Ontology Design: some suggestions

Steps in Developing an Ontology

- Requirement analysis
- Consider reuse
- Enumerate terms
- Define classes (concepts)
- Define properties
- Define constraints

Running example: Animal Ontology

Purpose and scope of the Animal ontology

To provide an ontology for an index of a children's book of animals including

- where they live
- what they eat
  (carnivores, herbivores and omnivores)
- how dangerous they are
- how big they are
- a bit of basic anatomy
  (number of legs, wings, toes, etc.)

Enumerate terms

Write down in an unstructured list all the relevant terms that are expected to appear in the ontology. For our animal ontology that could look as follows:

Dog	Carnivore	Dangerous
Cat	Plant	Pet
Cow	Animal	Domestic Animal
Person	Draught Animal	Farm Animal
Tree	Child	Food Animal
Grass	Parent	Fish
Herbivore	Mother	Carp
Male	Father	Goldfish
Female	Pig	

Define Classes

Take a group of things and ask what they have **in common**
and then what other '**siblings**' there might be

For example:

- Plant, Animal — Living Thing (might add Bacteria, Fungi?)
- Cat, Dog, Cow, Person — Mammal (might add Goat, Rabbit?)
- Cow, Goat, Sheep, Horse — Ungulate (hoofed animal)
  (what others are there? do they divide amongst themselves? even/odd-toed?)
- Wild, Domestic — Domestication (what other states?)

Organise the Concepts

Choose some **main axes**:

- add **abstractions** where needed
  (e.g., Living Thing, Mammal, Fish)
- identify **relations**
  (e.g., eats, owns, parent of)
- identify **definable things**
  (e.g., Draught Animal, Father, Herbivore)

  i.e., things where you can say clearly what it means
  try to define a dog precisely — very difficult (a "natural kind")

Self-standing things vs. Modifiers

- **self-standing things** can exist on their own (roughly nouns)
  (e.g., people, animals, houses, actions, processes)
- **modifiers** 'modify' other things (roughly adjectives and adverbs)
  (e.g., wild/domestic, male/female, healthy/sick, dangerous/safe)

Arrange Concepts/Properties into Hierarchy

Reorganise everything but "definable" things into pure **trees** —
these will be the "primitives"

self-standing	modifiers	relations	definable
– LivingThing	Domestication	eats	Carnivore
– Animal	– Domestic	owns	Herbivore
– Mammal	– Wild	parentOf	Child
– Cat	Use	...	Parent
– Dog	– Pet		Mother
– Cow	– Food		Father
– Person	– Draught		FoodAnimal
– Pig	Dangerousness		DraughtAnimal
– Fish	– Dangerous		
– Carp	– Safe		
– Goldfish	Sex		
– Plant	– Male		
– Tree	– Female		
– Grass	Age		
	– Adult		
	– Child		

Defining Classes and a 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)

- If a class has more than a **dozen** direct subclasses,
  additional **subcategories** may be necessary
  (compare to bullets in a list)

  However, if no natural classification exists, the long list may be more natural

- Class names should be either **all singular** or **all plural**
  (*Animal* is **not a kind-of** *Animals*)

Properties

Identify the **domain** and **range** constraints for properties

- Animal *eats* LivingThing

  domain: Animal
  range: LivingThing

- Person *owns* LivingThing except Person

  domain: Person
  range: LivingThing
  and not Person

- Animal *parentOf* Animal

  domain: Animal
  range: Animal

Identify **property restrictions**: what can we say about **all instances** of a class?

- all Cows *eat* some Plants
- all Cats *eat* some Animals
- all Pigs *eat* some Animals and *eat* some Plants
- ...

descriptions of self-standing things

Definable things

Paraphrase and **formalise** the **definitions** in terms of the primitives, relations and other definables

- "A Parent is an Animal that is a *parent of* some other Animal"

$$\text{Parent} \equiv \text{Animal} \sqcap \exists \text{parentOf}.\text{Animal}$$

- "A Herbivore is an Animal that *eats* only Plants"

(NB: all Animals *eat* some LivingThings)

$$\text{Herbivore} \equiv \text{Animal} \sqcap \forall \text{eats}.\text{Plant}$$

- "An Omnivore is an Animal that *eats* both Plants and Animals"

$$\text{Omnivore} \equiv \text{Animal} \sqcap \exists \text{eats}.\text{Plant} \sqcap \exists \text{eats}.\text{Animal}$$

Without a paraphrase we cannot tell if we disagree on what you **meant** to represent and how you **represented** it.

Modifiers

- Identify modifiers that have **mutually exclusive values**
  (Domestication, Dangerousness, Sex, Age)

  NB. Uses are not mutually exclusive
  (can be both Draught and Food)

- Extend and complete lists of values
  (Dangerousness: Dangerous, Risky, Safe)

- Define a **functional property** for every such a modifier

- There are two ways of specifying values for modifiers
  - **value partitions** (classes that partition a quality)
  - **value sets**
    (individuals that enumerate all states of a quality)

Domestication
- Domestic
- Wild

Use
- Pet
- Food
- Draught

Dangerousness
- Dangerous
- Safe

Sex
- Male
- Female

Age
- Adult
- Child

Specifying Values: Value Partitions

Example: a parent quality — Dangerousness

- Define **subqualities** for each degree: Dangerous, Risky, Safe
  - all subqualities are **disjoint**
  - subqualities '**cover**' parent quality, i.e.,

$$\text{Dangerousness} \equiv \text{Dangerous} \sqcup \text{Risky} \sqcup \text{Safe}$$

- Define a **functional property** *hasDangerousness*
  - range is the parent quality, i.e., Dangerousness
  - domain must be specified separately

$$\text{DangerousAnimal} \equiv \text{Animal} \sqcap \exists \text{hasDangerousness}.\text{Dangerous}$$

Specifying Values: Value Sets

Example: a parent quality — SexValue

- Define **individuals** for each value: male, female
  - values are **different** (**NOT assumed in OWL**)
  - value type is '**enumeration**' of values, i.e.,

$$\text{SexValue} \equiv \{\text{female}, \text{male}\}$$

- Define a **functional property** *hasSex*
  - range is the parent quality, i.e., SexValue
  - domain must be specified separately

$$\text{MaleAnimal} \equiv \text{Animal} \sqcap \exists \text{hasSex}.\text{male}$$

"Roles"

To keep primitives **disjoint**:

- need to distinguish the **roles** things play in different **situations** from what they **are**: e.g.,
  - pet, farm animal, draught animal
  - professor, student
  - doctor, nurse, patient
- often need to distinguish **qualifications** from **roles**
  - a person may be qualified as a doctor but playing the role of a patient
- Roles usually **summarise** relations
  - "to play the role of pet" is to say that there is somebody for whom the animal is a pet
  - "to play the role of doctor" is to say that there is somebody for whom the person is acting as the "doctor" — or some "situation" in which they play that role

But we often do **not** want to explain the situation or relation **completely**.

"Roles"

Example: DraughtAnimal, FoodAnimal, PetAnimal

- Identify "roles"
  - draught: cow, horse, dog
  - food: cow, horse
  - pet: horse, dog

- Define subclasses of AnimalUseRole:
  - FoodRole
  - PetRole
  - DraughtRole

$$\text{DraughtAnimal} \equiv \text{Animal} \sqcap \exists \text{hasRole}.\text{DraughtRole}$$