Ontology Development

- Ontology Engineering

- Step-By-Step: Developing an ontology

- Examples of Developing an ontology

Ontology Engineering

It is a methodology issues that is applied in building ontologies, manually, reusing ontologies, and using semi-automatic methods

| determine scope | consider reuse | enumerate terms | define classes | define properties | define constraints | create instances |

Constructing Ontologies main stages:
1. Determine scope
2. Enumerate terms
3. Define concepts/taxonomy
4. Define properties
5. Define facets

How to build an ontology?

- **Steps:**
  - determine domain and scope
  - enumerate important terms
  - define classes and class hierarchies
  - define slots
  - define slot restrictions (cardinality, value-type)
    - » Slot-cardinality
      - *Ex: Borders_with* multiple*, Start_point* single
    - » Slot-value type
      - *Ex: Borders_with-* Country

Step 1: Determine Domain and Scope

Domain: geography

Application: route planning agent

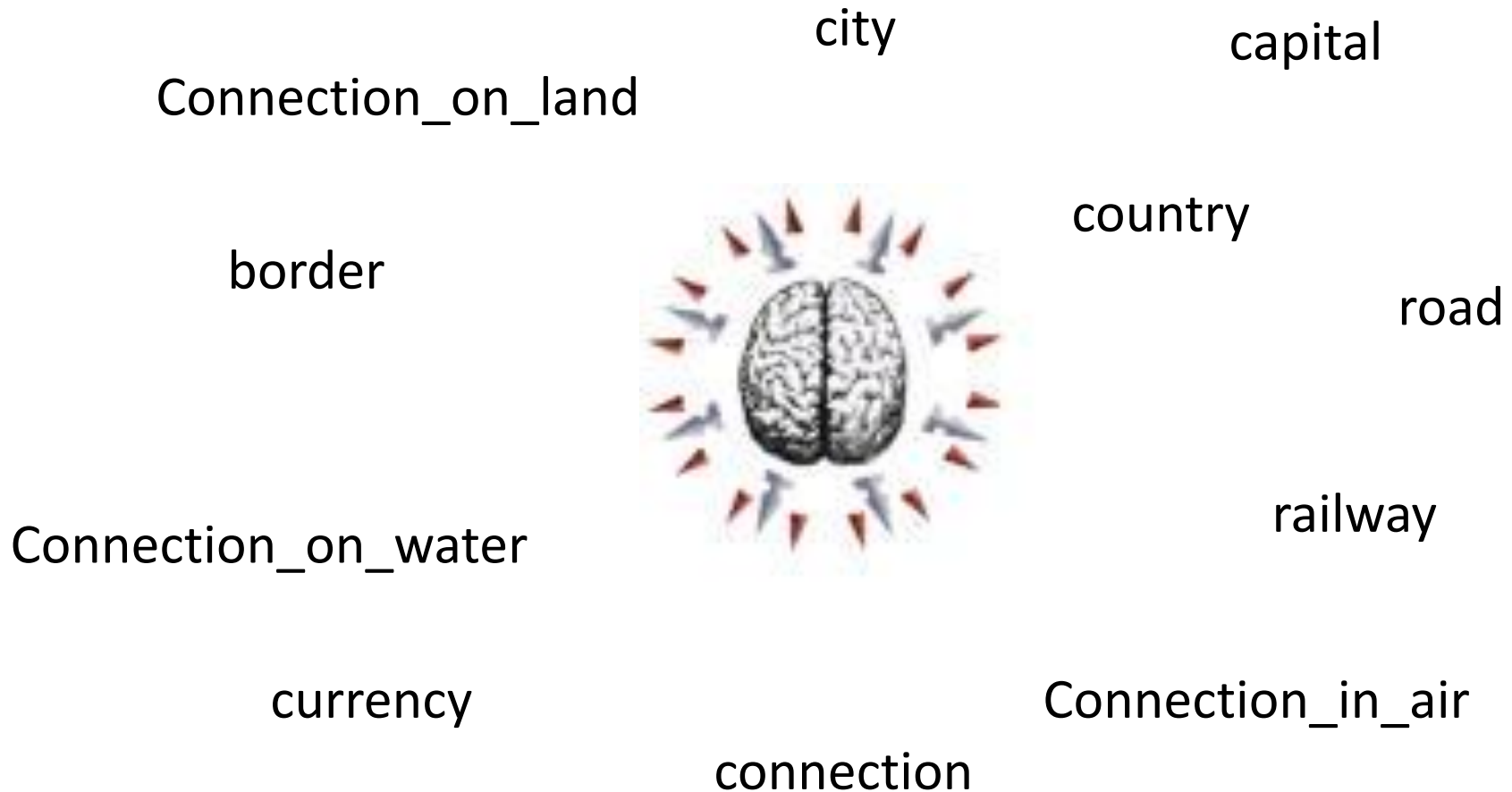**Possible questions:**
Distance between two cities?
What sort of connections exist between two cities?
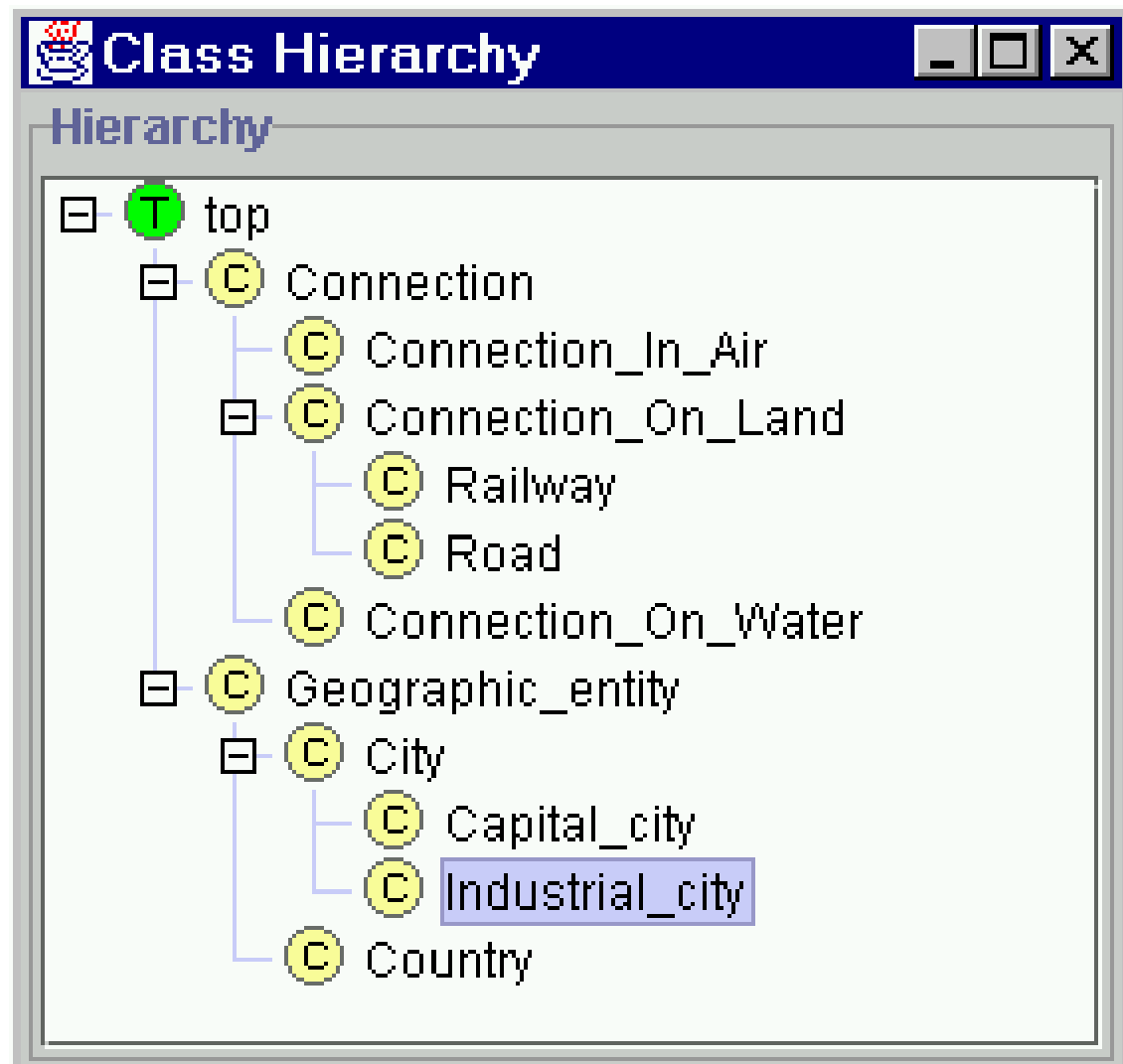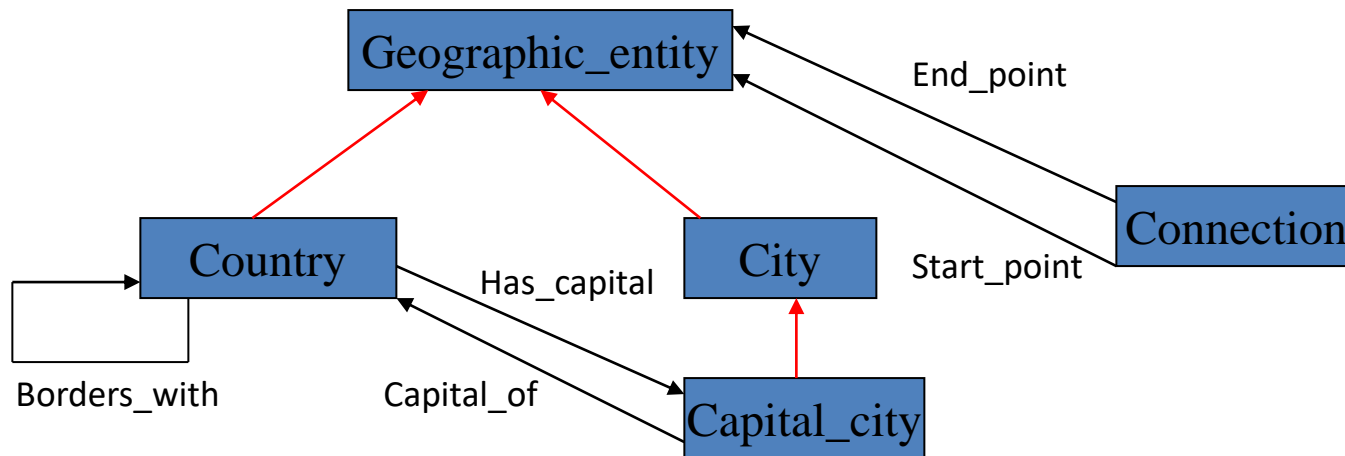In which country is a city?
How many borders are crossed?

