INFS3101/7100 Ontology and the Semantic Web

Module 12 Advanced Issues

Last lecture: Key Terms

- **OWL** extends RDFS. **Object** and **datatype properties.** Universal class **Thing. Restrictions** are subsets of property domains. **Names** not satisfy unique names assumption. Class descriptions can be **enumeration** or **boolean combination** of classes. **Ontology properties. OWL Full, OWL DL, OWL Lite.**

RDFS/OWL vs Requirements

- Modules 1 - 9 requirements for ontology representation
- How does RFDS/OWL meet these?
  - Individual? Yes
  - Player/role/act/fact? No
  - Identity? No
  - Complex objects, unity? No
  - Bulk? No
  - Subsumption? Yes. Limited defined subclass
  - Formal ontology constructs? No
- But OWL has facilities for open world:
  - URI, namespaces

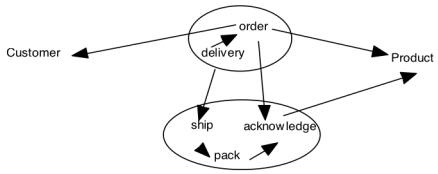
Ontology Platform Capability

- Ontology will be organised as series of metalevels of decreasing rigidity.
- Metamodel (eg MOF) most general, most rigid.
- Representation language (eg OWL) more specific, less rigid
- Ontology class and property system still more specific, less rigid
- Individuals most specific, most fluid
- Metamodels and representation languages have very low ontological commitment
- But there are many large classes of applications with more specific representation requirements, so can afford more ontological commitment

Specific Representation Requirements

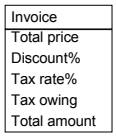
- Properties rather than attributes and associations
- Bulk classes
- Concept vs representational classes
- Dimension
- Mereological structures
- N-ary associations
- Extent-descriptive metaclasses
- We will look at each of these in turn

Properties rather than attributes and associations



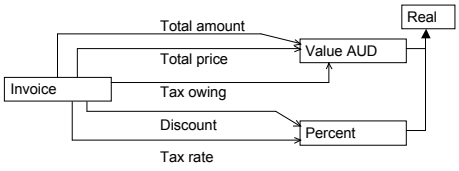
- Many relationships involving Product, all represented in the model

Properties rather than attributes and associations



- Three value amounts, two percentages
- Structural relationships not modeled
- Often in systems value sets implemented as say *real*
- Weak semantic content

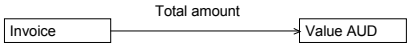
Properties rather than attributes and associations



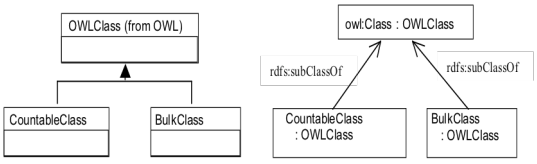
- Representation as properties shows structural relationships
- Semantically more meaningful
- Less scope for misunderstanding (increased clarity)

Bulk classes

- Bulk class lacks either unity or identity
- Projection, aggregation not meaningful with bulk classes, so is a structural issue



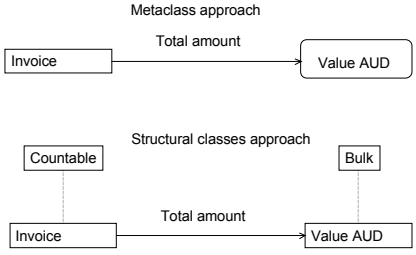
Bulk classes



Altering metamodel
Nicer tools

Adding structural classes
Easier to do

Bulk classes



Concept vs representational classes

- I am a person
- Most people call me Bob
- My mother calls me Robert
- The University knows me by a payroll number
- And also by a student number
- The Tax Department knows me by tax file no.
- Queensland Transport by my driver's licence no.
- All these are representations of the same thing
- For two to interoperate they must make a correspondence among their representations

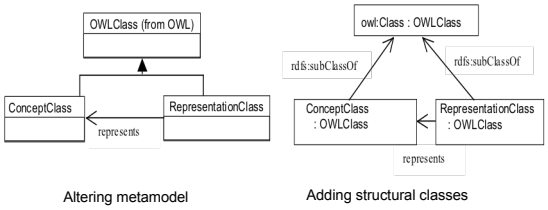
Concept vs representational classes

- A program can only see things by values of datatype properties, their representations
- In interoperation often necessary to link together several different representations of the same object.
  - Supplier product# with Purchaser product#
- Useful to have a concept class with no representations to help with this.
  - Product

Concept vs representational classes

- Particularly relevant to dimensions
- Concept class is the dimension
  - Distance, time, mass, value
- Representation classes are units
  - Metres, feet, centuries, seconds, kilograms, pounds, USD, EUR

