Common Errors In OWL

Alan Rector, Nick Drummond, Matthew Horridge,
Holger Knublauch, Jeremy Rogers, Robert Stevens,
Hai Wang, Chris Wroe

JISC

Introduction

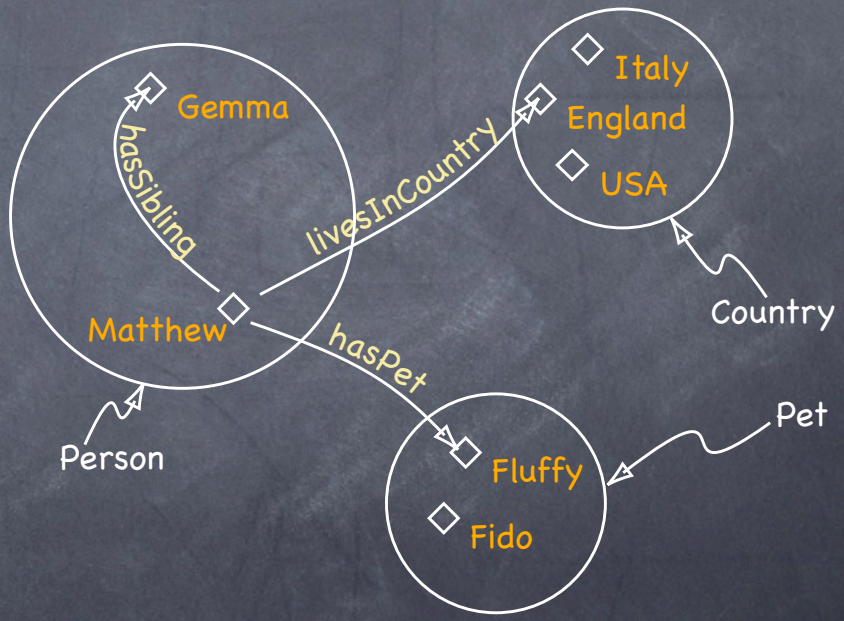
- The examples in this talk are based on courses about OWL that are taught at The University Of Manchester.
- Many newcomers to OWL make the same mistakes and incorrect assumptions about the language.
- Delivering courses on OWL has highlighted the pitfalls for new users.

What is OWL?

- The latest standard in ontology languages.
- Developed by the World Wide Web consortium (W3C).
- Based on RDF and DAML+OIL.
- Has formal mathematical foundations in Description Logics, which allows us to use a reasoner to help us to check the ontology as we build it.

Basic Elements Of OWL

- Individuals (instances)
- Properties (slots)
- Classes (concepts)

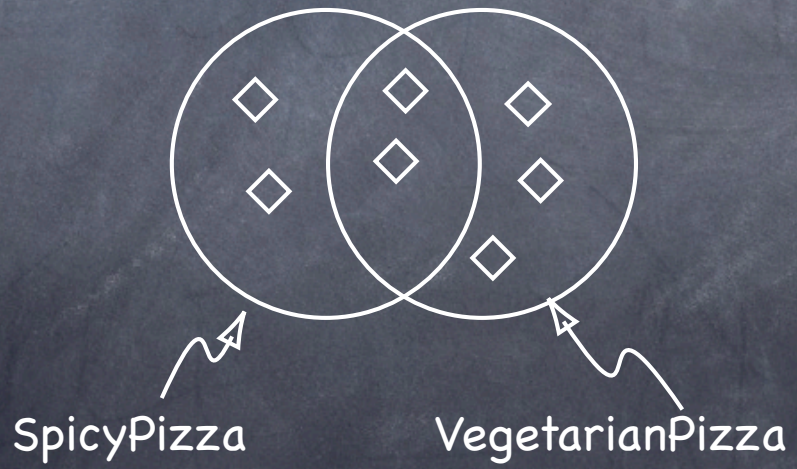


Common Mistakes

- Forgetting to make classes disjoint
- The mistaken use of universal rather than existential restrictions
- Open world reasoning
- Confusion about domain and range

