

COMP318

Ontologies and Semantic Web

Ontology Engineering

- Part 2



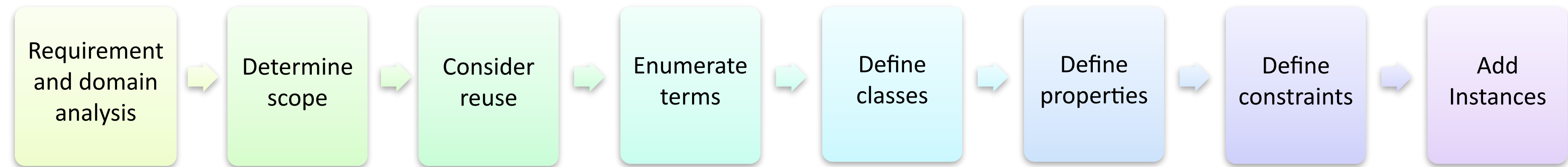
Dr Valentina Tamma

`V.Tamma@liverpool.ac.uk`

Where were we

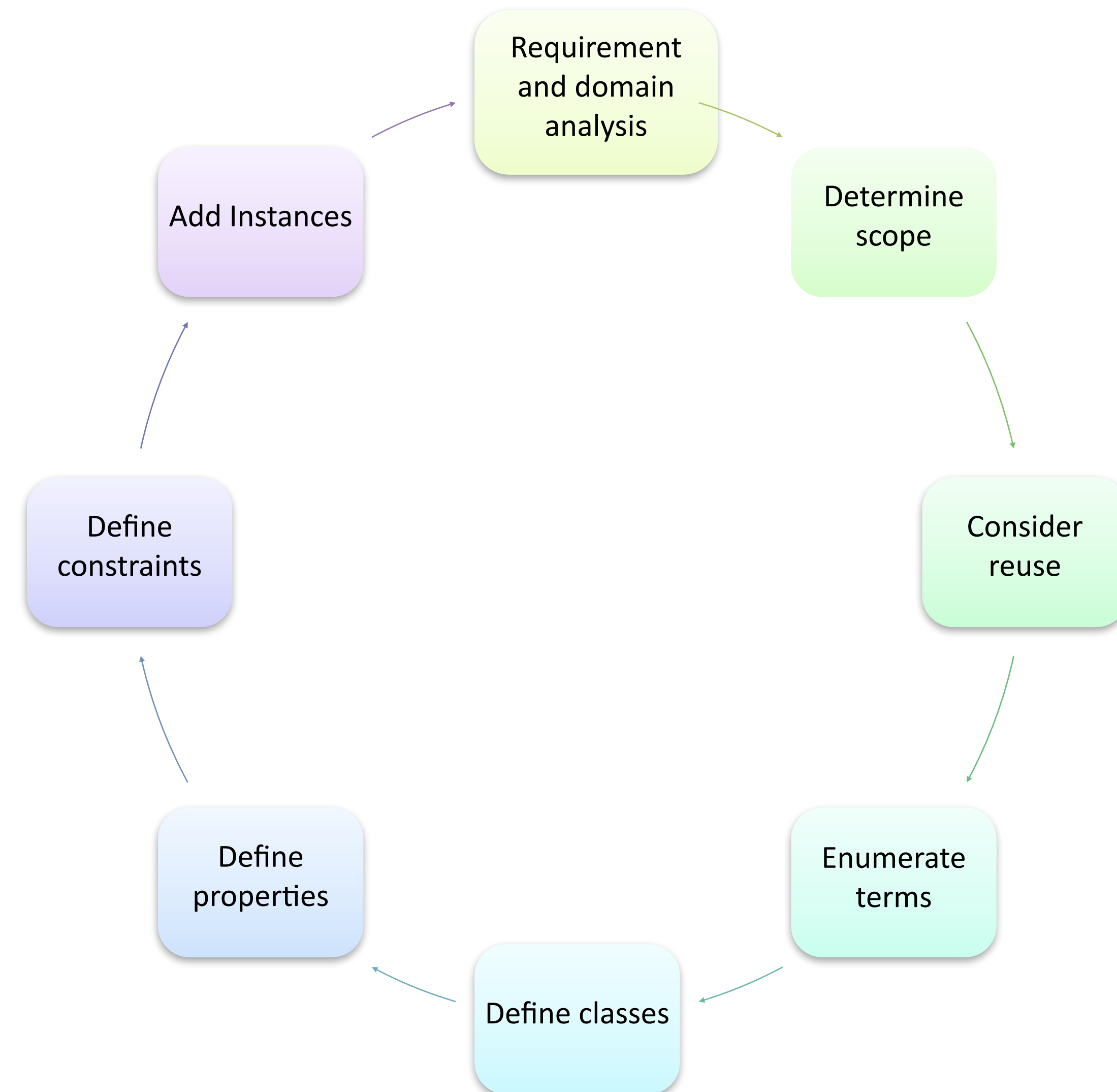
- Types of ontology
- Ontology engineering principles
- Ontology engineering methodologies