Step 2: Enumerate Important Terms

city

capital

Connection_on_land

country

border

road

Connection_on_water

railway

currency

Connection_in_air

connection

Step 3: Define Classes and Class Hierarchy

Class Hierarchy

Hierarchy

- T top
  - C Connection
    - C Connection_In_Air
    - C Connection_On_Land
      - C Railway
      - C Road
    - C Connection_On_Water
  - C Geographic_entity
    - C City
      - C Capital_city
      - C Industrial_city
    - C Country

Step 4: Define Slots of Classes

Step 5: Define slot constraints

- Constraints are Background knowledge on the domain

  - Adult_Elephants weigh at least 2,000 kg

  - All Elephants are either African_Elephants or Indian_Elephants

  - No individual can be both a Herbivore and a Carnivore

Ontology Main Elements

- Defining terms in the domain and relations among them

  - Defining concepts in the domain (classes).

  - Arranging the concepts in a hierarchy (subclass-superclass hierarchy).

  - Defining which attributes and properties (slots) classes can have and constraints on their values.
    - Property restrictions (type, cardinality, domain …)

  - Relations between concepts (disjoint, equality …)

  - Defining instance (individuals)and filling in slot values.

Ontology Components: **Classes**

- Classes are used to group things together.
- In most representations, members of classes must be *individuals*.
- In more expressive representations, *classes* may be also be allowed to be members of other classes.
- Classes can be *subsumed by*, or can *subsume* other classes ⇒ **subclasses** and **superclasses**.
- This leads to the class hierarchy, which is central to most ontologies.
- Some ontologies consist **only** of a class hierarchy – these are called **taxonomy**

Ontology Components: **Individuals**

- Individuals are **instances** or **objects**
- These are usually **concrete** (e.g. uk_prime_minister, FCI_student_1389203)
- They can be **abstract** (e.g. numbers and words)
- Two individuals may be equivalent (e.g. uk_prime_minister, Boris Johnson)
- It is not always clear whether something ought to be an individual or a class (e.g.uk_prime_minister)

Ontology Components: **Attributes**

- Attributes are aspects, properties, features, characteristics, or parameters that objects and classes can have.

- Attributes can link objects and classes to:
  - Specific values (integers, individuals or other literals)
  - Complex data types (e.g. enumerated lists)
  - Boolean values (true/false)
  - Other Classes

Ontology Components: **Relations**

- Relations describe how classes/individual relate to one another.

