Ontology 101:
The basics of developing ontologies

Tania Tudorache

Protégé Short Course
Stanford University
March 29, 2017

Overview

- Introduction
- Ontologies
  - What is an ontology
  - Step-by-step ontology development
- Further considerations

GOOGLE

who got the chemistry nobel prize in 2014

About 31,800,000 results (0.36 seconds)

Eric Betzig

The Nobel Prize in Chemistry 2014 was awarded jointly to **Eric Betzig**, **Stefan W. Hell** and **William E. Moerner** "for the development of super-resolved fluorescence microscopy".

The Nobel Prize in Chemistry 2014
www.**nobelprize**.org/**nobel_prizes/chemistry/laureates/2014/** Nobel Prize

Born: January 13, 1960 (age 55), Ann Arbor, MI

Influenced by: William E. Moerner

Spouse: Ji Na

Parents: Robert Betzig

Institution: Howard Hughes Medical Institute

Siblings: Laura Betzig

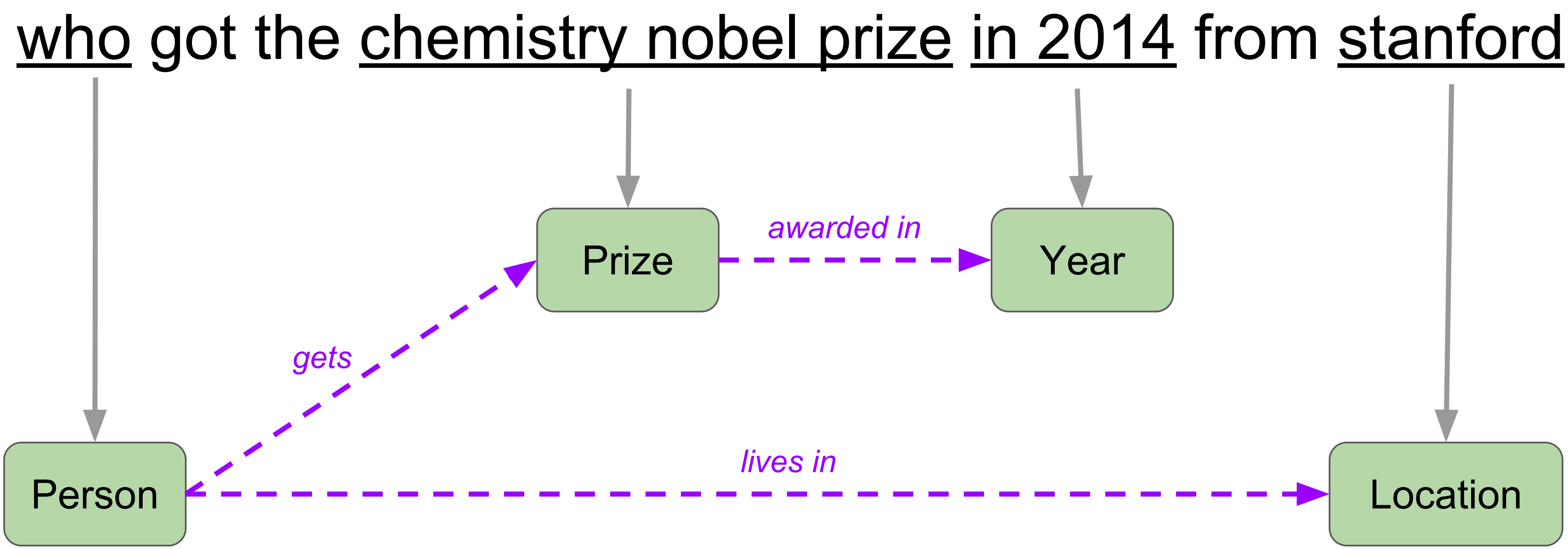
who got the chemistry nobel prize in 2014 from stanford

About 522,000 results (0.39 seconds)

Moerner, the Harry S. Mosher Professor of Chemistry at Stanford, won the 2014 Nobel Prize in Chemistry. He shares the prize with **Eric Betzig**, of Howard Hughes Medical Institute, and **Stefan W. Hell**, of the Max Planck Institute for Biophysical Chemistry, in Germany.

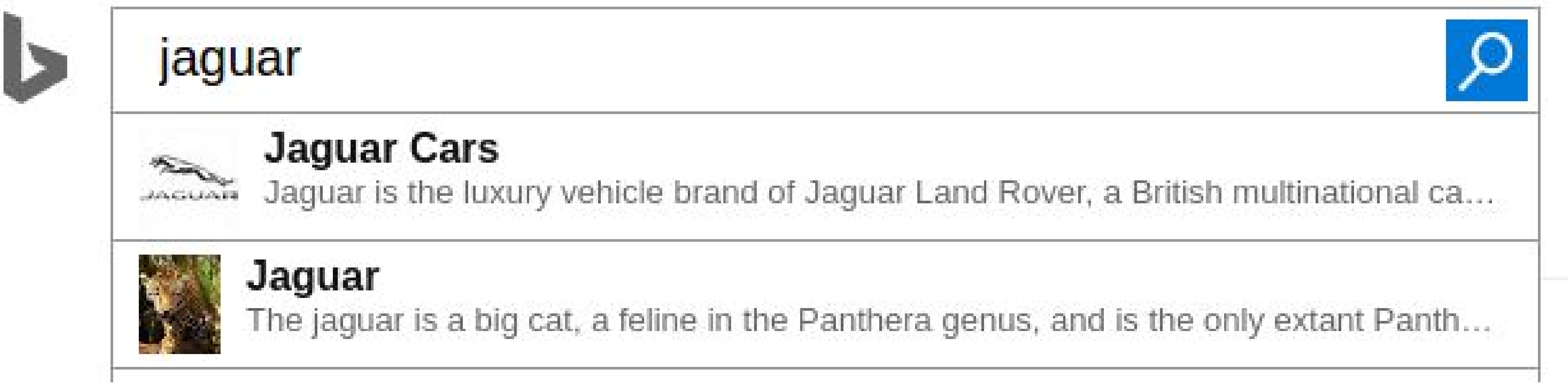
Nobel Laureates | Stanford University - Stanford News
news.**stanford**.edu/**nobel**/

Feedback

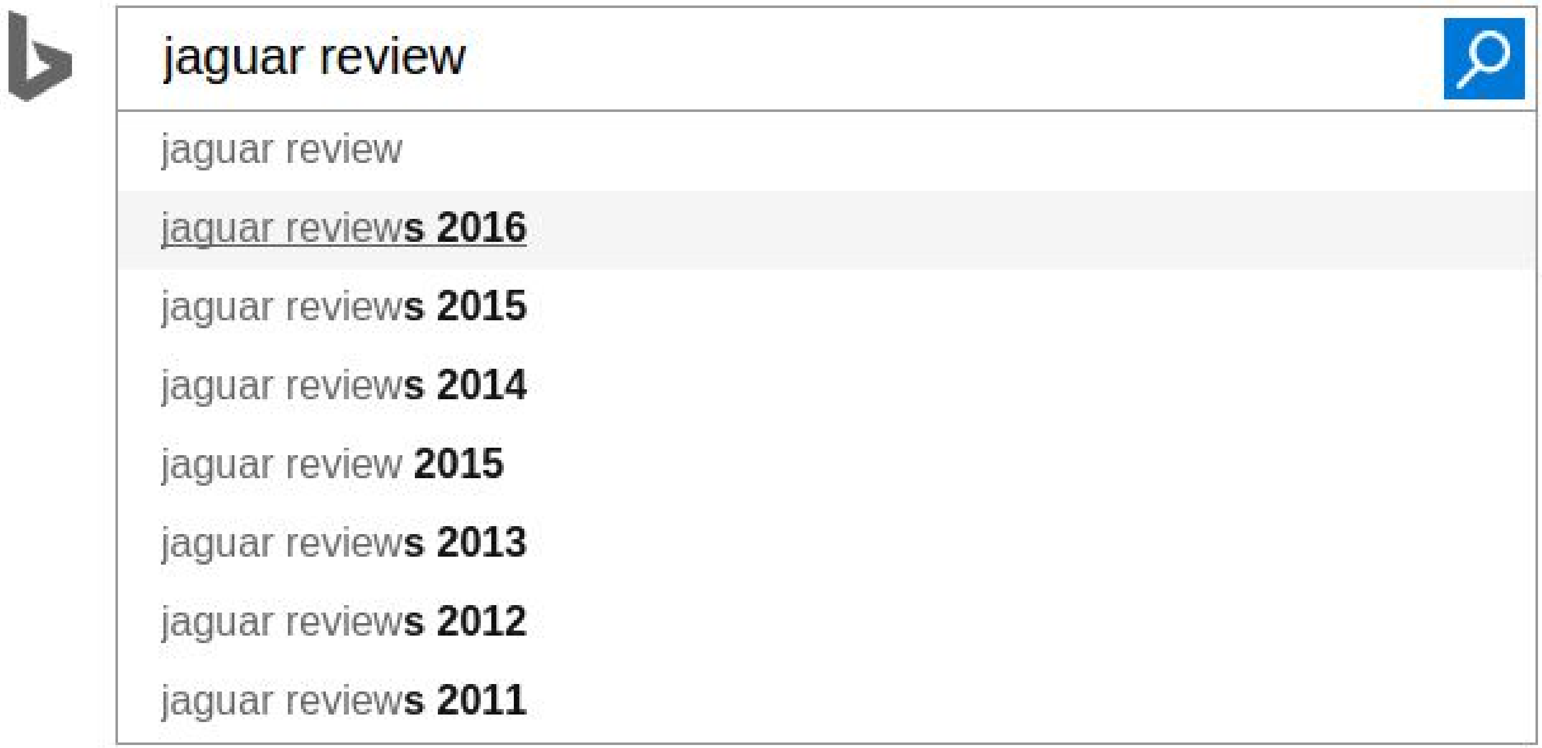


(over-simplified)