Ontology 101

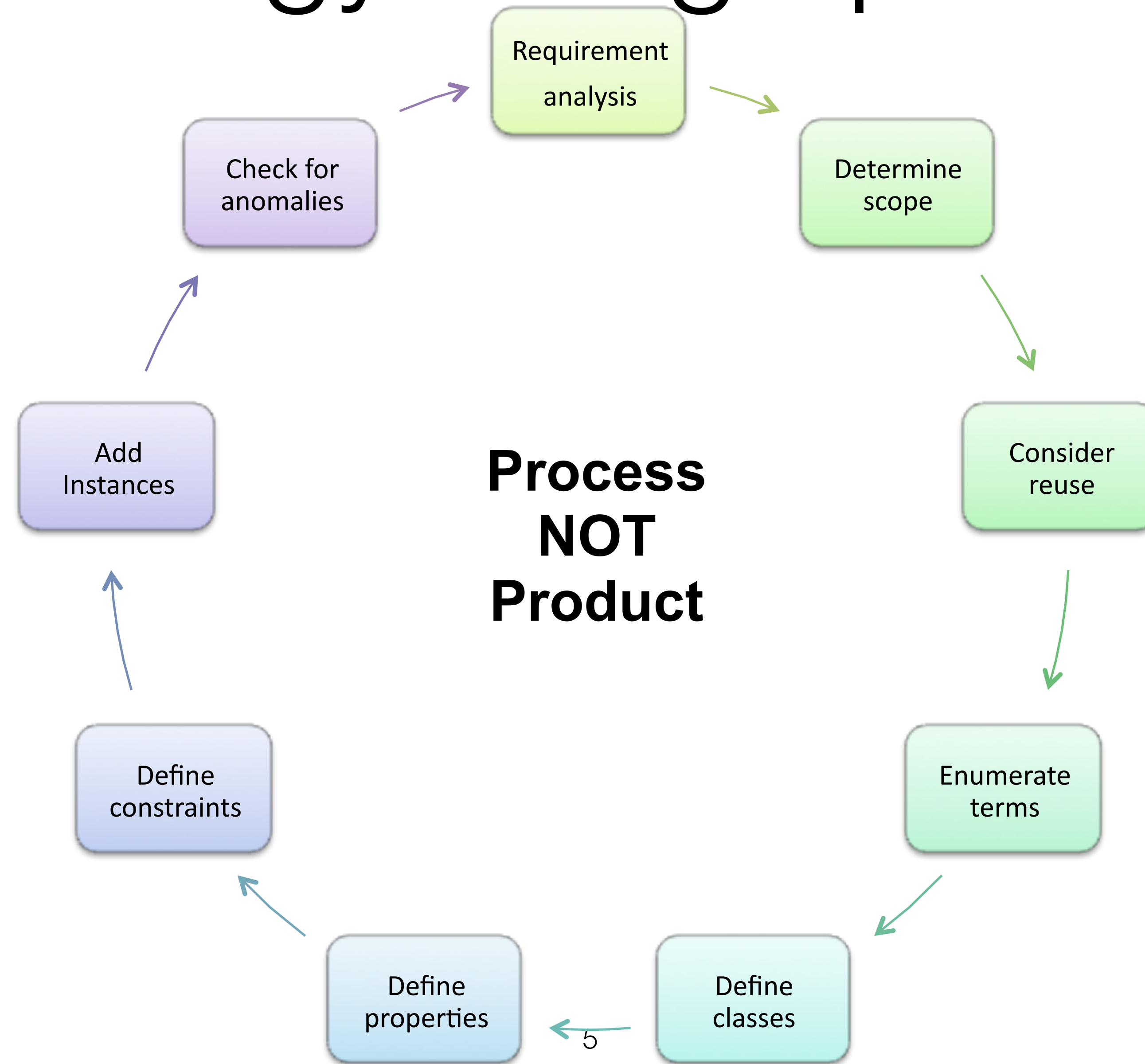


But really more like...

