Modeling Best Practices and Common Pitfalls

Protégé Short Course
March 29 - 31, 2017
Stanford University

Topics

- Open World vs. Closed World Assumption
- Unique Name Assumption
- How to pick an ontology?
- Entity Naming
- Class Hierarchies
- Primitive vs. Defined Classes
- Classes vs. Individuals
- Ontology Maintenance

Open World Assumption vs. Closed World Assumption

Closed World Assumption (CWA)

What is not known to be true, must be false

- Intuition: The system has complete information
- Common in databases
- Example:
  - We assert: *Whole Foods in Palo Alto sells Biscoff Cookies and Walkers Shortbread.*
  - Query: *Does Whole Foods in Palo Alto sell Cheerios*?
  - Answer: *No*

Open World Assumption (OWA)

What is not known to be true, is simply unknown

- Intuition: The system has incomplete information
- Example:
  - We assert: *Whole Foods in Palo Alto sells Biscoff Cookies and Walkers Shortbread.*
  - Query: *Does Whole Foods in Palo Alto sell Cheerios*?
  - Answer: *I don't know*

OWL uses the OWA

Open World Assumption (cont)

When expected inferences do not show up, think whether you have forgot about the OWA in your modeling!

Unique Name Assumption

Unique Name Assumption (UNA)

Different names represent distinct individuals

- Example (in a closed system):
  - We assert: *Italy, USA, and France are countries.*
  - Query: *How many countries are there?*
  - Answer: *3*

Unique Name Assumption (cont)

- OWL does not make the UNA. However, we can declare that the individuals are different.

- An OWL reasoner may make new inferences, rather than reporting a conflict.

Unique Name Assumption Example

We assert:

- *Whole Foods in Palo Alto sells only one product.*
- *Whole Foods in Palo Alto sells Cheerios.*
- *Whole Foods in Palo Alto sells Biscoff Cookies.*

- One would expect that this is a conflict, but!
- In OWL, *Cheerios* and *Biscoff Cookies* will be interpreted to be the same object, because OWL does not make the UNA.
- Therefore, make sure you declare that individuals are different entities, if it is important for your domain, or you expect conflicts as the one above to be reported.

How to pick an ontology (to use or reuse)

Questions to ask

- General review and rating
- Usage information
  - Which applications have successfully used the ontology?
  - What problems were encountered?
- Coverage
  - Does it cover the domain properly?
  - Are there major gaps?
  - Are some parts developed better than others?

Questions to ask (cont)

- Correctness
- Concept-specific comments
  - Are there problems with specific concepts?
  - What alternative definitions should be used?
- Mappings to other ontologies and standard terminologies
- Examples
  - What can and cannot be done with the ontology?
- Citations and references

Entity Naming

Naming Convention

- Decide on a convention and use it throughout the ontology
- Most ontologies use:
  - Camel case for names
  - Upper case for class names
  - Lowercase for property names
  - Lowercase for individual names (less established compared to above ones)

Naming Convention (cont)

- Singular names (*Product* class, not *Products* class), although singular can sound weird for some words (e.g., *oats*, *nuts*)

- Give unambiguous, context--free, names:
  - *Food Dish*, not *Dish*
  - *English Department*, not English as a subclass of *Department*

Naming Options

1. IDs: UUIDs and labels for readability:
   - ✚ anyone can generate them
   - ✚ renaming (the label) is not a problem
   - ━ hard to debug
   - ━ hard to use in applications
   - ━ hard to query
2. Structured codes:

   need a naming server if developed in multiple clients

   - ━ points (ii - v) from UUIDs

Naming Options (cont.)

3. Human readable names:
   - ➖ rename will break dependencies in other ontologies or applications using the ontology
   - ➕ easy to read
   - ➕ easy to query

Class Hierarchies

When creating a class hierarchy..