Search engines can do disambiguation

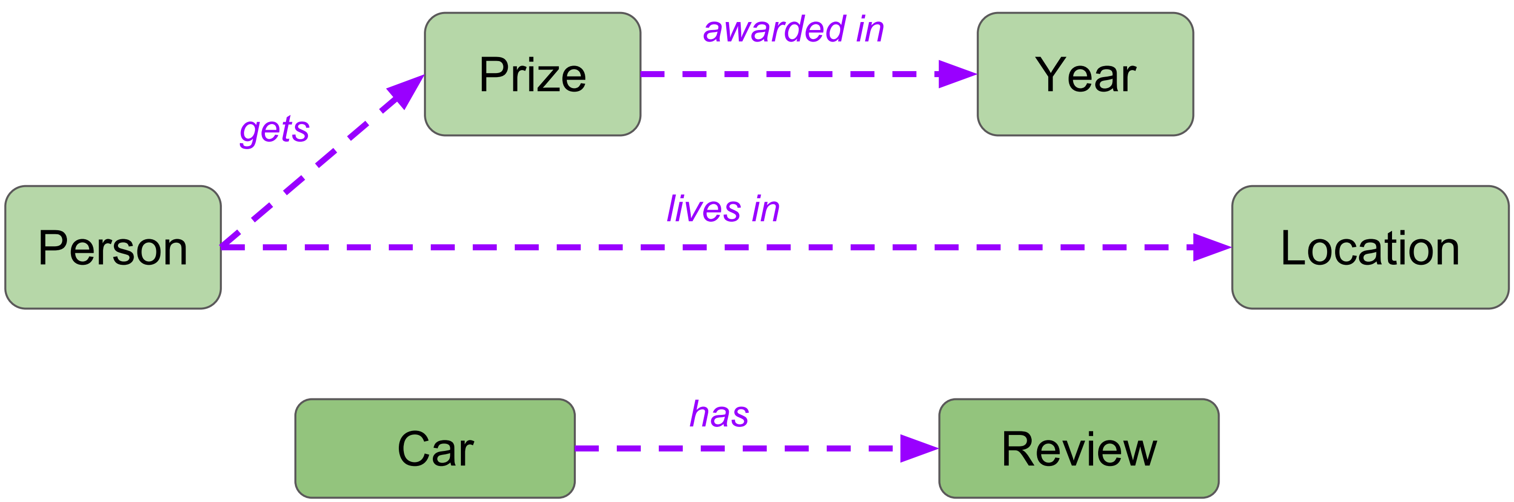


Search for “jaguar” returns both the car brand and the animal.



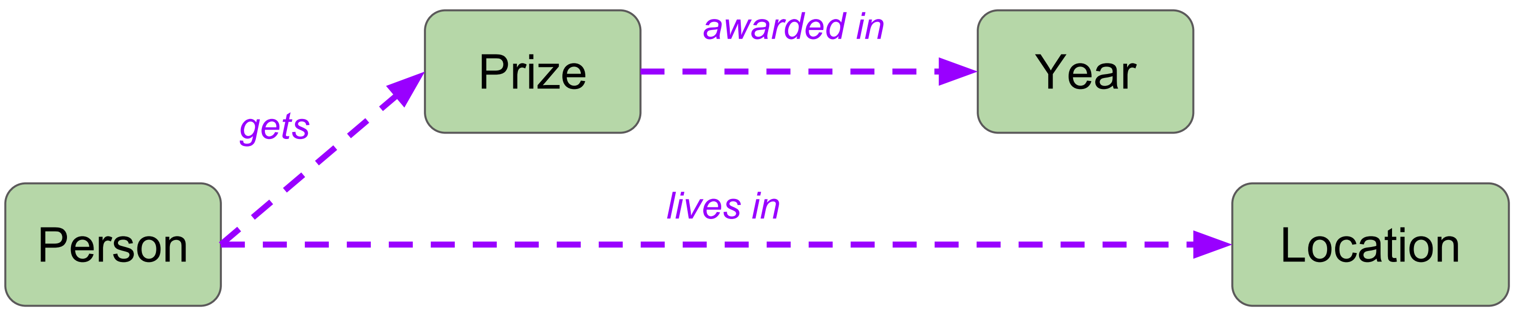
The search engine knows that "*review*" only applies to the "*car*" not to the "*animal*".

The machine has a representation of the world and it can operate on it.



Real world

Computer representation



Upper image source: https://eandscaltech.files.wordpress.com/2015/02/am_bretzig_prizeceremony_13.jpg

Overview

- Introduction
- Ontologies
  - What is an ontology
  - Step-by-step ontology development
- Further considerations

What is an ontology?

- Origin of the word: A branch of philosophy concerned with "*that which exists*"; that is, a description of the things in the world.

- Computer scientists: *a shared understanding or specification of the concepts of interest in a domain* of information that can be used by both computer and humans to describe and process that information. The goal with a computer science ontology is **to make knowledge of a domain computationally useful**.

Source: http://www.cs.man.ac.uk/~stevensr/menupages/background.php

What is an ontology? (cont.)

Most famous definition (Tom Gruber, 1992):

An ontology is a specification of a conceptualization.

An updated version:

http://tomgruber.org/writing/ontology-definition-2007.htm

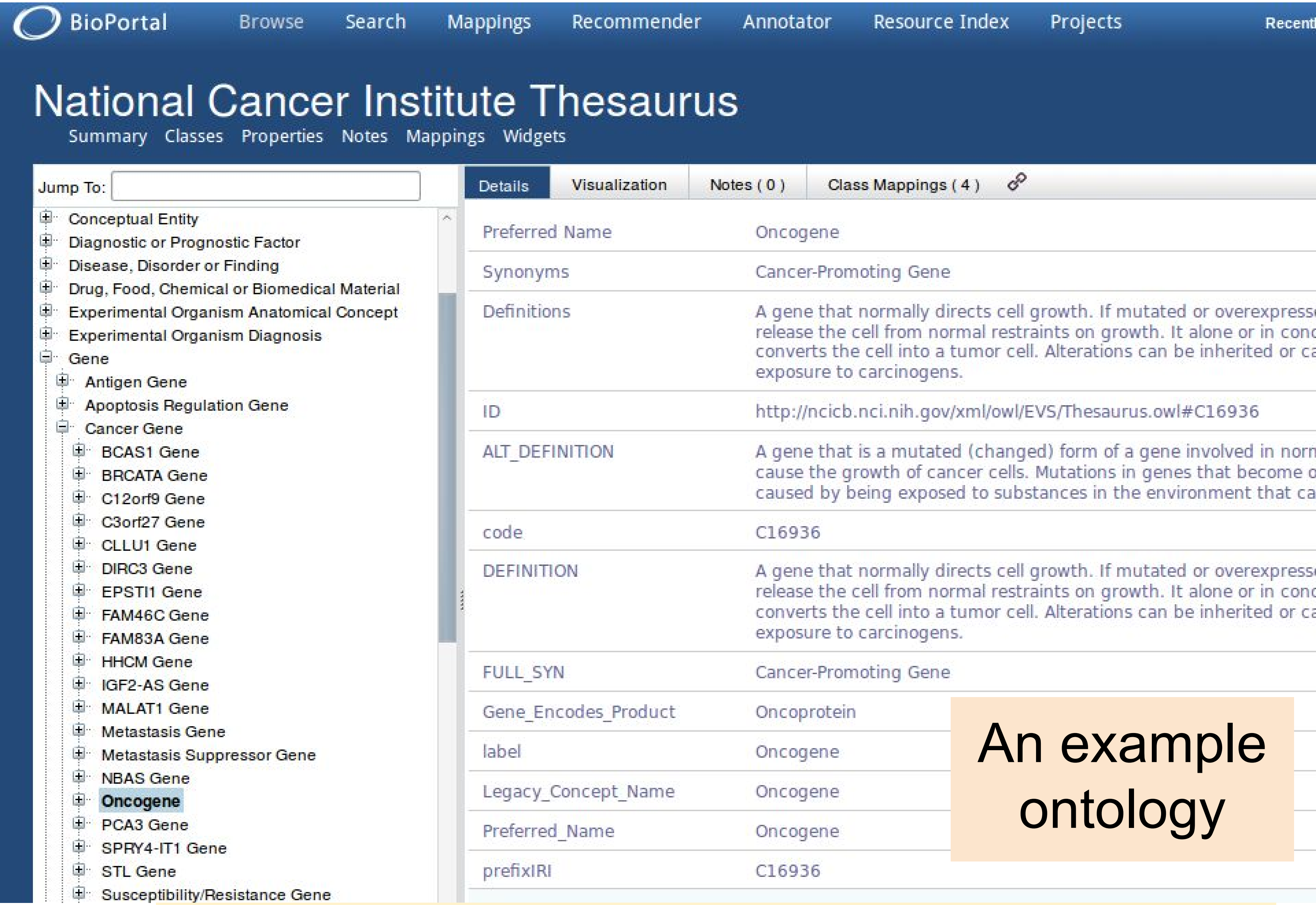
Gruber, 1992: http://www-ksl.stanford.edu/kst/what-is-an-ontology.html

What is an ontology

- An ontology is an explicit description of a domain:
  - concepts
  - properties and attributes of concepts
  - constraints on properties and attributes
  - individuals
- An ontology defines
  - a common vocabulary
  - a shared understanding

An Ontology

- Contains explicit definition of terms used by agents
- Provides unambiguous interpretation of terms
  - expresses semantics of constructs in a formal language (e.g., OWL)
- Intended to be shared among agents
  - emphasis on sharing in ontology development
- Developed for use with reasoning agents
  - used to infer new facts from existing definitions



An example ontology

http://bioportal.bioontology.org/ontologies/NCIT/

Why develop an ontology?