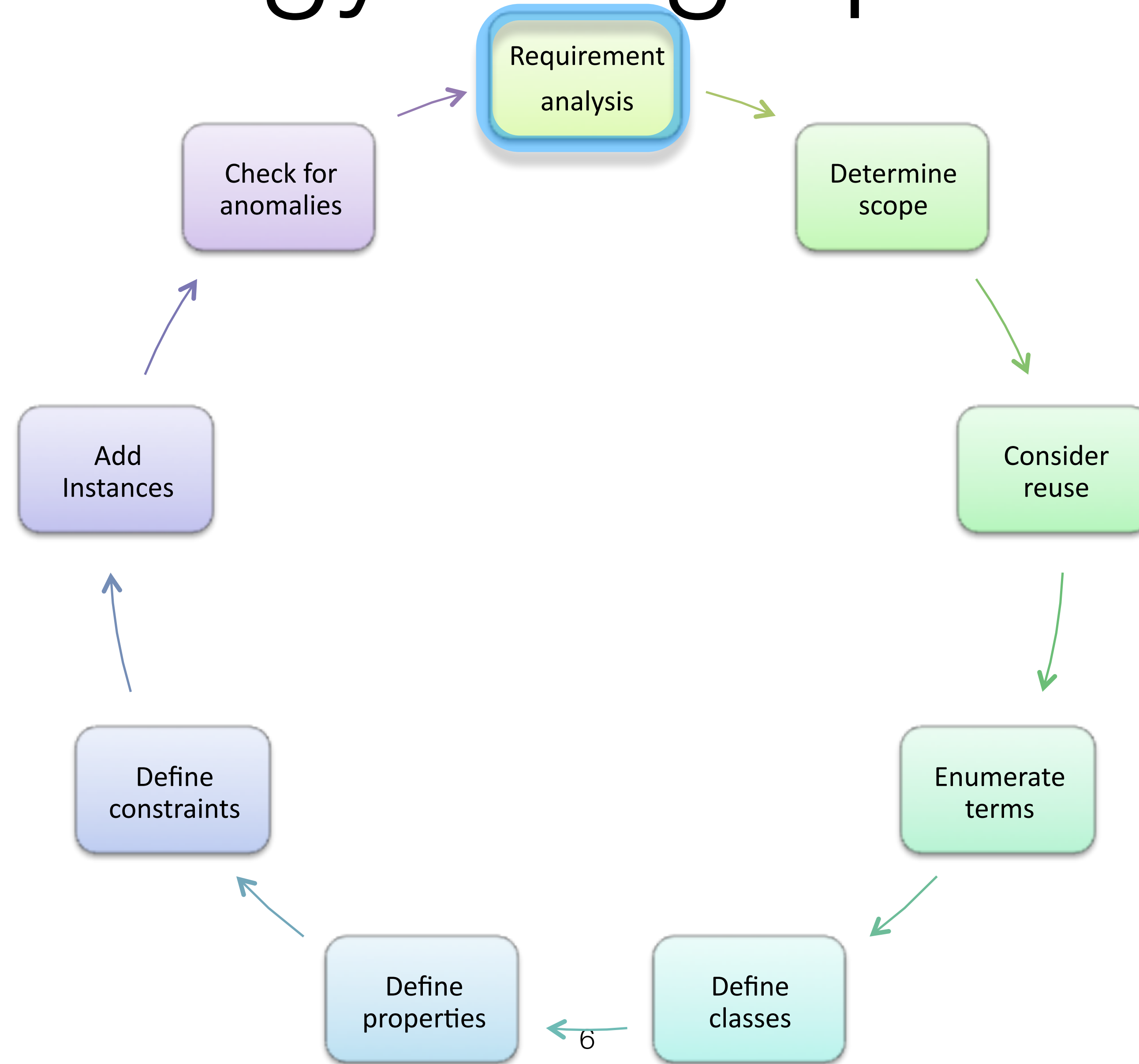
- An iterative Process that repeats continuously and improves the ontology
- there are always different approaches for modelling an ontology
- in practice the designated application decides about the modelling approach