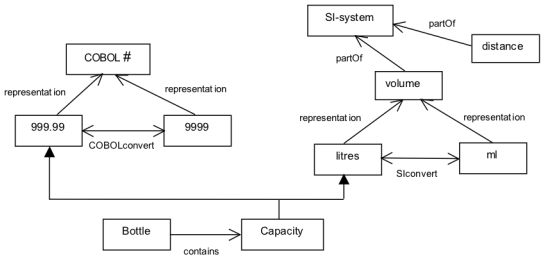
Concept vs representational classes



Altering metamodel

Adding structural classes

Dimension



Mereological Structures

- Lots of variety in the part-whole relationship
  - An engine is part of a car, a piston is part of an engine, so a piston is part of a car. Part-of transitive.
  - A player is part of a team, a team is part of a league, but a player is not part of a league. Part of here is intransitive.
  - An American presidential administration cannot exist without a President but it can exist without a Vice-President. Both President and Vice-President are parts of the administration.
  - A President cannot exist unless as part of an administration, but an ex-President can. Both President and ex-President are objects which can be parts, but the President depends on being a part while the ex-President doesn't.
  - An engine can be part of only one car, but a University course can be part of many program schedules.
  - A hole is part of a donut, but is not an object.
  - The US has 48 parts (states) which are geographically contiguous, but also two (Alaska and Hawaii) which are not.

Mereological Structures

- A material taxonomy of mereological structures
- Three boolean dimensions
  - Functional (role with respect to whole) handle of cup/ tree in forest
  - Homeomerous (similar to each other and the whole) grain – salt/ tree - forest
  - Separable (can be disconnected from the whole of which they are a part) tree-forest/ gin martini
- and five relations:
  - component/ integral object (handle/cup),
  - member/collection, portion/mass (grain/salt),
  - stuff/object (gin/martini),
  - feature/activity (paying/shopping),
  - place/area.
- But this taxonomy has a high ontological commitment

Mereological Structures

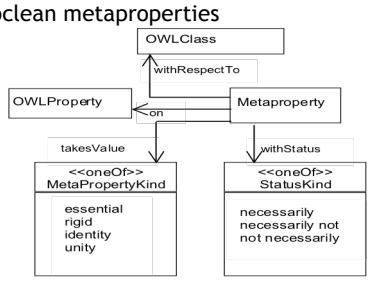
- A formal taxonomy with low commitment
  - Functional, inverse functional or non-functional.
    - Functional: an object may be a part of at most one whole (an engine of a car).
    - Inverse functional relationship: possibly several wholes have at most one of a part (in a parliamentary government, several ministries may be held by one minister)
    - Non-functional:an object may be a part of many wholes (an academic staff member and committees).
  - Non-functional can be transitive or intransitive. An engine is part of a car and a piston is part of an engine, so a piston is part of a car. A transitive property cannot be functional.
  - non-functional intransitive example: course/schedule, schedule/program rule. A course can be on many schedules, several program rules may share a schedule. But a course is not a program rule.

Mereological Structures

- More of formal taxonomy
  - A whole may depend on a part (a US presidential administration depends on the President) or may not (a car doesn't depend on a hubcap).
  - A part may depend on a whole (a competitor in a race depends on the race) or may not (a keyboard can exist apart from a computer).
  - Replacing a part may or may not result in a new instance of a whole. (Replacing a module in a software product results in a new version, replacing a player does not result in a new team.)
- Gives a total of 32 classes
- Properties in OWL are global
- So could define a package of properties modeling kinds of part-whole relationships

N-Ary Associations

- Ontoclean metaproperties



Extent-Descriptive Metaclasses

- Information about a world but not part of ontology
  - Size of populations
  - Specificity of attributes
    - Identifier maximal, essential property minimal
  - Likelihood of an optional property
  - Volatility
  - Default values
  - Etc.

Extent-Descriptive Metaclasses

- Model-driven architecture
  - Used to inform implementation design choices
- Query optimization
- Authoritativeness of elements of the world
  - All, or a selection?
  - Repository for creators, or a copy?
  - How current?
- Model in the same way as bulk/countable

Summary: Key Terms

- Widely-used structural features include: **countable/ bulk** classes, **concept/representation** classes, **dimension systems**, **mereological structures**, **metaproperties** and **extent-descriptive metaclasses**.All can be modelled using extensions to OWL.

Resources

- Essential
  - Notes chapter 12 Advanced Issues
- Further
  - Winston, M.E., Chaffin, R. and Herrmann, D. (1987) A taxonomy of part-whole relations Cognitive Science 11, 417-444 (available from Library site)