- To share common understanding of the structure of information
  - among people
  - among software agents

- To enable reuse of domain knowledge to avoid "re-inventing the wheel" to introduce standards

More reasons

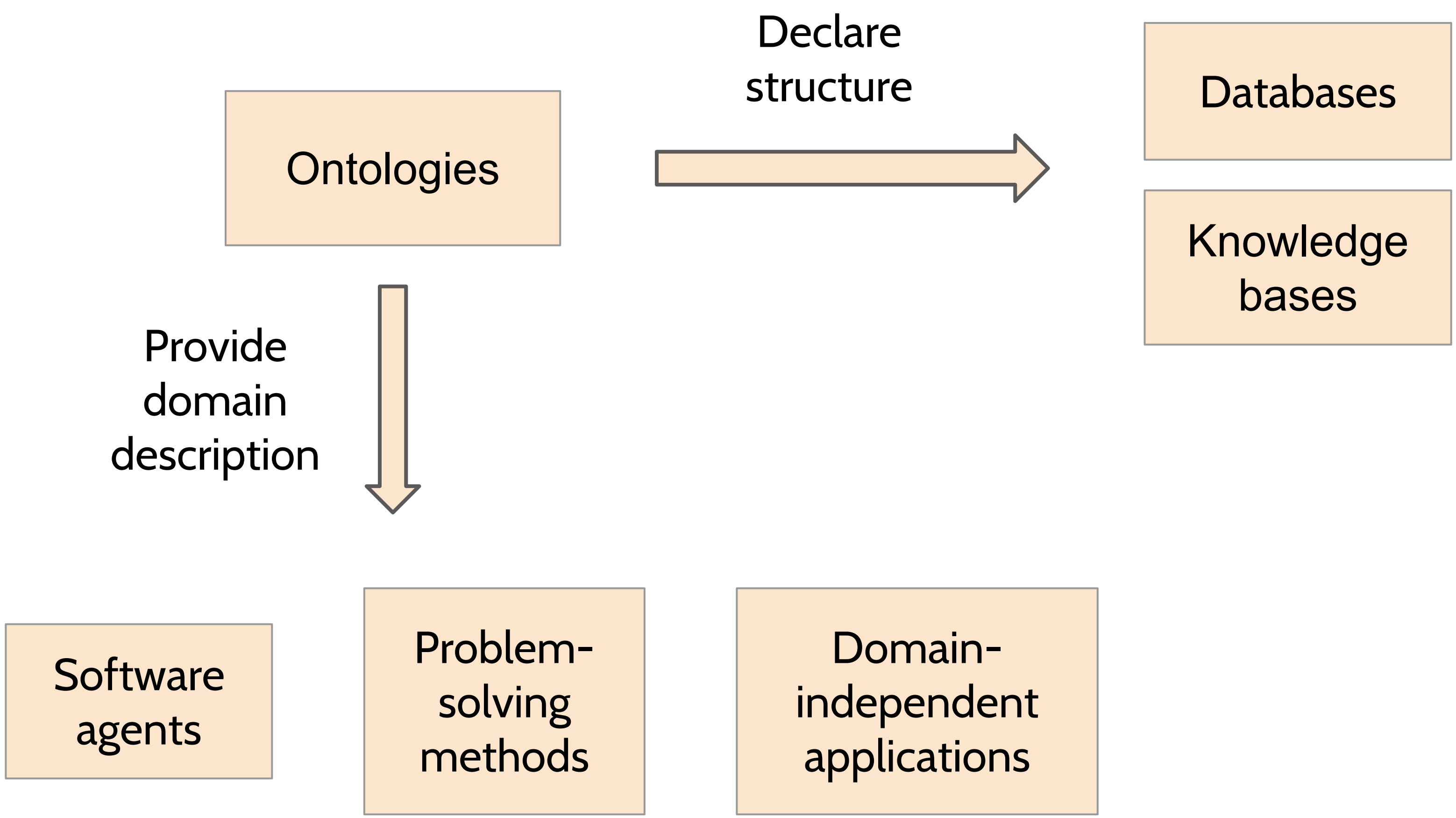
- To make domain assumptions explicit
  - easier to change domain assumptions
  - easier to understand and update legacy data

- To separate domain knowledge from the operational knowledge
  - re-use domain and operational knowledge separately (e.g., configuration based on constraints)

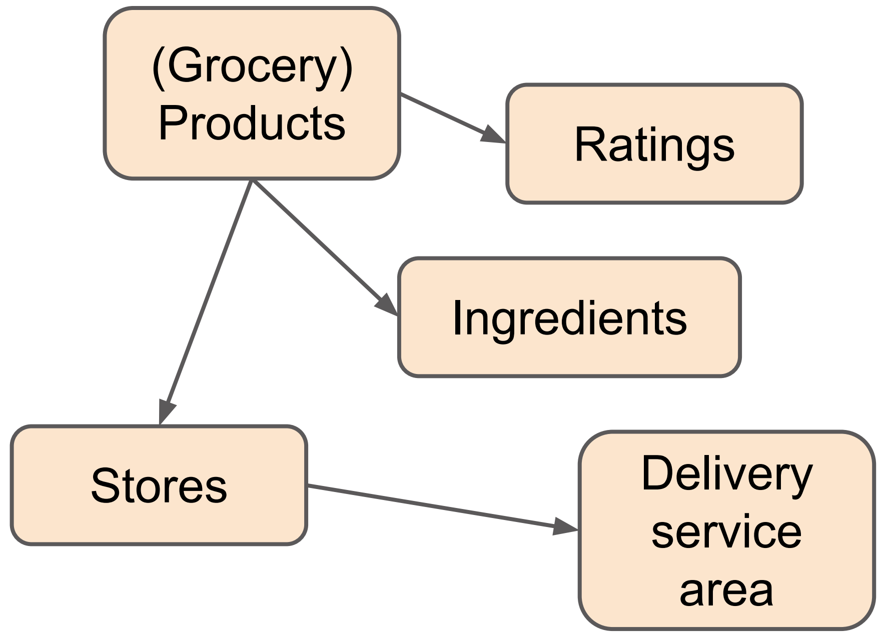
How are ontologies used?

- As shared, “standard” vocabularies for a domain, e.g., NCI Thesaurus
- As a way to integrate heterogeneous data sources
- For annotating data with standardized terms
- Faceted browsing
- For decision support

An ontology is often just the beginning



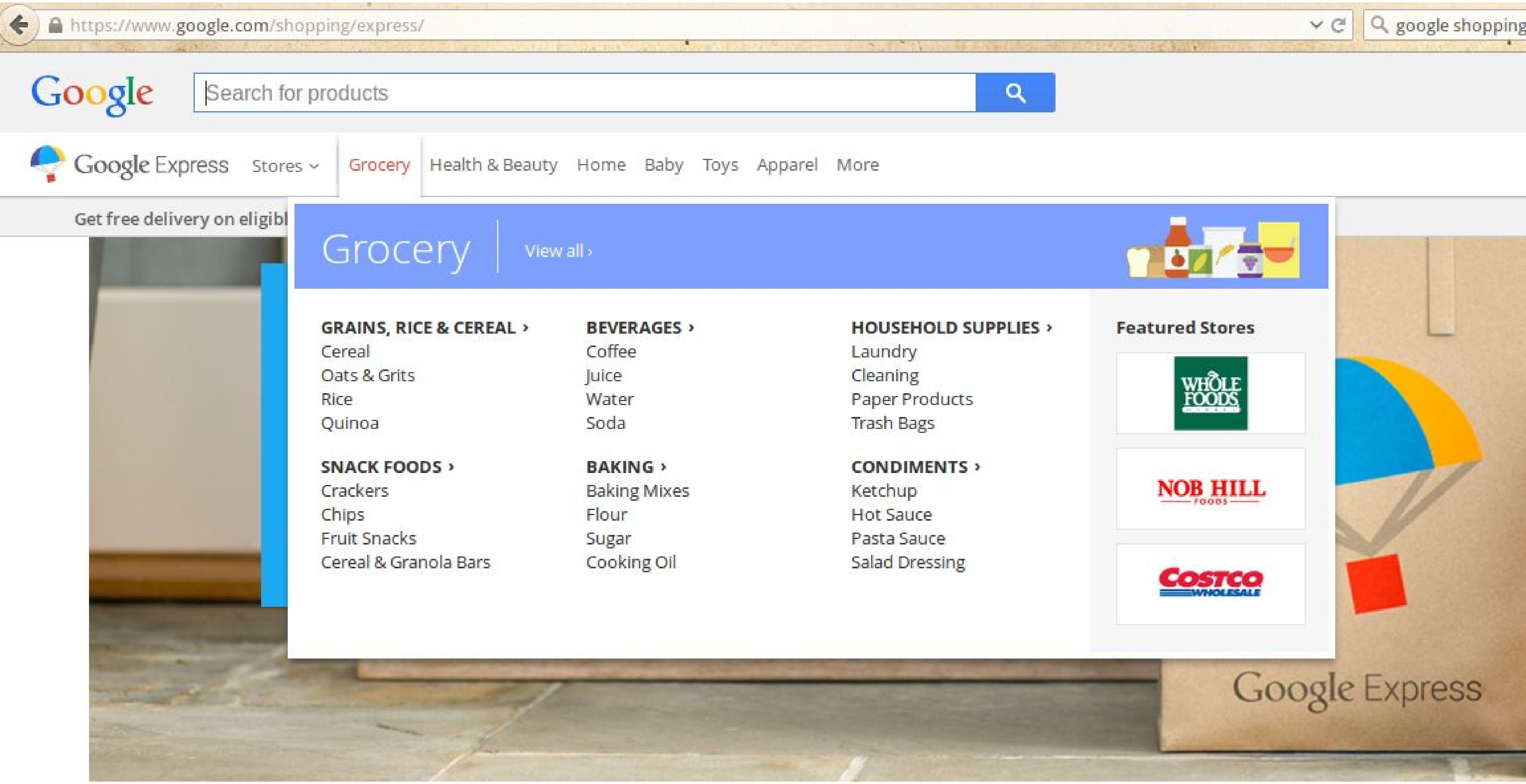
Running Example for the course - shopping express



Design an intelligent Web shop that allows customers to search and filter the products according to complex criteria

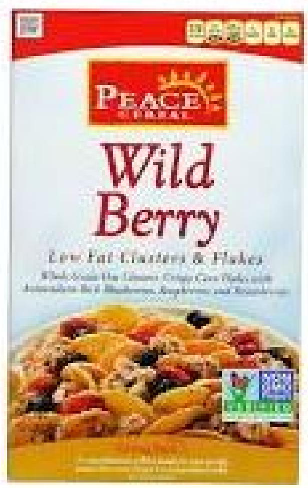
Which stores around my zip code have bananas on stock?