Ontology design process



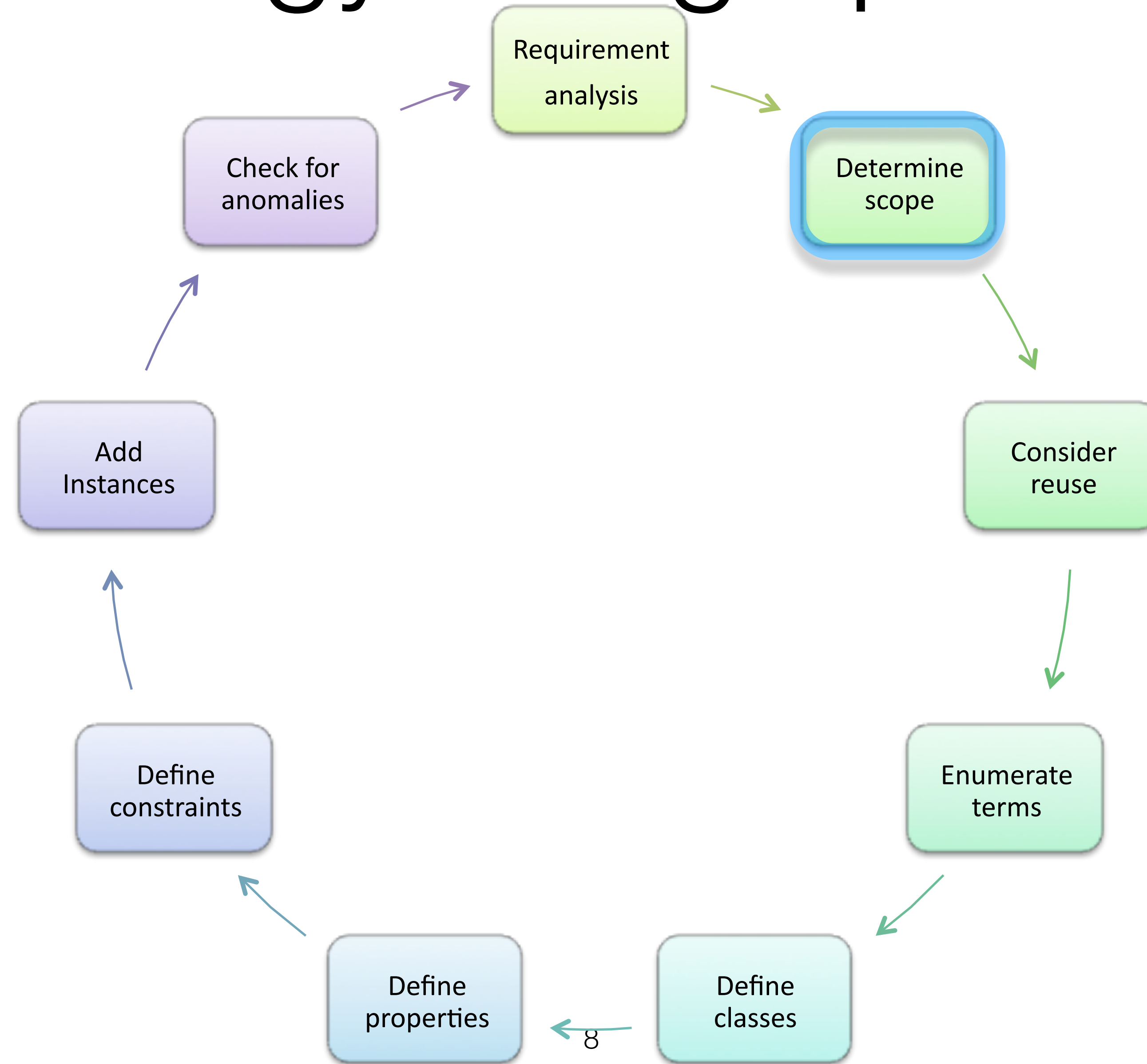
Ontology design process



Requirement analysis

- Requirements, Domain & Use Case Analysis are critical phases as in any software engineering design.
 - they allows ontology engineers to ground the work and prioritise.
- The analysis has to elicit and make explicit:
 - The nature of the knowledge and the questions (competency questions) that the ontology (through a reasoner) needs to answer;
 - *This process is crucial for scoping and designing the ontology, and for driving the architecture;*
 - Architectural issues;
 - The effectiveness of using traditional approaches with knowledge intensive approaches;