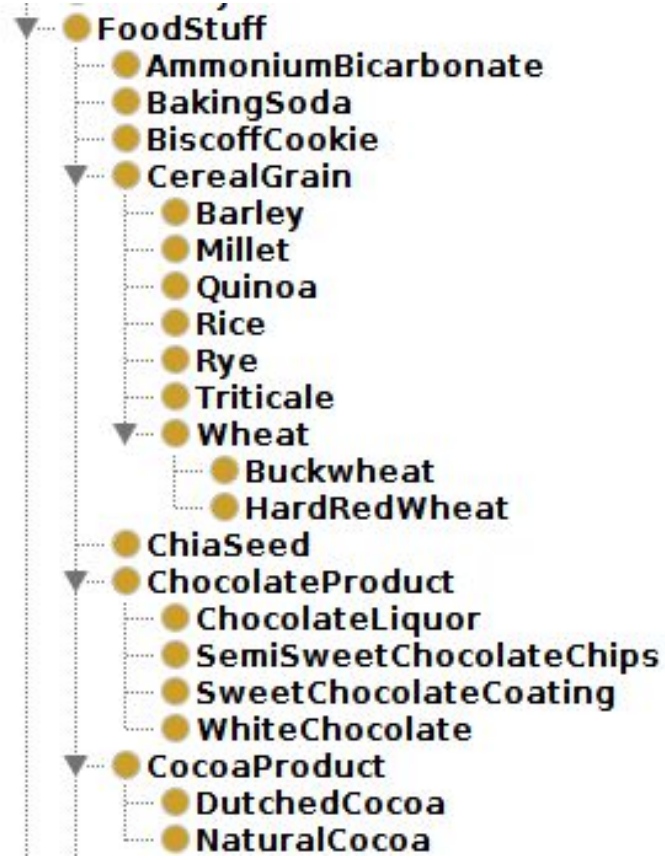
- The question to ask when building a hierarchy:
  - *Is each instance of the subclass an instance of its superclass?*

- The things to remember:
  - There is no single correct class hierarchy
  - But there are some guidelines

In OWL, let the reasoner do the work for you

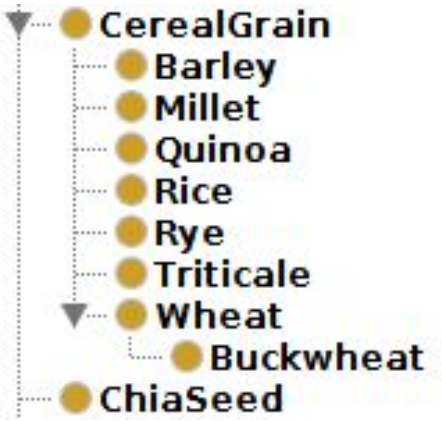
- Define some useful classes for your application/domain
  - *Meat Dish is defined a dish that contains meat*
  - Let the reasoner do the job for you, rather than you enumerate all the classes under all possible/useful classification criteria

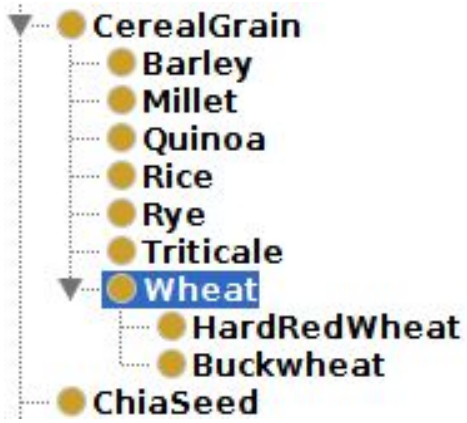
Siblings in the class hierarchy



- All the siblings in the class hierarchy must be at the same level of generality
- Compare to section and subsections in a book