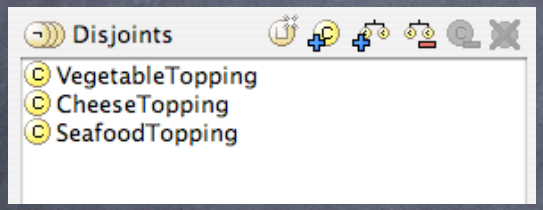
Disjoint Classes

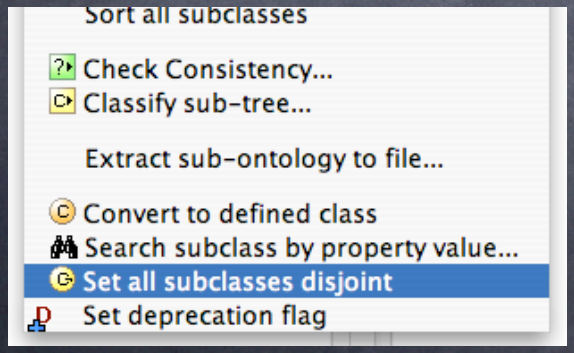
- OWL classes are assumed to overlap by default.
- For example, a SpicyPizza might also be a VegetarianPizza and vice versa.

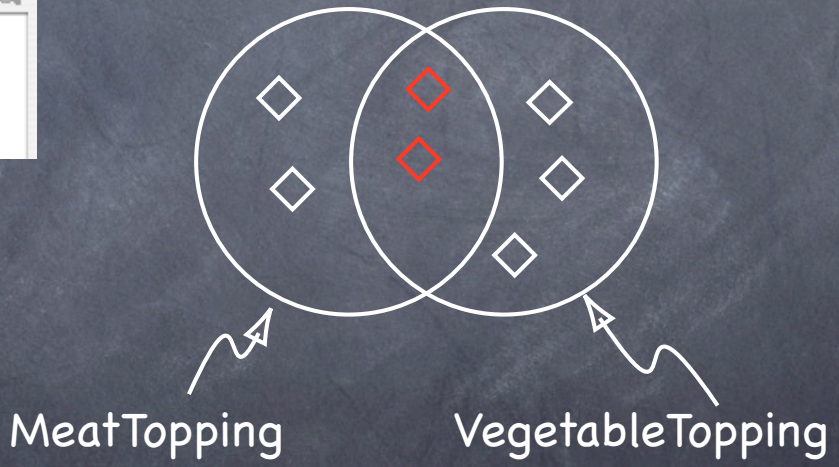


Disjoint Classes

- In situations where classes should not overlap, they must be explicitly made disjoint by the use of disjoint axioms







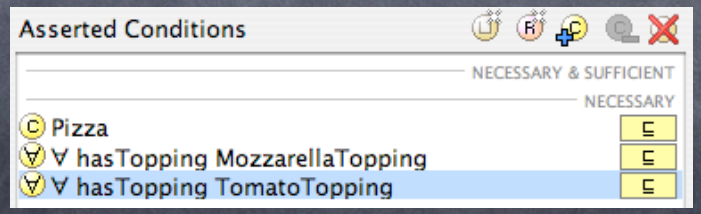
Restrictions

- Restrictions constrain the relationships between individuals.
- Many newcomers to OWL lean towards the use of universal (all values from) restrictions (∀).
- In general the 'default' type of restriction that should be used is an existential (some values from) restriction (∃).