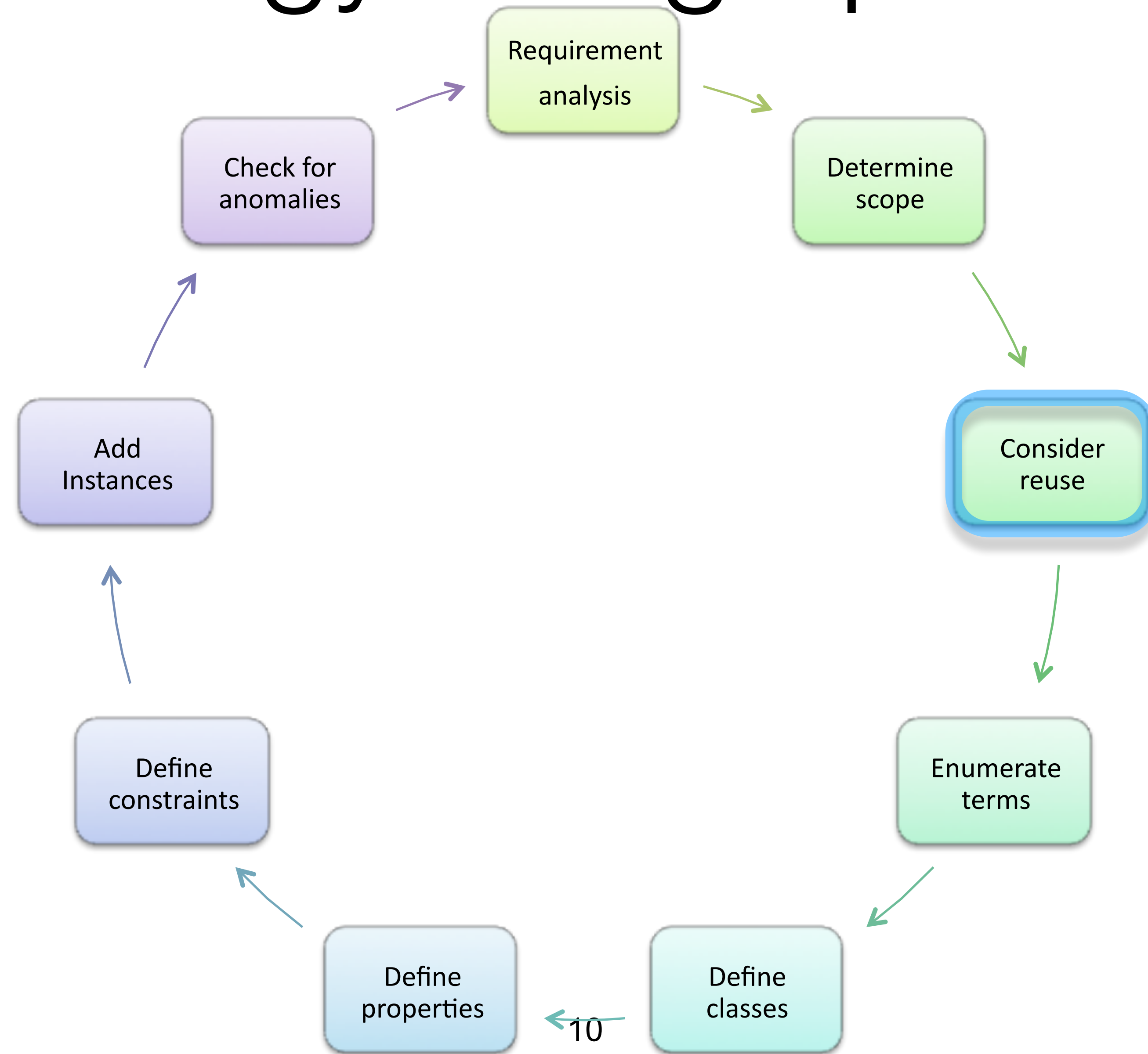
Ontology design process



Determine ontology scope

- There is no correct ontology of a specific domain
 - An ontology is an abstraction of a particular domain, and there are always viable alternatives
- What is included in this abstraction should be determined by
 - the use to which the ontology will be put
 - by future extensions that are already anticipated
- Addresses straight forward questions such as:
 - What is the ontology going to be used for
 - How is the ontology ultimately going to be used by the software implementation?
 - What do we want the ontology to be aware of?
 - What is the scope of the knowledge we want to have in the ontology?