Inspiration from Google Shopping Express



Inspiration from Google Shopping Express (cont.)

Q: Cereals with low salt



Peace All Natural Clusters & Flakes Cereal, Low Fat, Wild Berry - 10 oz box

$3.99

★★★★★ 3 reviews

NOB HILL FOODS

More stores

Nutritional information

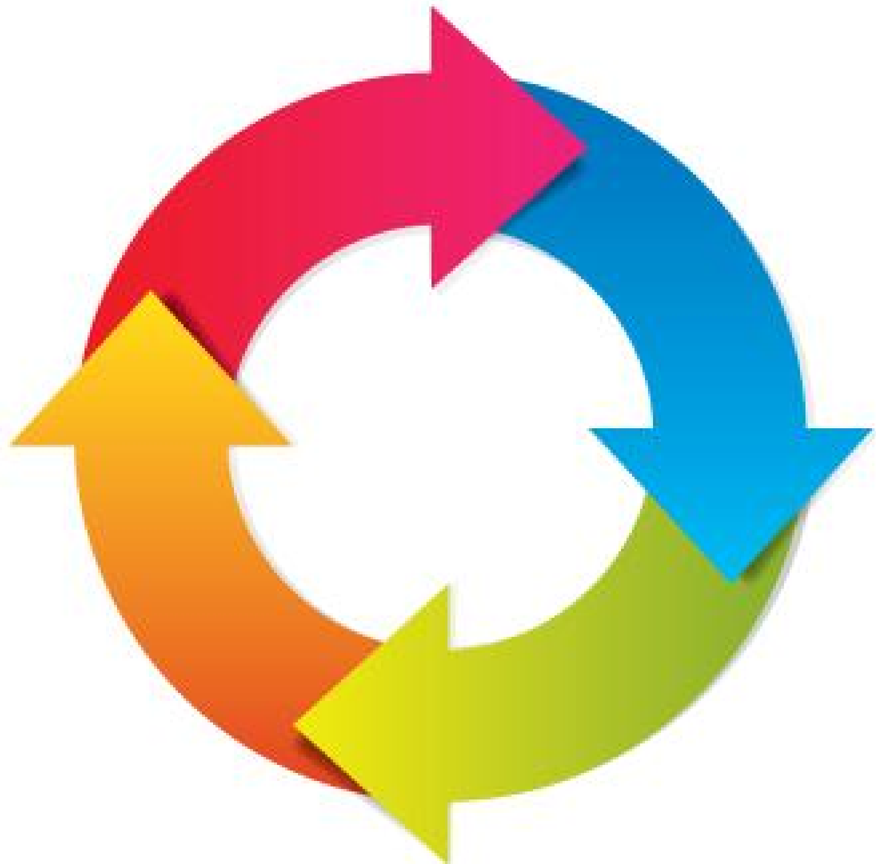
Serving Size	1 cup
Servings Per Container	5
Total Fat	3 g (5 %)
Saturated Fat	0 g (0 %)
Cholesterol	0 mg (0 %)
Sodium	330 mg (14 %)
Total Carbohydrate	44 g (15 %)
Dietary Fiber	3 g (12 %)
Sugars	7 g
Protein	5 g
Vitamin A	2 %
Vitamin C	2 %
Calcium	2 %
Iron	6 %

Overview

- Introduction
- Ontologies
  - What is an ontology
  - Step-by-step ontology development
- Further considerations

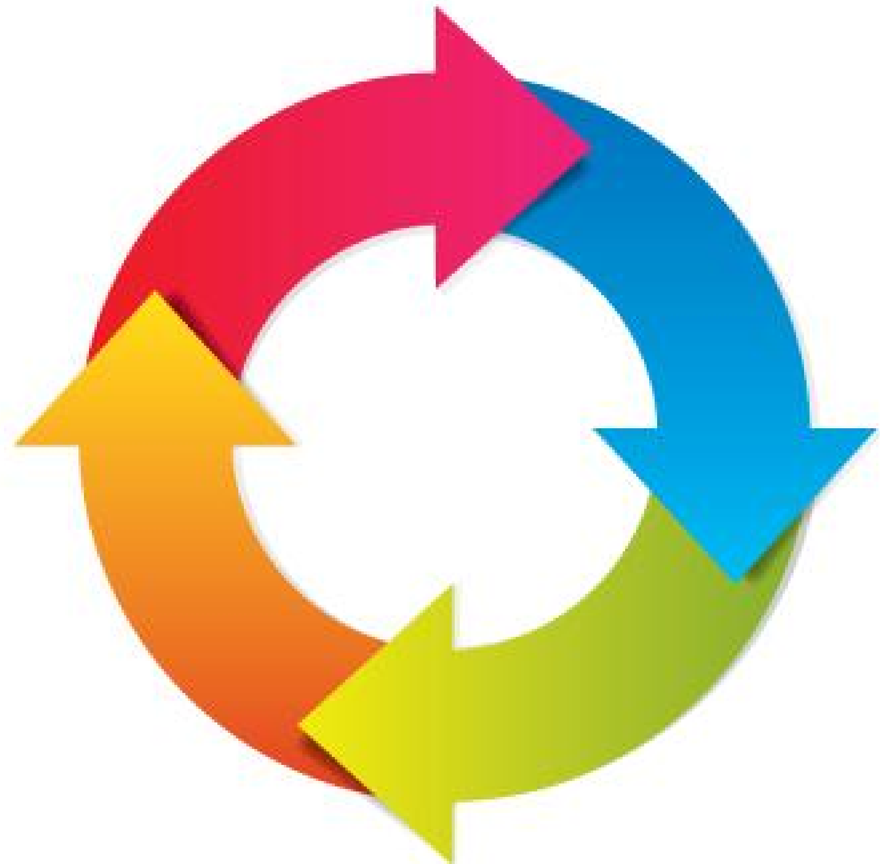
Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- Define classes
- Define properties
- Define individuals
- Test



Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- Define classes
- Define properties
- Define individuals
- Test



Determine the domain and scope

- What is the domain that the ontology will cover?
- What are we going to use the ontology for?
- What types of questions is the ontology supposed to answer?
- Who will use and maintain the ontology?

Answers to these questions may change during the ontology lifecycle

Determine the domain and scope (cont.)

- For our example:
  - the shopping domain
  - focus on groceries
  - trying to answer queries and filter grocery products based on some criteria

Determine the domain and scope (cont.)

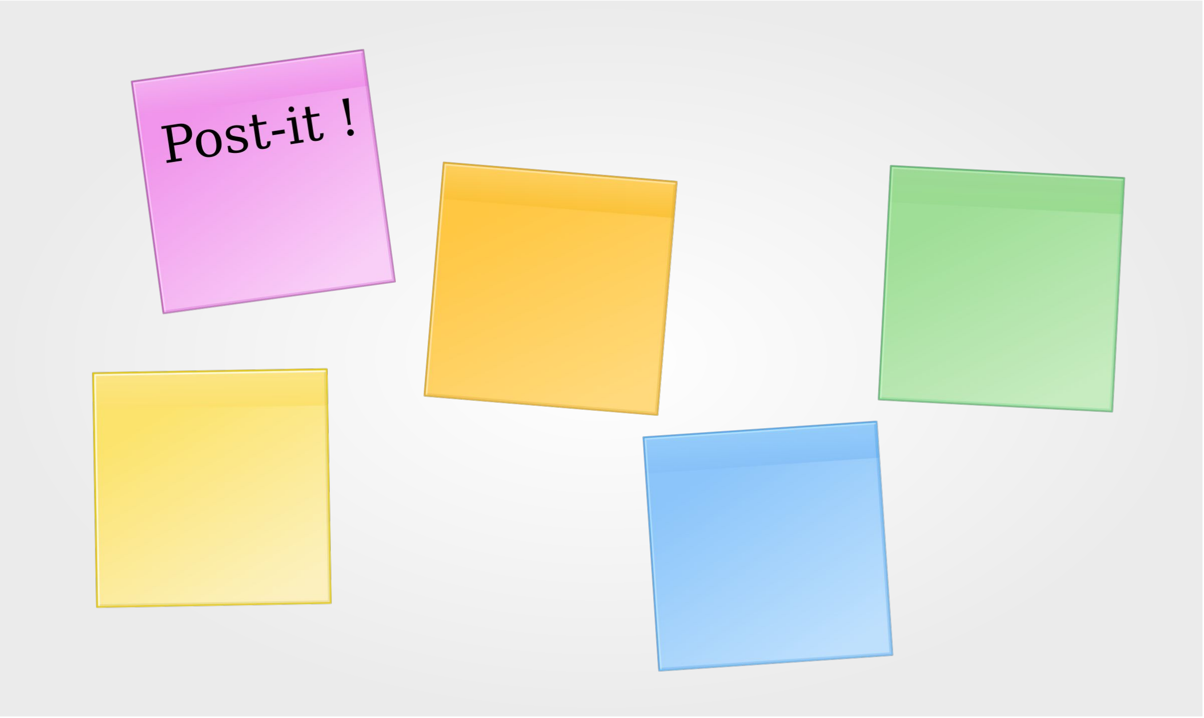
- Competency questions are questions that the ontology should be able to answer
- Helpful for:
  - defining the scope of the ontology
  - defining the vocabulary (main terms) of the ontology
  - testing the ontology
- Example:
  - *Which cereals have low salt?*
  - *What is the serving size for Cheerios?*
  - *Are there any children's multi-vitamins that don't contain glucose syrup?*

Hands-on: Competency questions for the shopping domain

- Imagine you are the customer of this Web shop. What kind of queries would you like to ask?
- Try to come up with several competency questions for the grocery domain and write each on a post-it note
- Try to focus on the grocery domain
- Use the handouts as examples
- You may also get inspiration from the Google Shopping Express page
- 15 minutes