Example

- Describe a Margherita Pizza using universal restrictions:

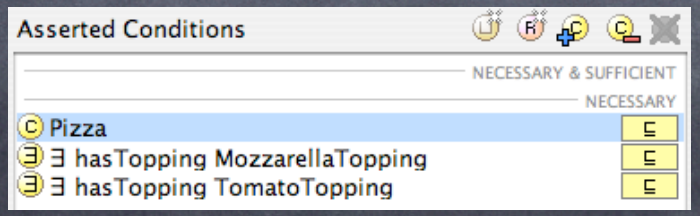
```
Class(MargheritaPizza
        Pizza
        restriction(hasTopping allValuesFrom(MozzarellaTopping))
        restriction(hasTopping allValuesFrom(TomatoTopping)))
```



Example

- Describe a Margherita Pizza using existential restrictions:

```
Class(MargheritaPizza
        Pizza
        restriction(hasTopping someValuesFrom(MozzarellaTopping))
        restriction(hasTopping someValuesFrom(TomatoTopping)))
```



Open World Reasoning

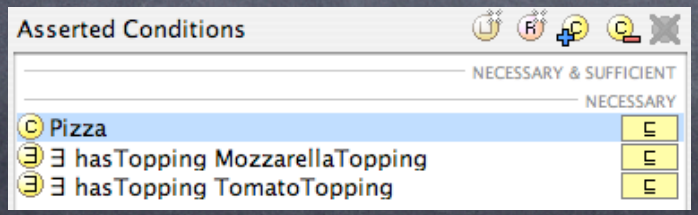
- OWL uses the Open World Assumption (OWA)
- Many OWL neophytes come from using closed world systems, such as databases.
- Information that has hasn't been explicitly added to a knowledge base is assumed to be 'missing' information, which could be added sometime in the future.
- How should we describe a Margherita Pizza?.....

A Margherita Pizza

(the intuitive way)

- Margherita Pizzas have toppings of Tomato and Mozzarella.

```
Class(MargheritaPizza
        Pizza
        restriction(hasTopping someValuesFrom(TomatoTopping))
        restriction(hasTopping someValuesFrom(MozzarellaTopping)))
```

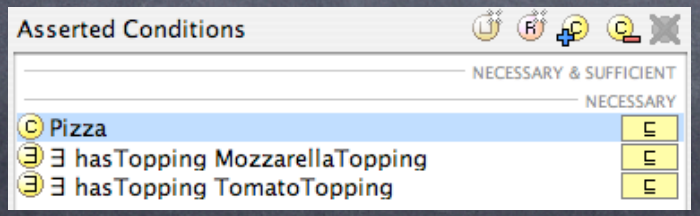


A Margherita Pizza

(the intuitive way)

- Margherita Pizzas have toppings of Tomato and Mozzarella

```
Class(MargheritaPizza
        Pizza
        restriction(hasTopping someValuesFrom(TomatoTopping))
        restriction(hasTopping someValuesFrom(MozzarellaTopping)))
```



What's wrong with this?

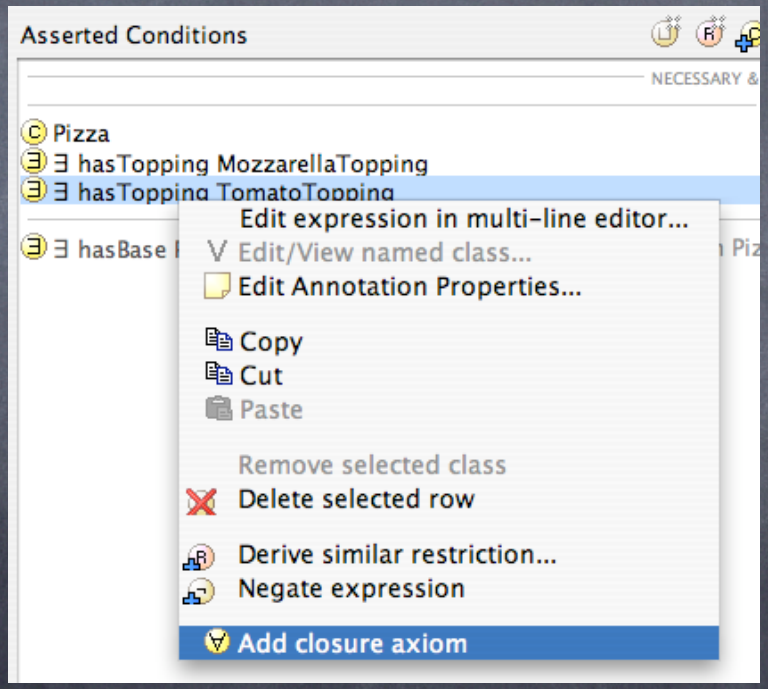
A Margherita Pizza

(the correct way)