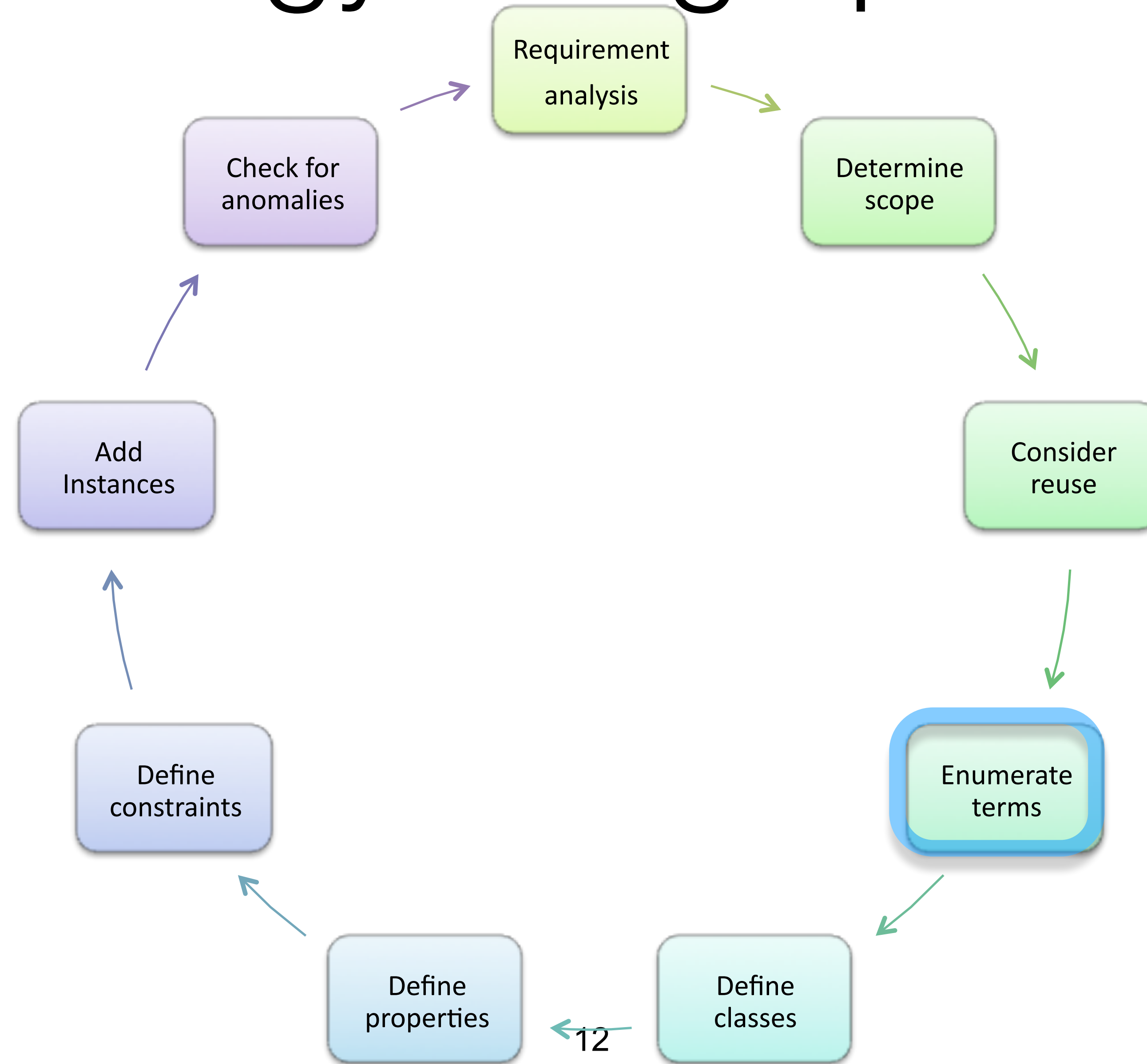
Ontology design process



Consider Reuse

- We rarely have to start from scratch when defining an ontology:
 - There is almost always an ontology available from a third party that provides at least a useful starting point for our own ontology
- Reuse allows to:
 - to save the effort
 - to interact with the tools that use other ontologies
 - to use ontologies that have been validated through use in applications:
 - standard vocabularies are available for most domains, many of which are overlapping
- Identify the set that is most relevant to the problem and application issues
- A component-based approach based on modules facilitates dealing with overlapping domains:
 - Reuse an ontology module as one would reuse a software module
 - Standards, complex relationships are defined such that term usage and overlap is unambiguous and machine interpretable

