

# Foundations of the Semantic Web: Ontology Engineering

## Building Ontologies 1b

### Classes and Instances Concepts & Individuals

#### Alan Rector & colleagues

Special acknowledgement to Jeremy Rogers & Chris Wroe

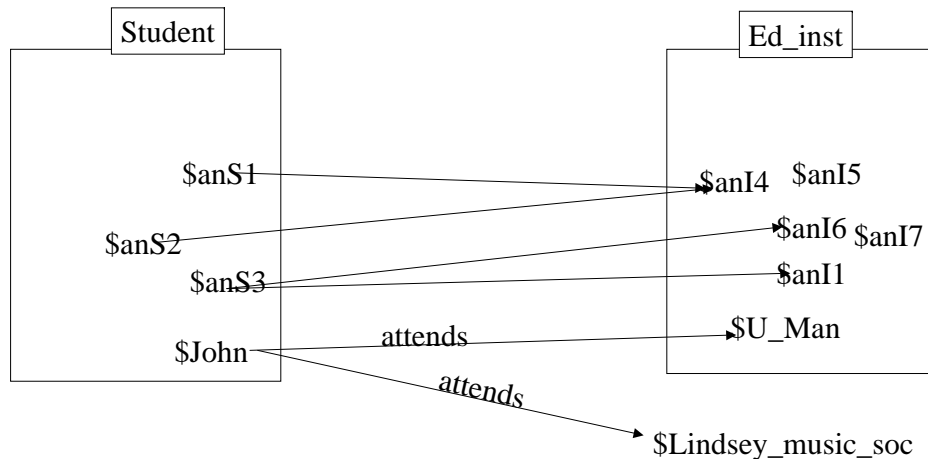
1

## “I am *an* individual”

- And my representation ought to be one to
  - Hence **\$Alan\_Rector** in the examples
- So are
  - This year’s version of CS646 –
    - Hence **\$CS646\_2003** in the examples
  - You
  - The University of Manchester
  - This room, its furniture, etc.
  - Your thoughts, understanding, ...
  - This lecture, the lab following it, ...
  - ...

2

## Individuals in our Example Everything with a ‘\$’ prefix<sup>1</sup>



<sup>1</sup> It would be more conventional to use '#', but this confuses OilEd terribly – taken as a namespace identifier

3

## What is a Class? An individual

- Vocabulary variations
  - “Class” ≈ “Type” ≈ “Concept” ≈ “Concept Representation” ≈ “Category”
  - “Individual” ≈ “Instance” ≈ “Object” (in OO programming)

4

## Individuals in Ontologies

- **Simple test 1:**  
**“Can it have kinds” – if so, it is a class**
  - “Kinds of dog” makes sense
  - “Kinds of person” makes sense
  - “Kinds of Alan Rector” does not make sense
  - “Kinds of Module” makes sense
  - “Kinds of CS646\_2003” does not make sense
  - “Kinds of jacket” makes sense
  - “Kinds of the ‘jacket I am wearing’” does not make sense

5

## Individuals in Ontologies (cont)

- Simple test 2:  
**If you say something about it,  
if you have made a new concept, then it is a class  
if you have just stated a fact about it, it is an individual.**
  - “Big dog” is a new class of dog
    - “Rover is big” just says something about Rover
      - Which would allow us to infer that Rover is a member of the class of “Big Dogs”
  - “Men with beards” is a new class
    - “Alan rector has a beard” is a fact about Alan Rector
      - Which would allow us to infer that he is a member of the class of “Men with Beards”

6

## Clues in English

- Articles + singular indicate individual
  - ‘the book there on the shelf’ – an individual
  - ‘a book’ – an unspecified individual
- Proper nouns (almost always) indicate individuals
  - Alan Rector, Ian Horrocks, Cross Street, Manchester, England, ...
- Plurals usually indicate classes
  - ‘the books’ – probably a class
    - Although possibly an individual aggregation
  - And perversely the English convention is to name classes in the singular

7

## More clues in English

- a ‘...that...’ clause and usually indicates a class
  - “The Modules that are available for ACS”
    - Perversely by convention Classes are given names in the singular in English
      - “Module that...”
- a ‘...which...’ clause depends on local usage
  - Some English stylebooks would have ‘which’ clauses used only for individuals, others say there is no real difference between ‘that’ and ‘which’
    - “MS Word usually asks for ‘that’ with plurals (classes) and ‘which’ with singulars
- No perfect guide, must take case by case.