Hands-on: Competency questions for the shopping domain

- We will discuss the competency questions
- We will try to organize them



Some competency questions

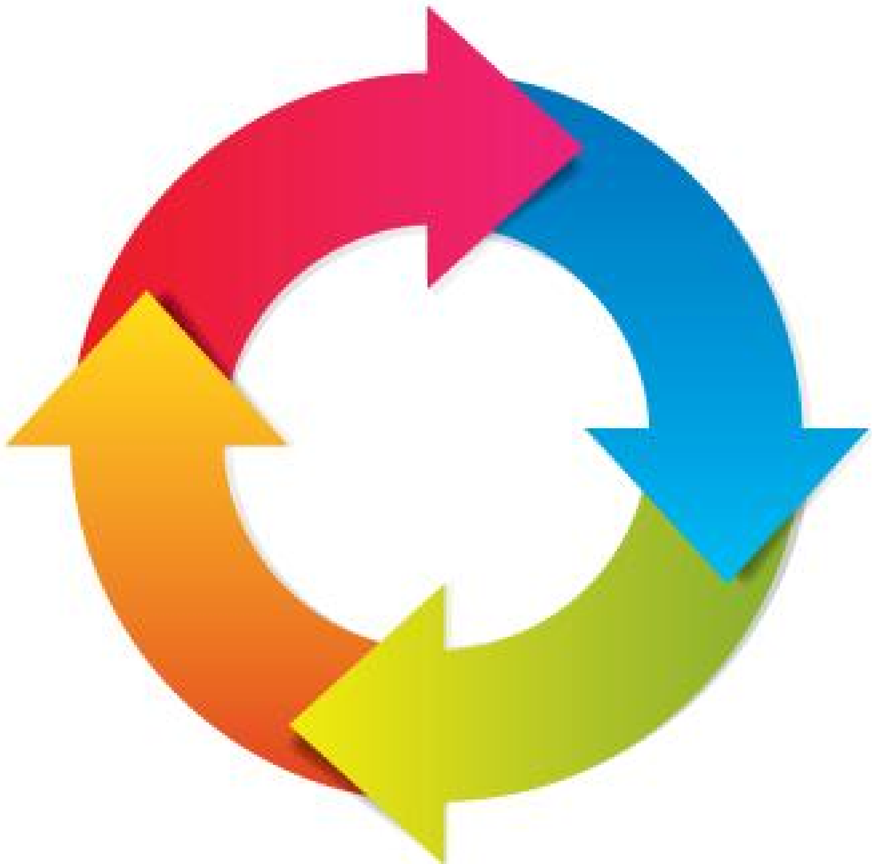
- What beverages are sugar-free?
- What are the ingredients of a product?
- What baked goods (such as cakes) do not contain canola oil?
- Does this product contain soy?
- Which supplements are vegetarian?
- What types of gluten free breads are there?
- Who is the producer of a certain product?
- Is this product high in salt?
- How many calories does a serving of this product contain?
- What kind of gluten free biscuits do you have?
- Does this product contain nuts (or traces of nuts)?
- Where can I buy this product from?
- Is this product sold from WholeFoods in Palo Alto?

Some considerations for the competency questions

- Competency questions can have different forms:
  - Yes/No questions: "*Is this product high in salt?*"
  - "Wh" questions: "*Which supplements are vegetarian?*"
- Try to formulate questions that can be formalized:
  - *"What beverages are sugar-free?*"
- Try to cover different areas of the domain
- Have a brainstorm session with the domain experts to come up with these questions
- At the end of the session, group the questions, and document them
- Create test cases based on these questions, and use them throughout the development

Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- Define classes
- Define properties
- Define individuals
- Test



Consider reuse

- Why reuse other ontologies?
  - to save the effort
  - to interact with the tools that use other ontologies
  - to use ontologies that have been validated through use in applications

Consider reuse (cont.)

Two types of reuse:

- Reuse of an ontology
- Reuse of specific parts of an ontology:
  - individual terms
  - a branch (structural)
  - a logical module (semantic)

What to reuse

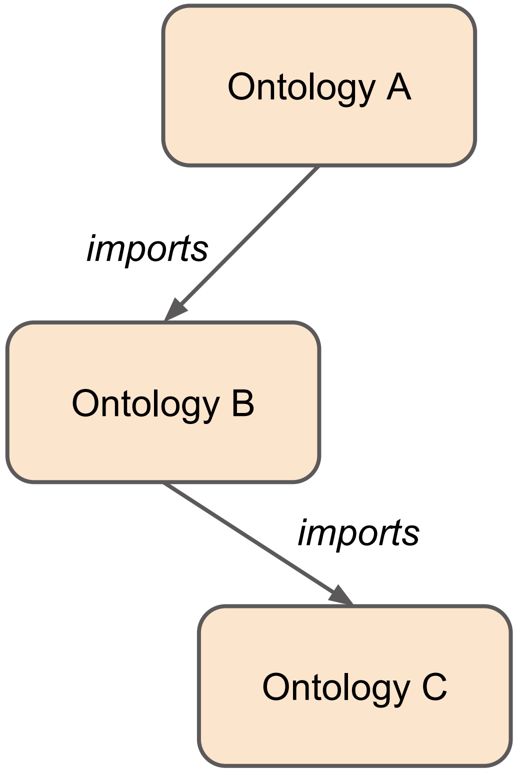
- Ontology libraries
  - Biomedical Ontologies: NCBO Bioportal (http://bioportal.bioontology.org)
  - Ontologies in various representation languages: Ontohub (https://ontohub.org/)
  - Linked Open Vocabularies (http://lov.okfn.org/dataset/lov)
  - The Manchester OWL Corpus - MOWL Corp (http://mowlrepo.cs.manchester.ac.uk/datasets/mowlcorp/)
  - The TONES ontology repository (http://rpc295.cs.man.ac.uk:8080/repository/)
  - Protégé ontology library: (http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library)
  - WebProtégé (http://webprotege.stanford.edu)

What to reuse (cont.)

- Semantic Web Search engines:
  - Swoogle (http://swoogle.umbc.edu/)
  - Falcons (http://ws.nju.edu.cn/falcons)
  - Watson (http://watson.kmi.open.ac.uk/WatsonWUI/)
- Useful indexes:
  - http://vocamp.org/wiki/Where_to_find_vocabularies
  - http://swl.slis.indiana.edu/repository/
  - http://www.w3.org/wiki/VocabularyMarket

Reuse of an entire ontology

- OWL provides a mechanism for importing an ontology into another one, through the *owl:imports* statement
- The import is by URI (points to the location of the ontology to import)
- The imports statement is transitive



References: http://www.w3.org/TR/owl-ref/#imports-def
http://protegewiki.stanford.edu/wiki/Importing_Ontologies_in_P41
http://protegewiki.stanford.edu/wiki/How_Owl_2.0_Imports_Work

Reuse of parts of an ontology

- If reusing single terms, you can use the IRI (identifier) of an entity, e.g.:

  The class *Gene* in *NCI Thesaurus* is identified by:

  http://ncicb.nci.nih.gov/xml/owl/EVS/Thesaurus.owl#C16612

- Tools you may use:
  - The ProtegeLOV plugin: http://boris.villazon.terrazas.name/projects/prolov/index.html
  - The MIREOT Protégé plugin:

    https://code.google.com/p/mireot-protege-plugin/

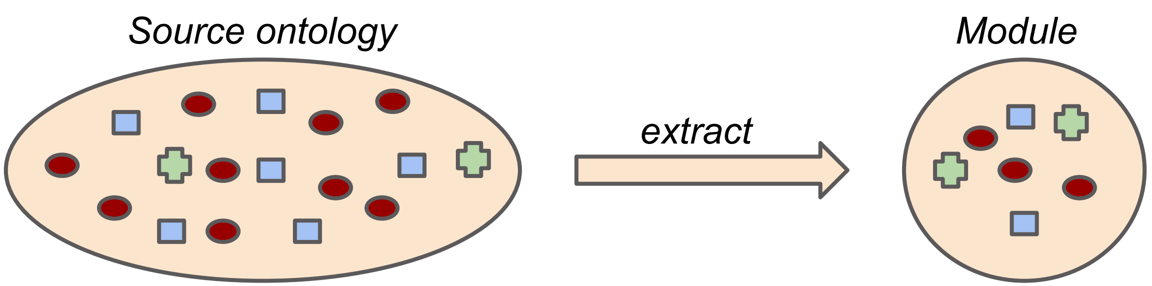
  - WebProtege BioPortal import portlet:

    http://webprotege.stanford.edu

Reuse of parts of an ontology (cont.)

- Module extractors allow users to extract modules that represent (only) the knowledge one wants to reuse

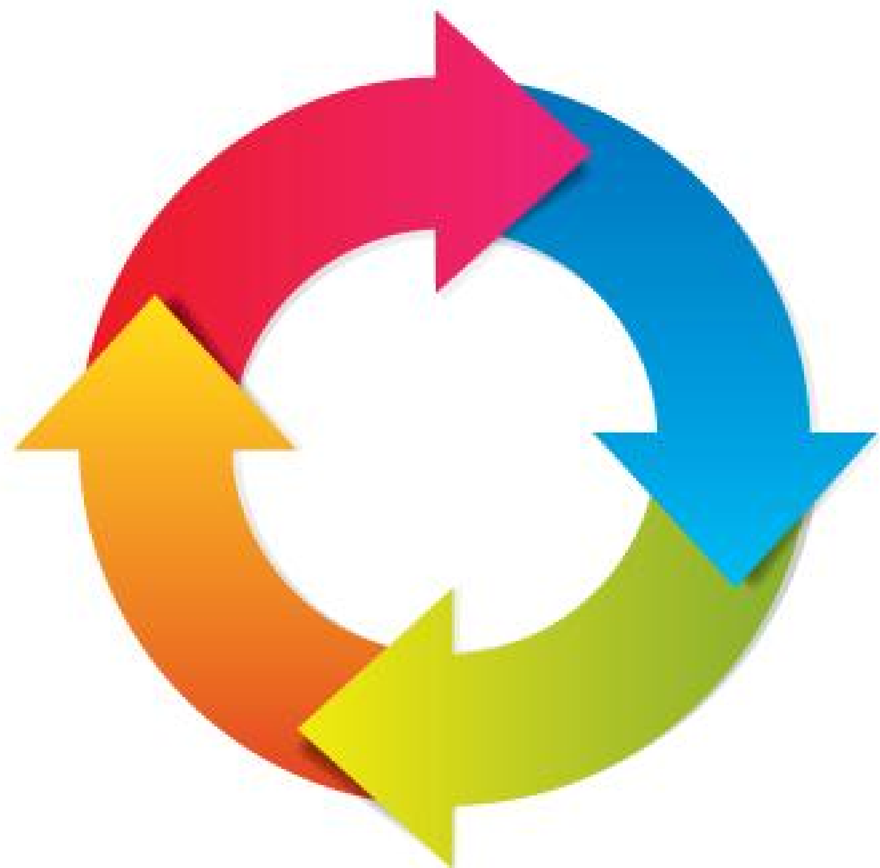
- We specify a set of terms, and we expect to obtain a module—a subset that represents all knowledge about these terms from the original ontology