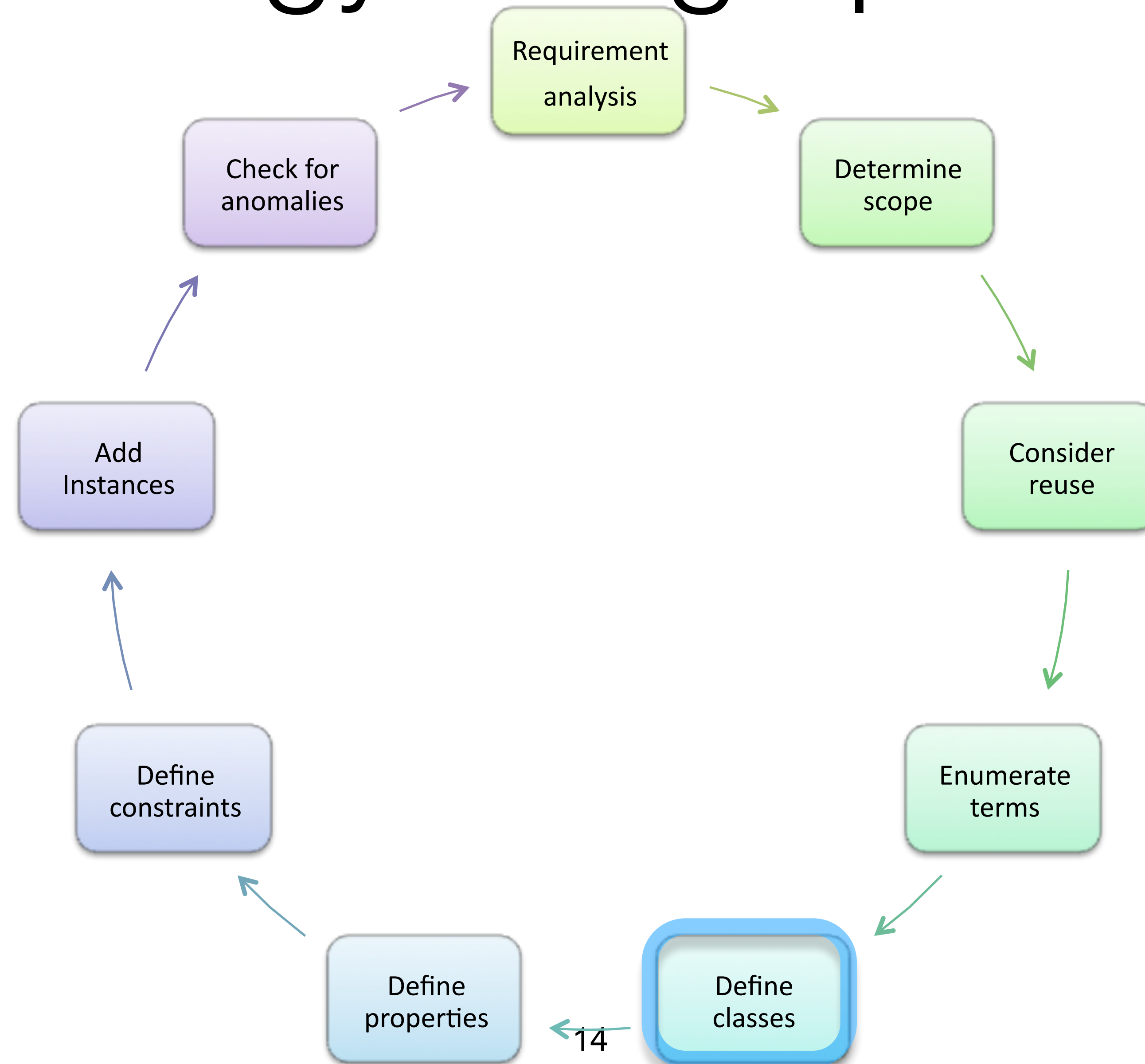
Ontology design process



Enumerate terms

- Write down in an unstructured list all the relevant terms that are expected to appear in the ontology
 - Nouns form the basis for class names
 - Verbs (or verb phrases) form the basis for property names
- Card sorting is often the best way:
 - Write down each concept/idea on a card
 - Organise them into piles
 - Link the piles together
 - Do it again, and again
 - Works best in a small group

Ontology design process



Define classes and their taxonomy

- A class is a concept in the domain:
 - Animal
 - cow, cat, fish
 - A class of properties
 - father, mother
- A class contains necessary conditions for membership
 - type of food, dwelling
- A class is a collection of elements with similar properties
- Instances of classes
 - A particular farm animal, a particular person
 - Tweety the penguin

How do we establish the taxonomy

- Relevant terms must be organised in a taxonomic hierarchy
 - Choose some main axes:
 - add ***abstractions*** where needed
 - Identify ***relations***
 - Identify ***definable things***
 - e.g. *Father is an animal who has children, Herbivore, is an animal who only eats grass...*
 - Not everything is definable, “natural kinds” cannot be defined as precisely in terms of properties or constraints
 - a cat is....?