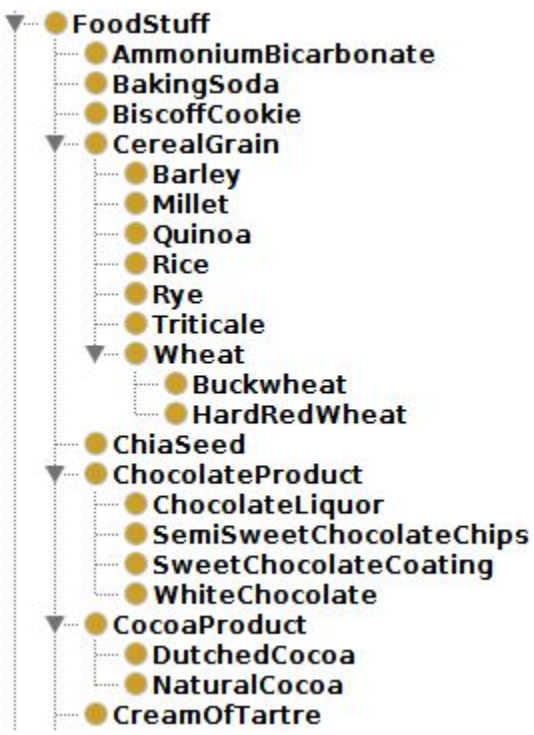
The perfect family size (I)





- If a class has only one child, there may be a modeling problem
- If the only Wheat we have is Buckwheat, why introduce the subhierarchy?
- Compare to bullets in a bulleted list

The perfect family size (II)



- If a class has more than a dozen children, additional subcategories may be necessary
- However, if no natural classification exists, the long list may be more natural

When to introduce a new class?

- Subclasses of a class usually have
  - Some kind of differentiation from superclass
  - Differentiation may be specified with additional axioms.

- But!
  - Differentiation between sub and superclasses may be left unspecified. For example, we do not specify the difference between many food stuffs.
  - Either the detail is not important, or we cannot precisely capture it.

Primitive vs. Defined Classes

Primitive and defined classes

- *Natural kinds* are difficult to give sufficient definitions
  - kinds of animals
- *Things that exist* vs. *Roles*
  - *Person* vs. *BookAuthor* (*BookAuthor* can be a defined class)
- Different views and perspectives (think about filtering)
  - Drugs classified by mechanism action

Classes vs. Individuals

Is it a class or an individual?

- Can it have kinds?
  - If so, it is a **Class**
  - Otherwise, it is an **Individual**

Examples:

- "*Kinds of dog*" makes sense
- "*Kinds of cookies*" makes sense
- "*Kinds of Bill Clinton*" does not make sense
- "*Kinds of bacon*" makes sense
- "K*inds of the one piece of bacon in this morning's breakfast*" does not make sense

Slide inspired by: A Practical Introduction to Ontologies & OWL, Duncan Hull and Nick Drummond

Classes vs. Individuals

- When you say something about it...
  - ...if you made a new concept, it is a **Class**
  - ...if you just stated a fact about it, it is an **Individual**

Examples:

- "*Big dog*" is a new subclass of dog
- "*Rover is big*" just says something about Rover
- "*Mouldy Pizza Ingredient*" is a new subclass of Pizza Ingredient
- "*The mouldy bit of cheese on the pizza I bought last Friday*" is a fact about the bit of cheese in your bad pizza experience

Slide inspired by: A Practical Introduction to Ontologies & OWL, Duncan Hull and Nick Drummond

Clues in English

- Proper nouns (almost always) indicate individuals
  - *Michelle Obama, Palo Alto, Italy*, …
- Articles + singular indicate individual
  - "*the book there on the shelf*" – an individual
  - "*a book*" – an unspecified individual
- Plurals usually indicate classes
  - "*the books*" – probably a class (Although possibly an individual aggregation)
  - Perversely, we conventionally name classes in the singular

Slide inspired by: A Practical Introduction to Ontologies & OWL, Duncan Hull and Nick Drummond

More clues in English..

- A *'...that...'* clause usually indicates a class
  - *"The people that drive buses"*
- 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'
- No perfect guide, must take case by case.

Slide inspired by: A Practical Introduction to Ontologies & OWL, Duncan Hull and Nick Drummond

Ontology Maintenance

You built it, now maintain it..

- Documentation
  - Always write human readable definition
  - Document design decisions
- Define the workflow
  - Who is doing what
  - Collaboration
  - Release cycles
- Assure quality
  - Run reasoner frequently
  - Design test cases