- Main webpage about modularity: http://owl.cs.manchester.ac.uk/research/modularity/
- Web-based module extractor: http://owl.cs.manchester.ac.uk/modularity/
- Desktop module extractor: https://github.com/rsgoncalves/module-extractor
- Protégé plugin: https://sites.google.com/site/ontologymodularity/

Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- <u>Identify the terms</u>
- Define classes
- Define properties
- Define individuals
- Test



Identifying the terms

- What are the terms we need to talk about?
- What are the properties of these terms?
- What do we want to say about the terms?

Identifying the terms (cont.)

Remember the competency questions:

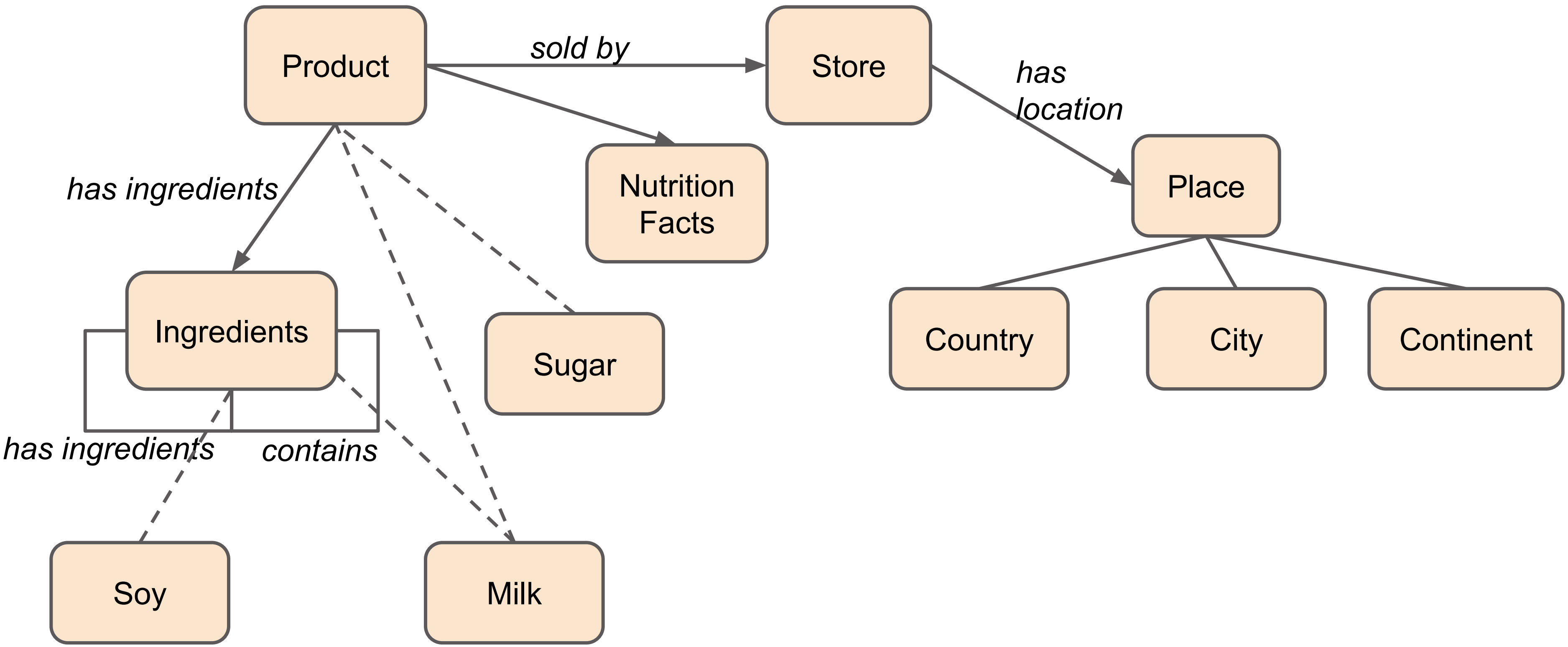
- What are the ingredients of a product?
- What baked goods (such as cakes) do not contain canola oil?
- Does this product contain soy?
- Is this product sold from WholeFoods in Palo Alto?

We use the terms from the competency questions as input to creating the entities in our ontology.

Terms we talk about

- Products, groceries, ingredients, stores, places
- contains, has ingredient, sells
- soy, canola oil, cake, Whole Foods, Palo Alto
- ...

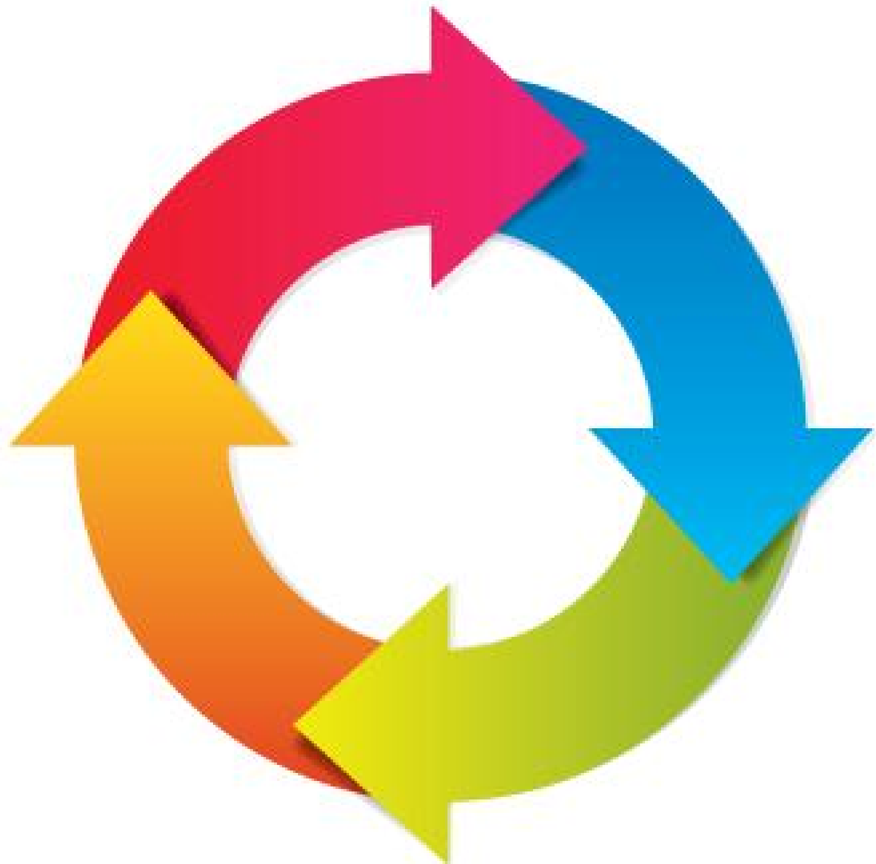
Imagine a mind map for the domain



This is something you do on paper to try to clarify the things your ontology will cover.
This is **NOT** a class diagram, or anything “formal” or “binding”, but just a helper for trying to understand the domain better. The meaning of the lines are not defined.

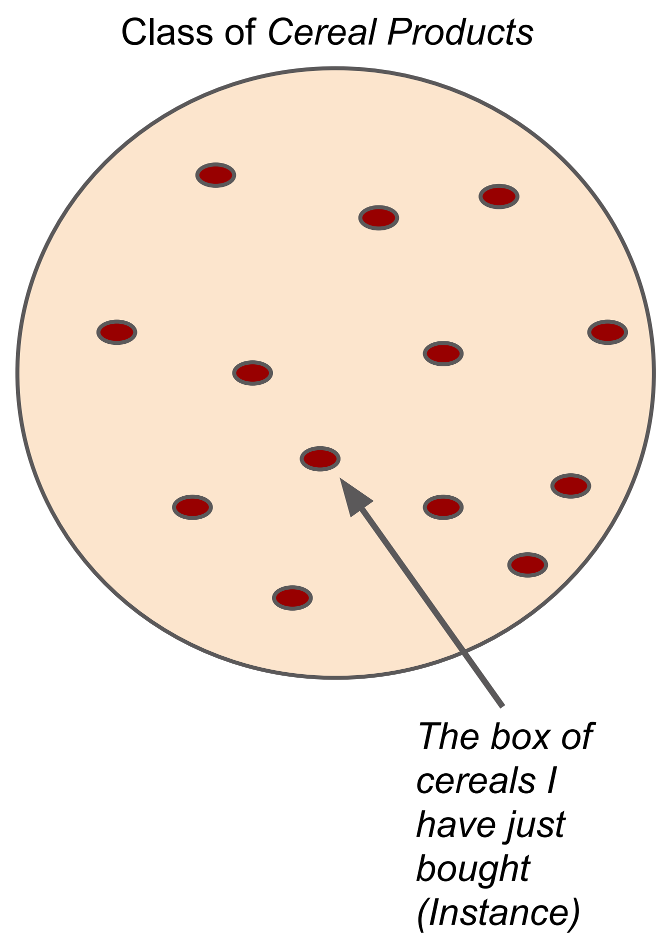
Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- <u>Define classes</u>
- Define properties
- Define individuals
- Test