8

## Leaf nodes are not Individuals

- Leaf node
  - Depends on ontology – may be very detailed, e.g.
    - Golden\_retriever\_bitch\_from\_karmella\_kennels\_from\_2003\_litter
      - Individual in that class “Halo”
- Even if there is only one possible individual, a leaf node is not an individual
  - Transferable\_skills\_course\_for\_first\_year\_PhD\_students\_in\_CS\_department
    - There might be other courses besides CS700
      - Its not impossible, just untrue
- Only individuals if there *could never be* kinds
  - CS646\_2003
    - There can never be a “kind” of this year’s course

9

## Keeping the Ontology Re-usable

- If we make leaf nodes individuals, we close off any extension to more granular kinds
  - Make the ontology specific to our immediate needs
  - Make extensions require radical surgery

10

## Comparison with “Instances” in databases, frames, and OO programming

- “Individuals” in ontologies are slightly different than in OO programming or data bases
- Test for individual
  - Ontologies – could it sensibly have kinds
  - Databases – is it going to be stored in a field in the database
  - OO programming – is it going to be an operational object in the program
  - RDF(S) – still some confusion
    - Anything can be an individual

11

## “Tangle at the Top”

- Many OO environments require that everything be an instance of something.
  - If everything must be an instance of something, then we have an infinite regress
    - Most systems stop it by having something be an instance of itself
      - Protégé, Smalltalk, and Java Class
      - RDF(S), OWL-Full: rdfs:resource
- Being an instance of yourself violates the semantics of OWL-DL
  - In OWL-DL, classes are not instances of anything
    - They are interpreted as the intensions of sets of individuals

12

## More vocabulary “Intensions” & “Extensions”

- “Intension”
  - The meaning of something
  - The definition of a class
    - “The lecturer the application part of this module”
    - “The evening star”
- “Extension”
  - The things which satisfy the meaning – the members of the class
    - Alan Rector
    - The planet Venus

13

## Extensional equality vs Intensional Equivalence

- Two sets are equal if their extensions are equal
  - In a particular model
    - The extensions of “The evening star” and “The morning star” are equal
- Two intensions are equivalent if if their extensions *must* be equal –
  - i.e. if their being unequal would be a contradiction in *any* model satisfying the same axioms
    - “Three sided polygon” is equivalent to “Three angled polygon” given the axioms of geometry

14

## Nominals - oneof

- Individuals should be able to be imported into class restrictions via *oneOf*
  - Staff\_for\_CS646\_2003  $\cong$   
*restriction* teaches someValuesfrom *oneOf* {CS\_646\_2003}
- Ignored in Racer, but can use hasValue ( $\ni$ )
  - Staff\_for\_CS646\_2003  $\cong$   
*restriction* teaches  $\ni$  cs\_646\_2003

15

## OWL-DL – and DLs are work best for Classes The “T-Box”

- Can be used as schemas for databases
  - “Closed world” reasoning
    - Negation as failure
- Can be used as an index for a store of instances
  - Excellent way to index things
- Difficult to use for true open world reasoning
  - Negation as impossibility/unsatisfiability
    - It is not known how to build a sound, complete, computationally tractable A-Box
      - In fact it is known that any sound complete A-Box will be worst case intractable.
- Little is lost if an individual is represented as a class
  - Much is lost if a class is represented as an individual
    - When in doubt, use classes

16

## Simulating Individuals as Leaf Nodes

- It often works better in current technology to simulate individuals as leaf nodes
  - We are providing a transparent way to do this, but it isn't finished yet.
  - Mark them in the comment field. Perhaps create a special annotation property.
    - pseudo-individual:true

17

## Individuals in Protégé

- On the Individuals Tab
  - A form is automatically generated for with a field for every property for which the class is explicitly in the domain.

18

## Individuals in Protege

- Protege handles individuals well, but Classifiers handle individuals badly
  - No support for individuals in FaCT reasoner
  - Limited support for individuals in Racer reasoner
  - Racer makes extra assumptions not made in OWL
    - All individuals are different
    - Reasoning is incomplete
      - All inferences found are correct, but some may not be found
- And this version has labelling problems
  - Labels things “Types” instead of “Classes” in the Individuals pane
- So we won't do much with individuals, but...

19