How do we establish the taxonomy

- Relevant terms must be organised in a taxonomic hierarchy
 - Distinguish between **self standing** things vs **modifiers**:
 - **Self standing** things exist in their own right
 - typically indicated by nouns, e.g. cat, people, animal, action, process...
 - **Modifiers** modify other entities
 - typically denoted by adjectives and adverbs, e.g. wild vs domestic, male vs female, healthy vs sick, etc
- Make sure self-standing terms, modifiers and relations are represented in pure trees
 - no multiple inheritance!
 - these will become the “primitive” concepts from which all other concepts can be defined
 - no definable things

Levels in the class hierarchy

- Different modes of development

- Top-down

- define the most general concepts first and then specialize them

- Bottom-up

- define the most specific concepts and then organize them in more general classes

- Combination (typical)

- breadth at the top level and depth along a few branches to test design

- But... there is no single correct hierarchy

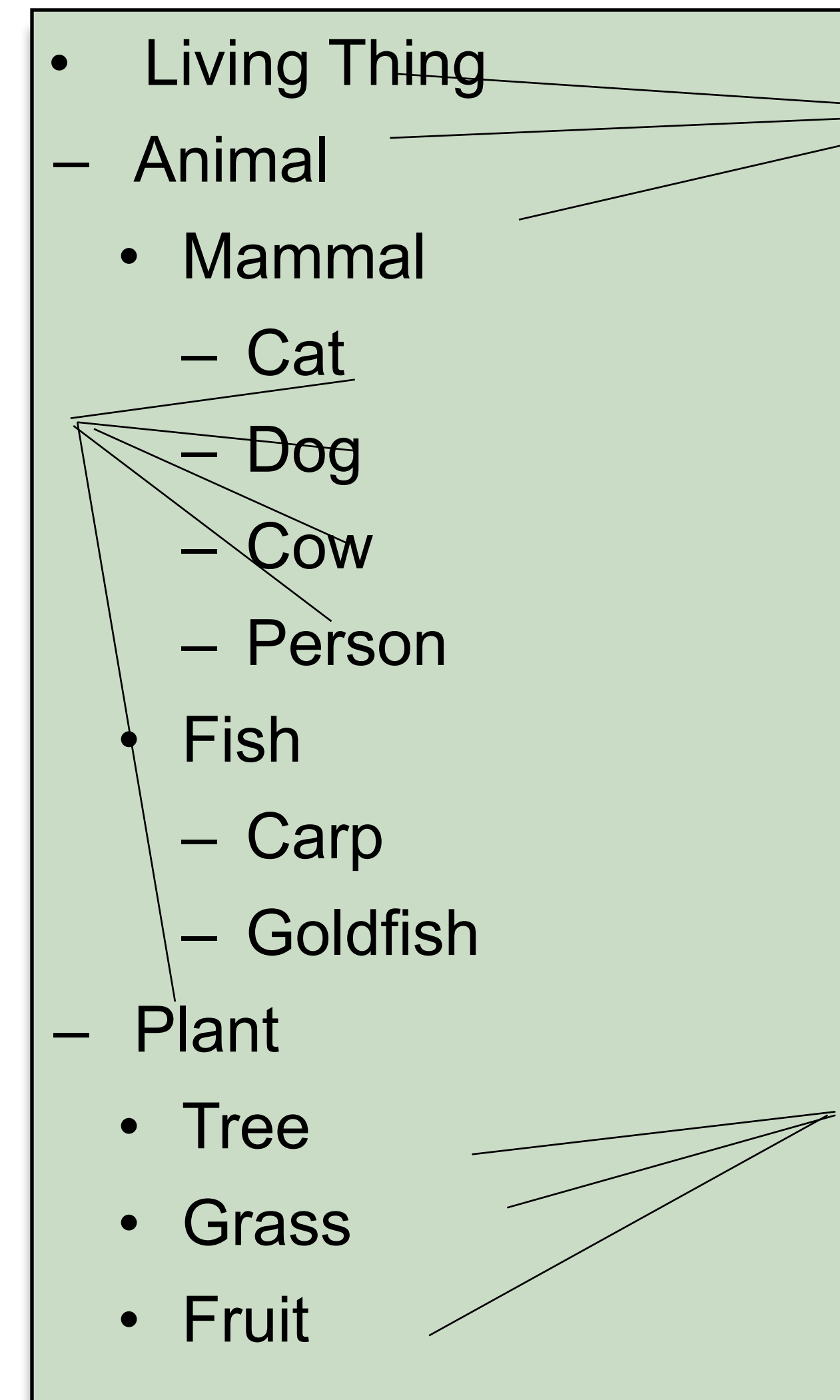
- guidelines helps us to identify the correct ones

- Criteria: ***is each instance of the subclass also an instance of the superclass?***

Middle level

Top level

Bottom level



COMP318

Ontologies and Semantic Web



End of Ontology Engineering

- Part 2

Dr Valentina Tamma
V.Tamma@liverpool.ac.uk

Summary

- Types of ontology
- Ontology engineering principles
- Ontology engineering methodologies

COMP318

Ontologies and Semantic Web



End of Ontology Engineering

- Part 2

Dr Valentina Tamma
V.Tamma@liverpool.ac.uk