- Typically, relations are defined between classes, and instantiations of relations are between individuals.
  - course(Course_Name, instructor, Level, Credits, Year)
  - course(CS-SW, Abeer, 4, 3, 2021/2022)

- More restricted representations may limit this, e.g. only allow binary relations.

EXAMPLES OF DEVELOPING AN ONTOLOGY

(C) Copyright 2022 by Prof. Abeer El-Korany

Example1: animals ontology

- Purpose & scope:
- 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
    - numbers of legs, wings, toes, etc.

Example1: Animals & Plants
1-Collect the concepts

- Dog
- Cat
- Cow
- Person
- Tree
- Grass
- Herbivore
- Male
- Female

- Carnivore
- Plant
- Animal
- Fur
- Child
- Parent
- Mother
- Father

- Dangerous
- Pet
- Domestic Animal
- Farm animal
- Food animal
- Fish
- Carp
- Goldfish

Example: Animals & Plants
Ontology Development
1-Organise the concepts

- Dog
- Cat
- Cow
- Person
- Tree
- Grass
- Herbivore
- Male
- Female

- Carnivore
- Plant
- Animal
- Fur
- Child
- Parent
- Mother
- Father

- Healthy
- Pet
- Domestic Animal
- Farm animal
- Draft animal
- Food animal
- Fish
- Carp
- Goldfish

Ontology Development (cont.)
2-Organize the concepts "Laddering"

- **Add abstractions where needed**
  - **e.g. "Living thing"**
- Take a group of things and ask what they have in common
  - Then what other 'siblings' there might be

- e.g.
  - Plant, Animal → Living Thing
    - Might add Bacteria and Fungi but not now (scalability)
  - Cat, Dog, Cow, Person → Mammal
    - Others might be Goat, Sheep, Horse, Rabbit,…
  - Cow, Goat, Sheep, Horse → Hoofed animal
    - What others are there? Do they divide amongst themselves?
  - Wild, Domestic → Domestication

Vocabulary note:
"Sibling" = "brother or sister"

Ontology Development (cont.)
Define:Self_standing_entities(concepts)

- Self-standing things vs. Modifiers
- Things that can exist on there own nouns
  - People, animals, houses, actions, processes, …
    - Roughly nouns
- Modifiers-
- Things that modify ("inhere") in other things
  - (e.g., wild/domestic, male/female, healthy/sick, dangerous/safe)
    - Roughly adjectives and adverbs

- Living Thing
  - Animal
    - Mammal
      - Cat
      - Dog
      - Cow
      - Person
    - Fish
      - Carp
      - Goldfish
  - Plant
    - Tree
    - Grass
    - Fruit

Define Property

- A property is a descriptions of self-standing things
- 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

Ontology Development (cont.)

- **Identify relations**
  - **e.g. "eats", "owns", "parent of"**
- **Identify definable things**
  - **e.g. "child", "parent", "Mother", "Father"**
    - **Things where you can say clearly what it means**
      - (Father, Herbivore, etc)
      - **Try to define a dog precisely – very difficult**
        - » **A "natural kind"**

Definable things

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

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

- A Herbivore is an Animal that eats only Plants"

(NB: all Animals eat some LivingThings)

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

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

Ontology Development (cont.)

Reorganise everything but "definable" things into pure trees

- Living Thing
  - Animal
    - Mammal
      - Cat
      - Dog
      - Cow
      - Person
    - Fish
      - Carp
      - Goldfish
  - Plant
    - Tree
    - Grass
    - Fruit

- Relations
  - eats
  - owns
  - parent-of
  - …
- Definable
  - Carnivore
  - Herbivore
  - Child
  - Parent
  - Mother
  - Father
  - Food Animal

Ontology Development (cont.) Modifiers

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

- Not mutually exclusive usage.

  - (can be both Draught and Food)

- There are two ways of specifying values for modifiers

  - ❖ value partitions (classes that partition a quality .e.g child-adult)