Defining classes

- A class is a concept in the domain
    - a class of products
    - a class of ingredients
    - a class of dairy products
- A class is a set of elements with similar properties
- Instances of classes
    - a box of cereals you are buying from the store

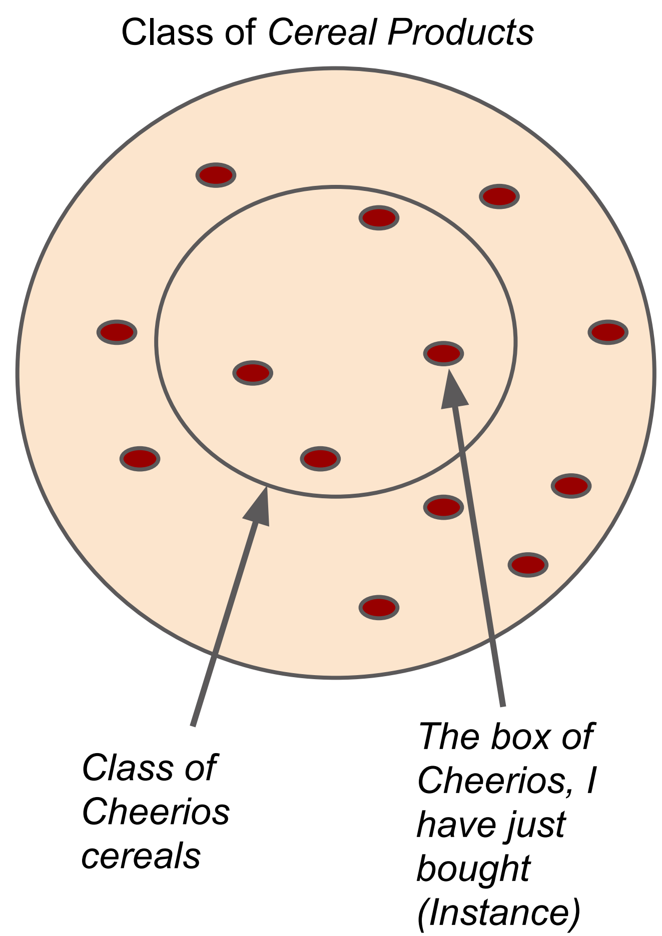


Class inheritance

- Classes usually constitute a taxonomic hierarchy (a subclass-superclass hierarchy)
- A class hierarchy is usually an IS-A hierarchy:

  an instance of a subclass is an instance of a superclass

- If you think of a class as a set of elements, a subclass is a subset

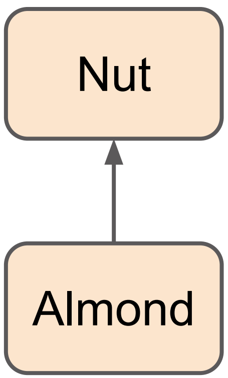


Class inheritance examples

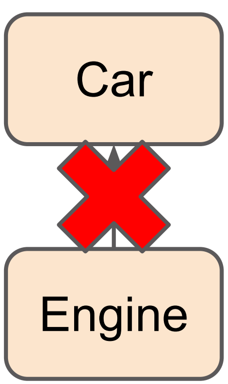
- Apple is a subclass of Fruit
  - *“Every apple is a fruit”*
- Cheerios is a subclass of Cereal Product
  - *“(Every) Cheerios is a cereal product”*
- Cinnamon is a subclass of Spice
  - *“(Every) cinnamon is a spice”*
- Vitamin B12 is a subclass of Vitamin
  - *“Vitamin B12 is a vitamin”*

When in doubt, just say it

(Every) __*subclass*__ is a __*class*__.



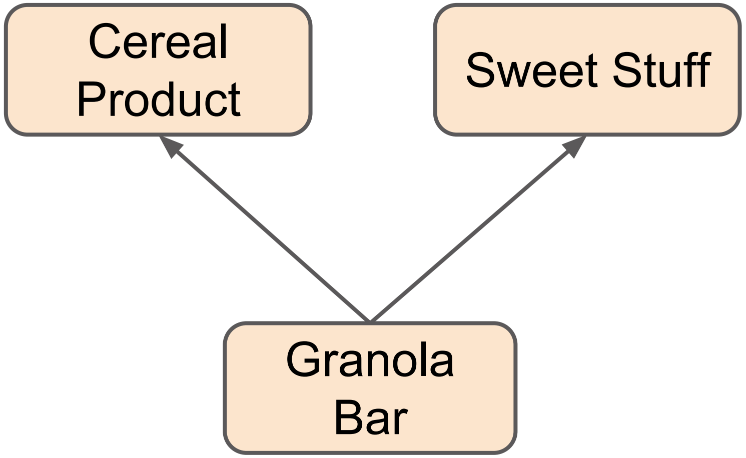
Every almond is a nut.



Every engine is a car.

Multiple inheritance

- A class may have multiple parents, and will inherit the restrictions from both parents
- Not recommended modeling: let the reasoner do the hierarchies for you (covered later today)

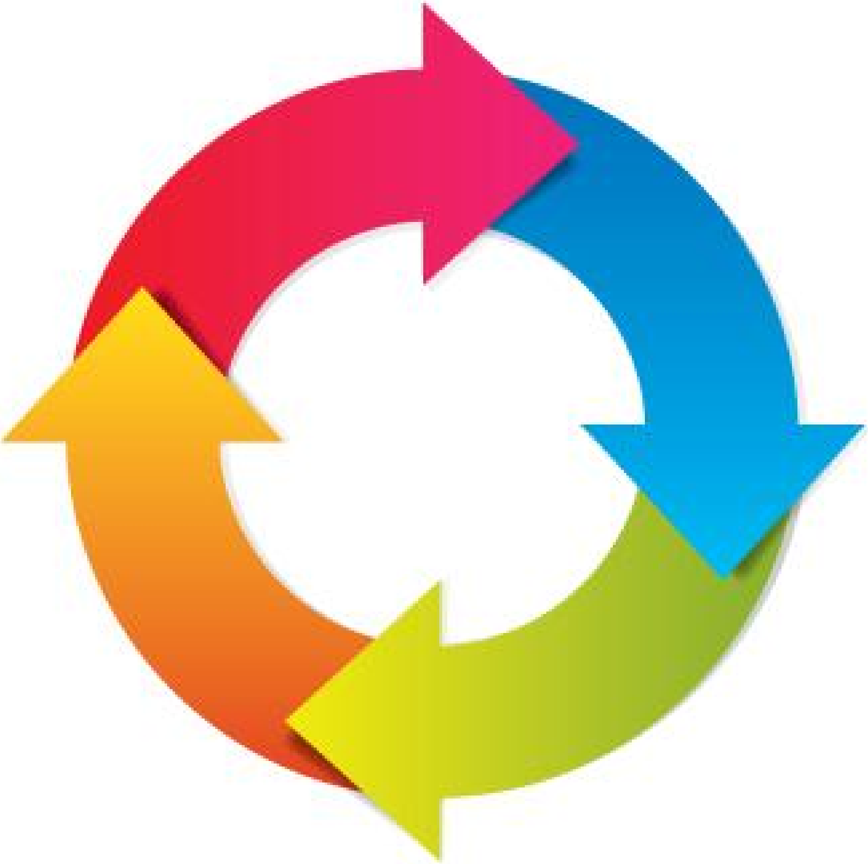


Hands-on exercise: classifying ingredients

- Work in groups of 4
- You will get a stack of cards with ingredient names on them (e.g., baking soda, milk, canola oil, corn starch, cream of tartar, etc.)
- Try to group these cards into categories that make sense to your group
- Once you identify the groups, try to give a name to that group, and write down the criteria used for creating the group
- You may need to use Wikipedia or online search to figure out some of the ingredients
- 15 minutes

Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- Define classes
- <u>Define properties</u>
- Define individuals
- Test



Defining properties

Properties may be derived from the competency questions, the mind map, or statements about the domain.

Products <u>have a price</u>. → price

Products are <u>produced by</u> a manufacturer. → produced by

Products <u>have an expiration date</u>. → expiration date

Products <u>have ingredients</u> food stuff. → has Ingredients

Properties describe instances

Properties *associated* to a class describe the attributes and relationships of the instances of the class.

Class level

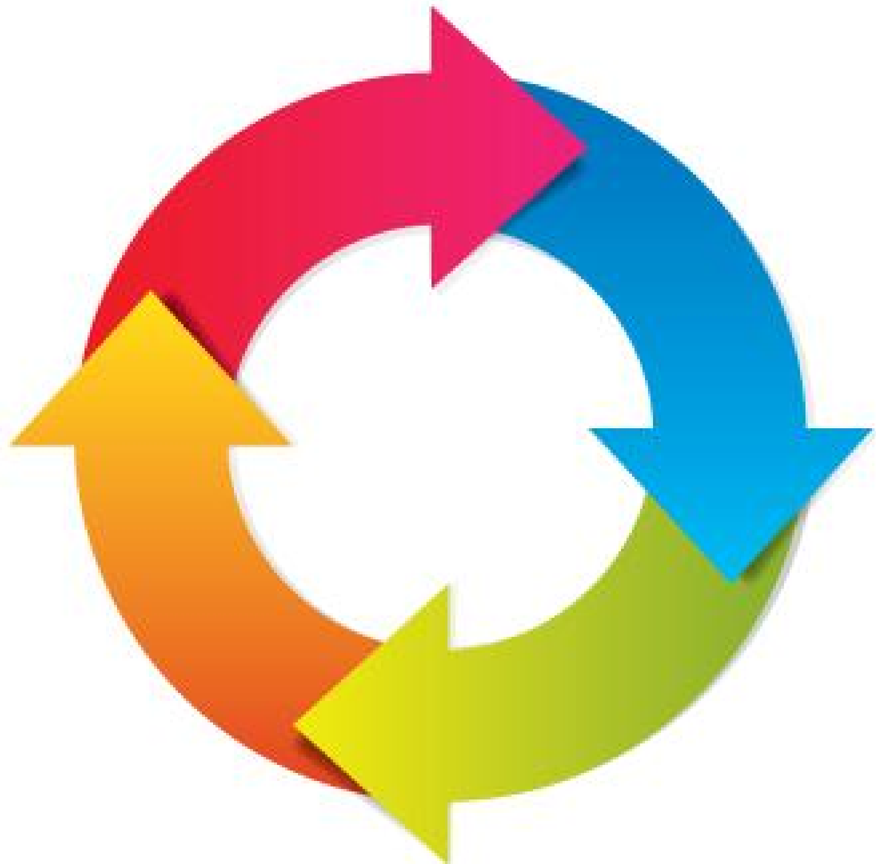
A product is produced by a manufacturer.

