useless (1987-1990)
  - Doyle and Patel: *Two dogmas of Knowledge Representation*

28

A brief history of KR

- 'Maverick' incomplete/intractable logic systems (1985-90)
  - GRAIL, LOOM, Cyc, Apelon, …,
- Practical knowledge management systems based on frames
  - Protégé
- The German School: Description Logics (1988-98)
  - Complete decidable algorithms using tableaux methods (1991-1992)
  - Detailed catalogue of complexity of family – "alphabet soup of systems"
- Optimised systems for practical cases (1996-)
- Emergence of the Semantic Web
  - Development of DAML (frames), OIL (DLs) → DAML+OIL → OWL
    - **Development of Protégé-OWL**
    - **A dynamic field – constant new developments & possibilities**

29

And beware Ontologies are not databases!

- Ontologies are (mostly) about the classes –
  - Can be used to represent database aspects of ***schemas***
    - *What must be true of any database consistent with the schema*
  - The Terminology
    - *What must be true of any concept consistent with the ontology*
  - The "T-Box" – for "terminology box"
- Limited functionality for individuals ('instances')
  - Primarily to help define classes
    - The class of John's shirts, The class of cities in Japan
  - To describe individuals use
    - A database
    - Triple representation (RDF or Topic Maps)
    - An instance store
    - Perhaps with an ontology as the schema
  - Open world instead of closed world
    - Individuals in ontologies (The "A-Box") poorly understood and very high computational complexity

30

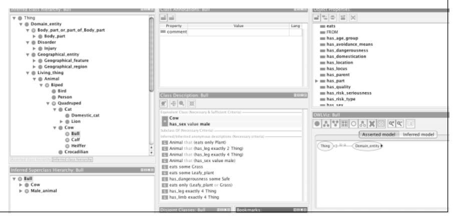
Approach

- Design patterns
  - Analogous to Java design patterns
    - Standard ways to do things
      - Someday they will be supported by tools, but today you have to do it yourself
  - Being codified by Semantic Web Best Practice Working Group
- Elephant traps
  - Common errors & misconceptions
    - Especially those that seem to work at first
- Foundations of knowledge representation
  - 200 to 2000 years of experience & mistakes you need not repeat
- Common dilemmas & tradeoffs
  - Things for which we don't have a perfect answer

31

Protégé OWL: New tools for ontologies

- Transatlantic collaboration
- Implement robust OWL environment within PROTÉGÉ framework
  - Version 4-A1pha - complete rewrite
  - You will be guinea pigs - and we will have human facts folk seeing what problems you have
  - New ideas for debugging, visualisation, ontology management, etc.



Protégé-OWL & CO-ODE

- Joint work: Stanford & U Manchester + Southampton & Epistemics
  - Please give us feedback on tools – mailing lists & forums at:
    - **protege.stanford.edu**
    - **www.co-ode.org**
- Don't beat your head against a brick wall!
  - Look to see if others have had the same problem; If not…
  - ***ASK!***
    - ***We are all learning.***

33

OWL-DL & Classification

- Not all of OWL-DL can yet be implemented
  - We will deal mostly with what can be classified using Racer or FaCT++
  - Not all of the things that are implemented scale successfully
    - All classifiers are worst-case exponential (or worse)
- FaCT++
  - Classifier being developed here
    - Dmitry Tsarkov/Ian Horrocks
- Pellet
  - Classifier from originally MindSwap (U Maryland) www.mindswap.org but now here
    - Bijan Parsia
    - Best integrated with Protégé at the moment.
- We will try to provide warnings of things which cannot be classified or do not scale
  - But you may discover new things on your own

34

Example Ontologies for this Module

- Pizzas
  - For the mechanics of OWL and Protégé/OWL
    - Simple – no ontological problems, just mechanics
- Animals for best practice examples and ontology building
  - The example for you to work from
    - Also for examples of parts and wholes
- The University and courses
  - Your job is to build an ontology for the University by analogy to the examples
    - with some specific help
    - Leads on to major ontological issues
- Simple Upper Ontology
  - To put it together
    - Mostly about the University

35

Building Ontologies

- Basic Concepts and Mechanics

36

Why it's hard (1)

- Clash of intuitions
  - Subject Matter Experts motivated by custom & practice
    - Prototypes & Generalities
  - Logicians motivated by logic & computational tractability
    - Definitions and Universals
- Transparency & predictability vs
  Rigour & Completeness
- Neophytes (you?) caught in the muddled middle

37

Why it's hard (2)

- Conflation of Models
  - Meaning: Correctness of Classification & retrieval
  - Indexing: Task of discovery, search, or finding
  - Use: Task of data entry, decision support, …
  - Acquisition: Task of capturing knowledge
- Assuring quality & managing change
  - Quality assurance: Criteria for whether it is 'correct'
  - Evolution Coping with change
  - Regression testing Controlling changes & maintaining Quality