  - ❖ value sets (individuals that enumerate all states of a quality e.g. Dangerousness: Dangerous, Risky, Safe)
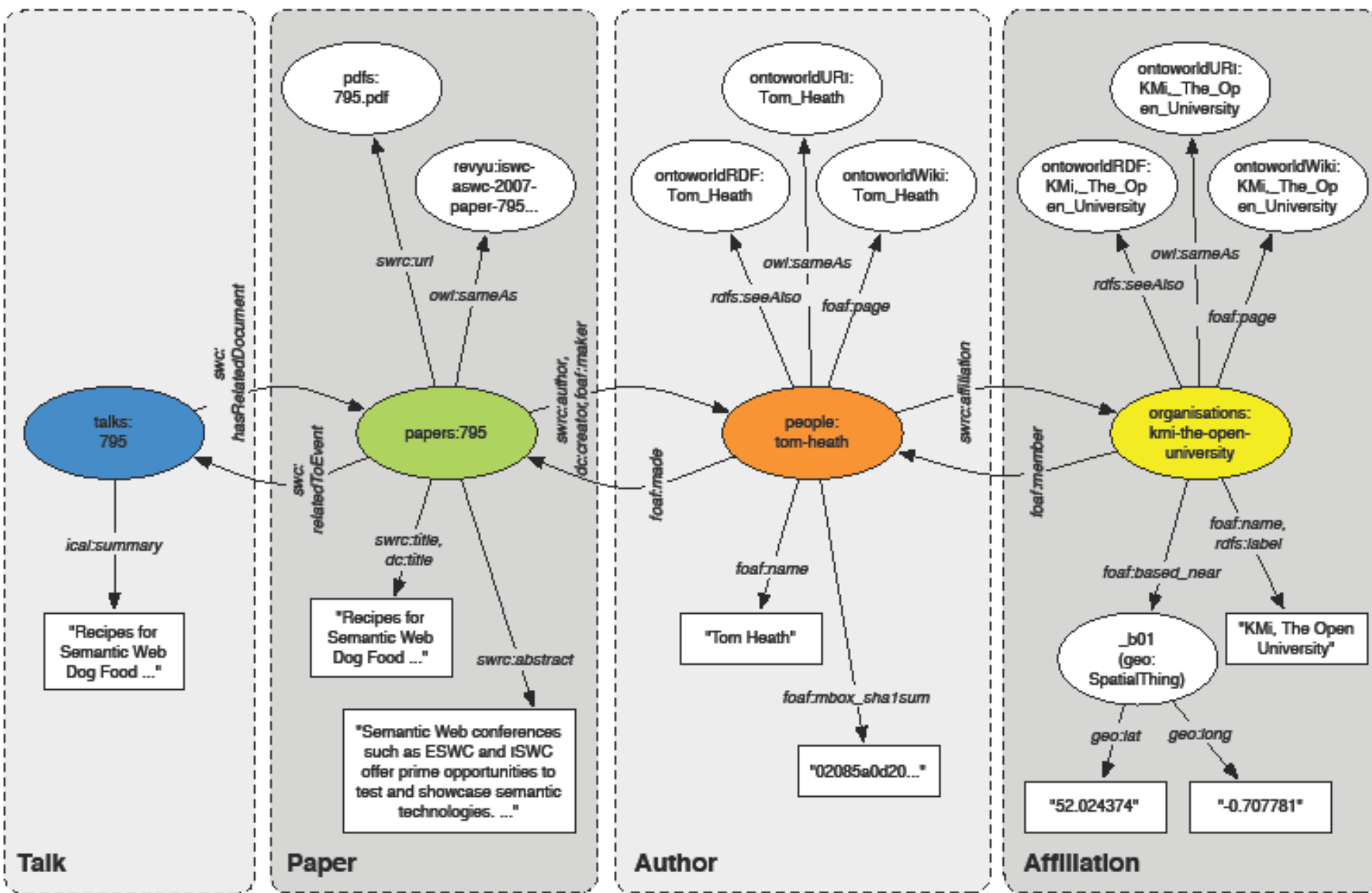
Ontology Development (cont.)

Reorganise everything but "definable" things into pure trees

- Living Thing
  - Animal
    - Mammal
      - Cat
      - Dog
      - Cow
      - Person
    - Fish
      - Carp
      - Goldfish
  - Plant
    - Tree
    - Grass
    - Fruit

- Modifiers
  - domestic
    - pet
    - Farmed
      - Draft
      - Food
    - Wild
  - Health
    - healthy
    - sick
  - Gender
    - Male
    - Female
  - Age
    - Adult
    - Child