This Cheerios box is produced by General Mills.

Instance level

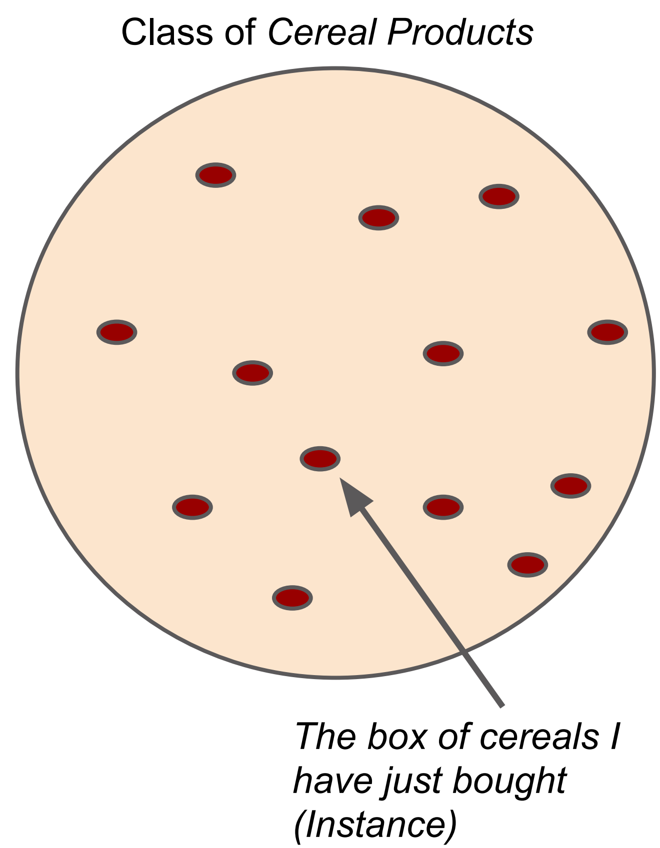
Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- Define classes
- Define properties
- <u>Define individuals</u>
- Test



Individuals

- An individual has to conform to the definitions and restrictions of its class
- *Instance* and *individual* often used interchangeably
- The class is also known as the *type* of the individual
- The class *Person* would have instances, such as *John Doe or Barack Obama*

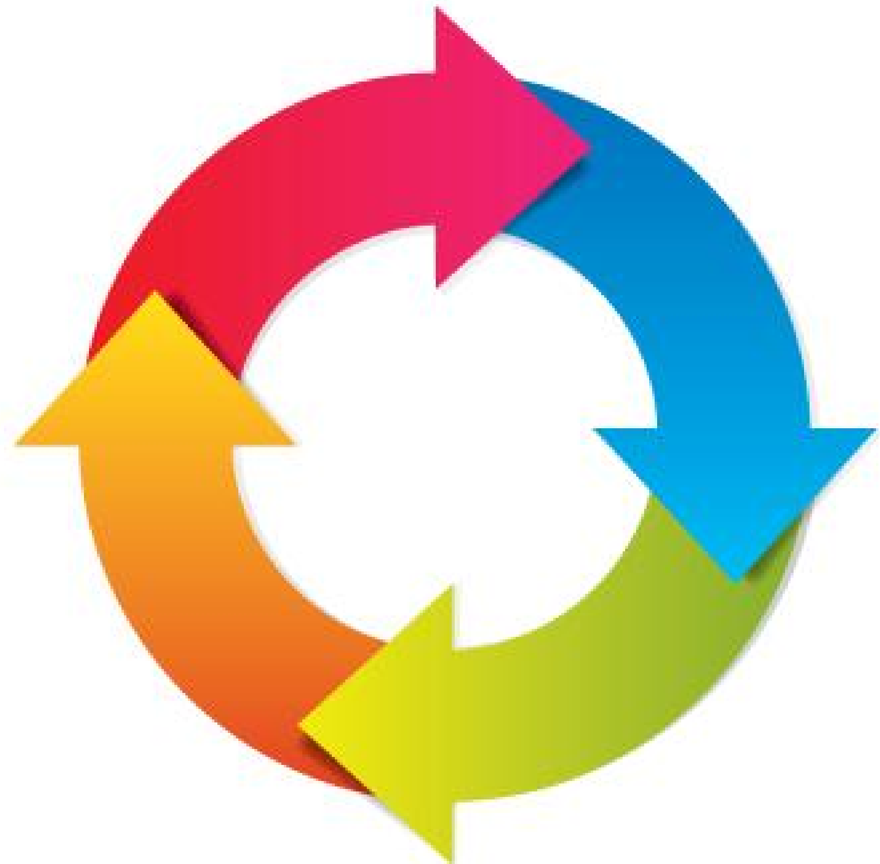


Individuals (cont.)

- Individuals are the last level in the ontology, they cannot be further specialized
- They represent a materialization of your descriptions at the class level
- This is the level where the actual data is put in
- (They are usually the cause of the "inconsistent ontology" error given by the reasoner)

Ontology engineering: an iterative process

- Determine the scope
- Consider reuse
- Identify the terms
- Define classes
- Define properties
- Define individuals
- <u>Test</u>



Testing your ontology

- Check if the competency questions can be answered by your ontology (good check for coverage)
- Run the reasoner to check if the ontology is consistent (may be part of the development workflow)
- Create some “junit” tests that should always return the expected values (e.g., using DL queries, or SPARQL queries)
- Create some probe classes and make sure they always classify as expected (they may need to be always unsatisfiable, or always to be classified under a certain parent)
- If applicable, try to instantiate the ontology, and see if you succeed.

Overview

- Introduction
- Ontologies
  - What is an ontology
  - Step-by-step ontology development
- Further considerations

Defining the scope of an ontology

“How task-dependent are ontologies? Presumably, the kinds of things that actually exist do not depend on our goals. In that sense, ontologies are not task-dependent.

For example, in the domain of fruits, we would focus on particular aspects of reality if we were developing the ontology for the selection of pesticides; we would focus on other aspects of reality if we were developing an ontology to help chefs select fruits for cooking”

Source: Chandrasekaran, 1999 http://www.csee.umbc.edu/courses/771/papers/chandrasekaranetal99.pdf

Approaches to modeling

- There are two approaches:
  - Build a generic ontology
    - conceptual model
    - more reusable
    - a more accurate reflection of the domain
    - may not be always practical
    - may be harder to use in applications
    - likely will not contain individuals
  - Build an application ontology
    - conceptual model highly adapted to a use case
    - less reusable
    - easy to use in your specific application
    - would not necessarily be an accurate reflection of the domain
    - will likely contain more individuals

Multilingual ontologies

- Some ontologies need to be translated into multiple languages
- The common way of dealing with this is to define rdfs:labels in different languages, and let the application choose which label for a which language to display
- For example, for the Product class:
  - rdfs:label “Product”@en
  - rdfs:label “Produto”@es
  - rdfs:label “Produkt”@de
- Make sure that all entities (classes, properties, individuals) contain labels for all the languages
- Translate also other annotation properties (e.g., rdfs:label, etc.)

Standards for metadata and other ontologies

- There are a few metadata standards that are well known and adopted in the community, try to use them and not reinvent the wheel
- **Dublin Core (DC)**: small set of vocabulary terms that can be used to describe web resources (video, images, web pages, etc.), as well as physical resources such as books or CDs, and objects like artworks (http://dublincore.org/)
- **PROV-O**: an ontology for representing provenance (http://www.w3.org/TR/prov-o/)
- **Simple Knowledge Organization System (SKOS):** is a W3C recommendation designed for representation of thesauri, classification schemes, taxonomies, subject-heading systems, or any other type of structured controlled vocabulary. (Commonly used: skos:label, skos:prefLabel, skos:altLabel, skos:definition) http://www.w3.org/2004/02/skos/

Standards for metadata and other ontologies (cont.)

- **Friend-of-a-Friend (FOAF)**: is a machine-readable ontology describing persons, their activities and their relations to other people and objects. http://www.foaf-project.org/

- More here: http://www.w3.org/standards/techs/rdfvocabs#w3c_all

Ontology design patterns

- Some modeling patterns (e.g., part-of) are very common and needed in a lot of ontologies
- A design pattern is a small set of reusable modeling constructs that solve a particular modeling issue independent of the domain
- Two online repositories gathered the most common ontology design patterns:
  - http://ontologydesignpatterns.org
  - http://www.gong.manchester.ac.uk/odp/html/

Modeling Best Practices

- [ ] 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

Review the modeling best practices presentation from the Resources page:

http://tinyurl.com/POSC201703

Further resources

- Ontology 101: A guide to creating your first ontology: http://protegewiki.stanford.edu/wiki/Ontology101
- Semantic Web Best Practices and Deployment Working Group's Semantic Web Tutorials page: http://www.w3.org/2001/sw/BestPractices/Tutorials
- Protégé OWL Tutorial: A step-by-step guide to modeling in OWL using the popular Protégé OWL tools: http://owl.cs.manchester.ac.uk/publications/talks-and-tutorials/protg-owl-tutorial/

Further resources (cont.)

- A Description Logic Primer: http://arxiv.org/abs/1201.4089
- Brief introduction to ontologies, by Robert Stevens: http://www.cs.man.ac.uk/~stevensr/menupages/background.php
- User documentation on the Protégé wiki: http://protegewiki.stanford.edu/wiki/Protege4UserDocs
- ACE Plugin, a natural language interface to OWL: http://attempto.ifi.uzh.ch/aceview/