- Margherita Pizzas have toppings of Tomato and Mozzarella - moreover, they only have toppings of Tomato and Mozzarella.

```
Class(MargheritaPizza
        Pizza
        restriction(hasTopping someValuesFrom(TomatoTopping))
        restriction(hasTopping someValuesFrom(MozzarellaTopping))
        restriction(hasTopping allValuesFrom(TomatoTopping or MozzarellaTopping)))
```

Creating Closure Axioms



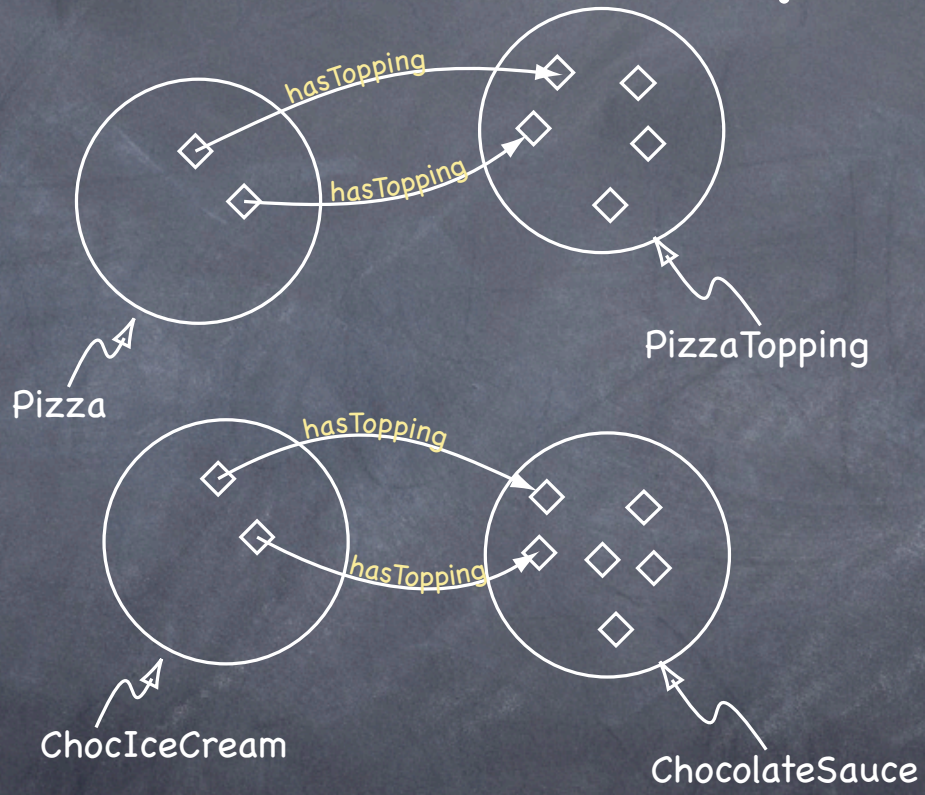
Domain and Range

- Domain and Range are a source of confusion for newcomers to OWL.
- Domain and range are not constraints to be checked. They are axioms which are used by the reasoner to make inferences.
- 'Violating' a domain or range constraint does not necessarily mean that the ontology is inconsistent or contains errors.