38

Why its hard (3)

- Confusion of terminology and usage
  - Religious wars over words and assumptions
- The intersection of
  - Linguistics
  - Cognitive science
  - Software engineering
  - Philosophy
  - Human Factors
- A jumble of syntaxes

39

Vocabulary

- "Class" ≈ "Concept" ≈ "Category" ≈ "Type"
- "Instance" ≈ "Individual"
- "Entity" ≈ "object", Class or individual
- "Property" ≈ "Slot" ≈ "Relation" ≈ "Relationtype" ≈ "Attribute" ≈ Semantic link type" ≈ "Role"
  - but be careful about "role"
    - Means "property" in DL-speak
    - Means "role played" in most ontologies
      - E.g. "doctor_role", "student role" …

40

Syntaxes

- Three official syntaxes + Protégé-OWL syntax
  - Abstract syntax-- -Specific to OWL
  - *N3 ----------------* -*OWL & RDF* -*used in all SWBP documents*
  - XML/RDF ------- -very verbose, not for human consumption
  - *"German DL"----* -*very concise, symbolic*
  - First order logic - - complete but more powerful than DL
  - *Manchester Syntax----* - *Intuitive keywords and infix notation*
- This tutorial uses simplified abstract syntax
  - someValuesFrom → ***some*** ∃
  - allValuesFrom → ***only*** ∀
  - intersectionOf → AND ⊓
  - unionOf → OR ⊔
  - complementOf → NOT ¬
  - complete ***definition*** ***necessary & sufficient***
  - partial ***description*** ***necessary***
- Protégé/OWL can generate all syntaxes except German

41

Why its hard (4)

- Clash with vocabulary and practice of related software disciplines
- Most OO analysis produces a set of templates
  - E.g. a Java Class is a template for a Java object
    - Nothing is permitted until there is a place for it in the template
- OWL is a way of specifying constraints
  - The criteria for being a member of a class
    - Everything is permitted until ruled out by a constraint

42

Clash with intuitions of related fields

- Object Oriented Programming
  - Java,a C++, Smalltalk, etc.
    - But OO programming is not knowledge representation
- Object Oriented Design (Databases)
  - But data models are not ontologies either
    - Although UML is often a good starting point
      - Additional a-logical issues
        - » Difference between attributes and relations
        - » Issues of life cycle and handling of aggregation
        - » Notion of an instance
        - » Implicitly "closed world"
- Frame based systems, Semantic Nets,… Traditional AI
  - Where it all started but real differences
- RDF(S), Topic Maps and other node-and-arc symbolisms
  - "What's in a link?"
  - The battles in standards committees continue

43

Summary of Approach
Steps in developing an Ontology (1)

1. Establish the purpose
   - Without purpose, no scope, requirements, evaluation,
2. Informal/Semiformal knowledge elicitation
   - Collect the terms
   - Organise terms informally
   - Paraphrase and clarify terms to produce informal concept definitions
   - Diagram informally
3. Refine requirements & tests

44

Summary of Approach
Steps in implementing an Ontology (2)

4. Implementation
   - Develop normalised schema and skeleton
   - Implement prototype recording the *intention as a paraphrase*
     - Keep track of what you meant to do so you can compare with what happens
       - Implementing logic-based ontologies is programming
   - Scale up a bit
     - Check performance
   - Populate
     - Possibly with help of text mining and language technology
5. Evaluate & quality assure
   - Against
   - Include tests for evolution and change management
   - Design regression tests and "probews"
6. Monitor use and evolve
   - ***Process not product!***

If this were three modules…

1. Knowledge elicitation and analysis
   - A quick overview
2. **Implementation**
   - **A solid introduction**
3. Evolution, ontology alignment, and management
   - Left for another module
     - But a major motivation for the methods taught in this module
       - Normalisation and documentation of intentions

Plan of Labs

- Lab 1 – the mechanics of OWL in Protégé Owl
  - The pizza example
- Lab 2 – Ontology building the life cycle
  - A more realistic example
  - Start building the University example
    - On the pattern of the lecture example of animals
- Lab 3
  - Problems and tricks of the trade
  - DL problems (IH)
- Lab 4
  - More on patterns and parts and whole
- Lab 5
  - Upper ontologies and clarification of the mini project

More Reasons

- To make domain assumptions explicit
  - easier to change domain assumptions (consider a genetics knowledge base)
  - easier to understand and update legacy data
- To separate domain knowledge from the operational knowledge
  - re-use domain and operational knowledge separately (e.g., configuration based on constraints)
- To manage the combinatorial explosion