- Relations
  - eats
  - owns
  - parent-of
  - …
- Definable
  - Carnivore
  - Herbivore
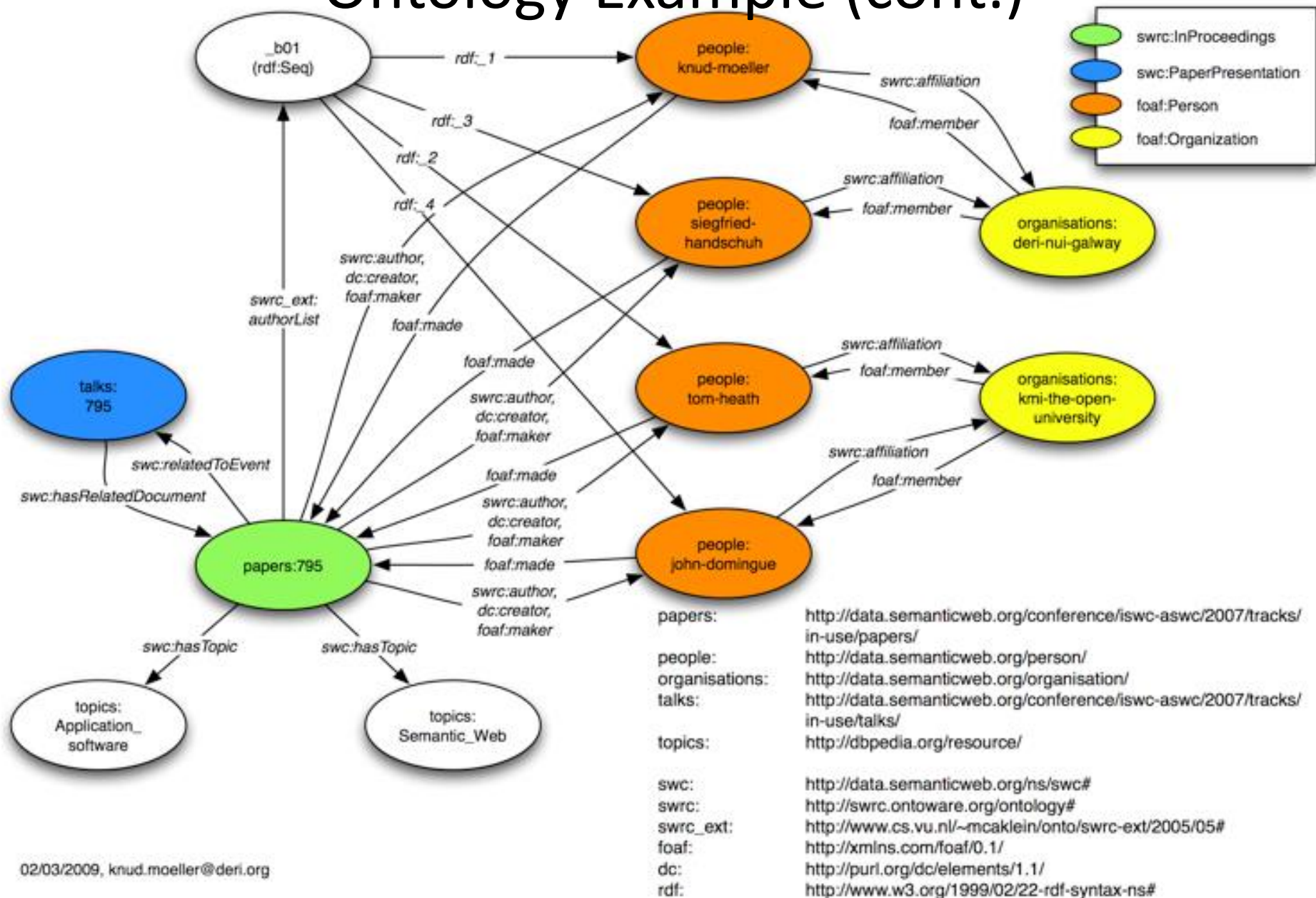 - Child
  - Parent
  - Mother
  - Father
  - Food Animal

Ontology Example

2- How can we create an ontology for the academic research domain (people, publications, etc)?

Ontology Example (cont.)

papers: http://data.semanticweb.org/conference/iswc-aswc/2007/tracks/in-use/papers/
people: http://data.semanticweb.org/person/
organisations: http://data.semanticweb.org/organisation/
talks: http://data.semanticweb.org/conference/iswc-aswc/2007/tracks/in-use/talks/
topics: http://dbpedia.org/resource/

swc: http://data.semanticweb.org/ns/swc#
swrc: http://swrc.ontoware.org/ontology#
swrc_ext: http://www.cs.vu.nl/~mcaklein/onto/swrc-ext/2005/05#
foaf: http://xmlns.com/foaf/0.1/
dc: http://purl.org/dc/elements/1.1/
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

02/03/2009, knud.moeller@deri.org