Domain Example

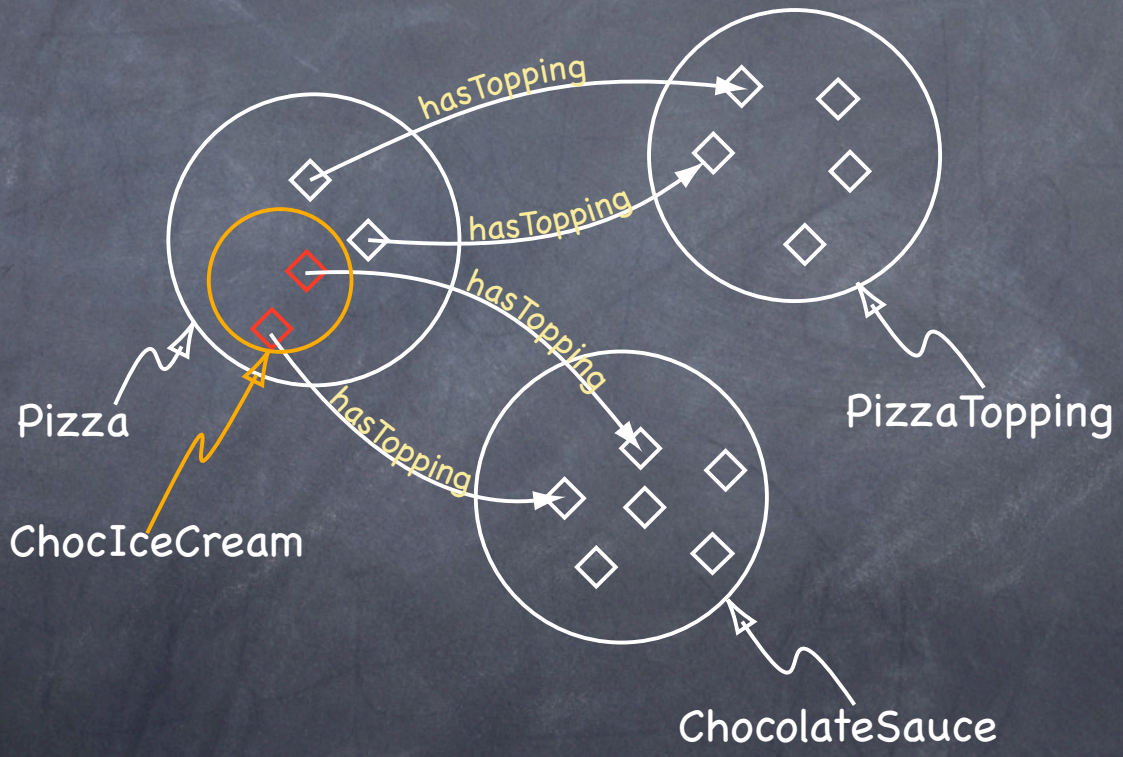
- Consider the hasTopping property to have a domain of the class Pizza.
- Now consider some individuals that are members of the class ChocIceCream which have toppings of ChocolateSauce.

Domain Example



What happens when we send this to the reasoner?

Domain Example



Conclusions

- Ensure that disjoint axioms are used correctly.
- The most common form of restrictions are existential restrictions. Use universal restrictions with caution.
- Remember that OWL uses the open world assumption. Descriptions of classes should be 'closed off' where appropriate.

Conclusions

- Domain and range axioms are often a source of confusion. They should be used with care as they can cause unexpected side effects.
- Using a reasoner can help in the detection of errors in the ontology, and ensure that the intended meaning of the ontology matches the logical meaning of the ontology.

Conclusions

- Protégé-OWL is positioned amongst the next generation of ontology tools.
- It has been designed to help minimise the errors that people often make.
  - It features a test frame work to help people catch errors and spot potential pitfalls early on.
  - Wizards and shortcuts help to speed up ontology development and reduce the opportunity to make mistakes by making the correct thing to do the easy thing to do.

Topics Not Covered

- Common logical issues - the linguistic verses that logical use of 'and' and 'or'.
- Difference between Primitive and Defined classes.
- Multiple inheritance - primitive concepts should ideally only have one parent concept.

Resources

- A Practical Guide To Building OWL Ontologies Using the Protege-OWL Plugin.
- OWL Pizzas: Practical Experience of Teaching OWL-DL: Common errors and common patterns. (Rector et al)
- http://protege.stanford.edu/plugins/owl/
- http://www.co-ode.org