Ontology Design Methodology

Content

1. What is ontology?
2. Why we develop ontology?
3. Why we need formal methodology?
4. What is ontology engineering?
5. Ontology Design Methodologies
6. Ontology Design Activity
7. Development Phases of Ontology
8. Ontology Design Patterns

What is ontology?

- An ontology is an explicit , formal specification of shared conceptualization

Why we develop ontology?

- Develop a shared common understanding of the structure and meaning of information
  - among people
  - among software agents
  - between people and software
- Reuse of domain knowledge
  - to avoid “reinventing the wheel”
  - to introduce standards to allow interoperability
- To make domain assumptions explicit
  - easier to change domain assumptions
  - easier to understand and update data

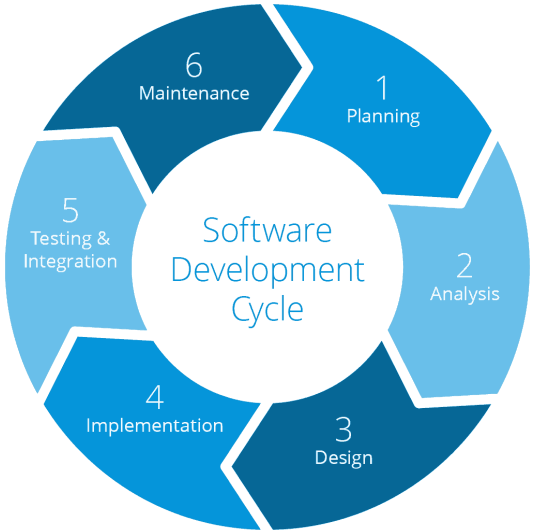
Why we need a formal methodology

- Development of consistent ontologies
- Efficient development of complex ontologies
- Distributed development of ontologies

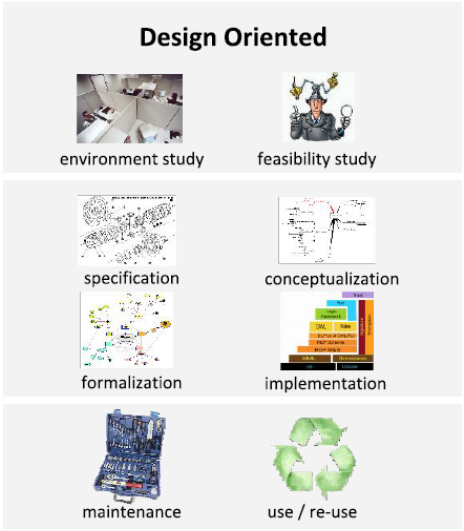
What is Ontology Engineering?

- Meaning of Engineering?
    - concerned with the design, building, and use of software
- Set of tasks related to the development of ontologies for a particular domain

Comparison of Software development and Ontology Development



Software Development

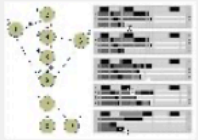


Ontology Development

Ontology Design

Self Reading

Management



scheduling

control

quality assurance

Design Oriented

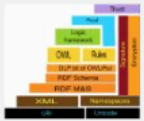
environment study

feasibility study

specification

conceptualization

formalization

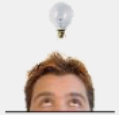


implementation

maintenance

use / re-use

Support



knowledge acquisition

evaluation

integration

documentation

merging

configuration management

alignment

Ontology Development Activity

- Pre Development Activities
  - Environment study
    - We see which applications will use this ontology
  - Feasible study
    - Can ontology be really develop?
- Development
  - Specification
  - Conceptualization
  - Integration
  - Implementation
- Post Development Activities
  - Maintenance
    - Update the new changes in that domain
  - Reuse
    - Already develop vocabulary

Ontology Development Methodology

- Many ontology development methodology
  - Ontology Development 101 (OD101)
  - UPON
  - Methontology
  - eXtreem Design
  - NeOn
  - Co-design

OD101


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

OD101

Its an Iterative approach

UPON

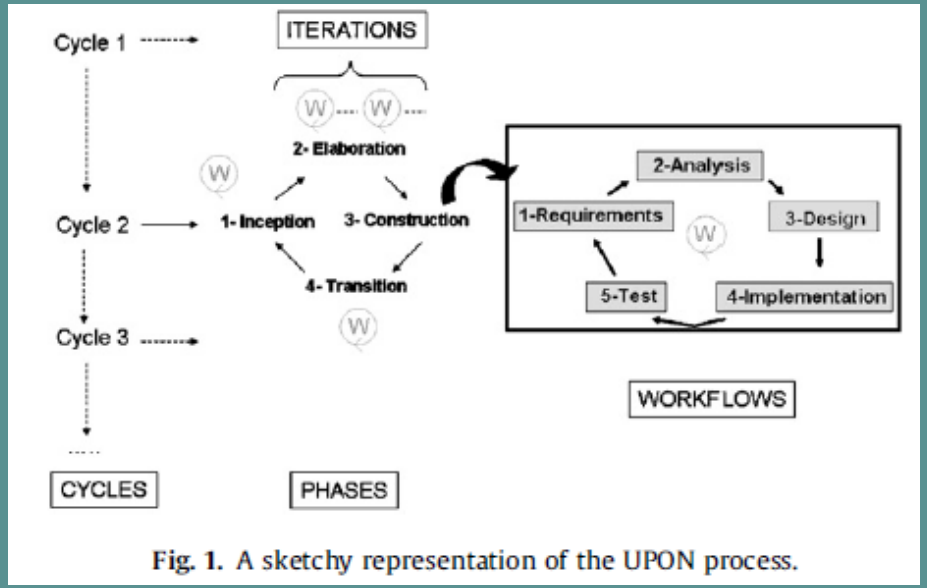


Fig. 1. A sketchy representation of the UPON process.

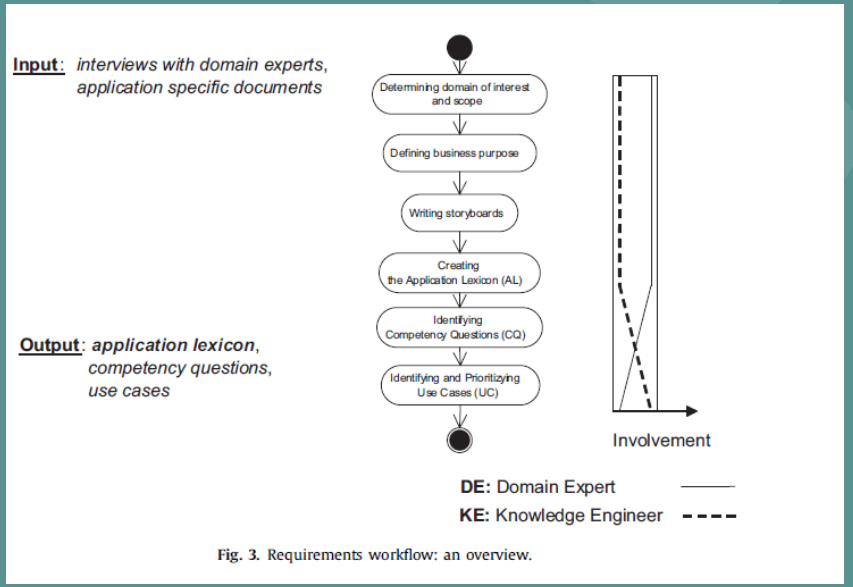
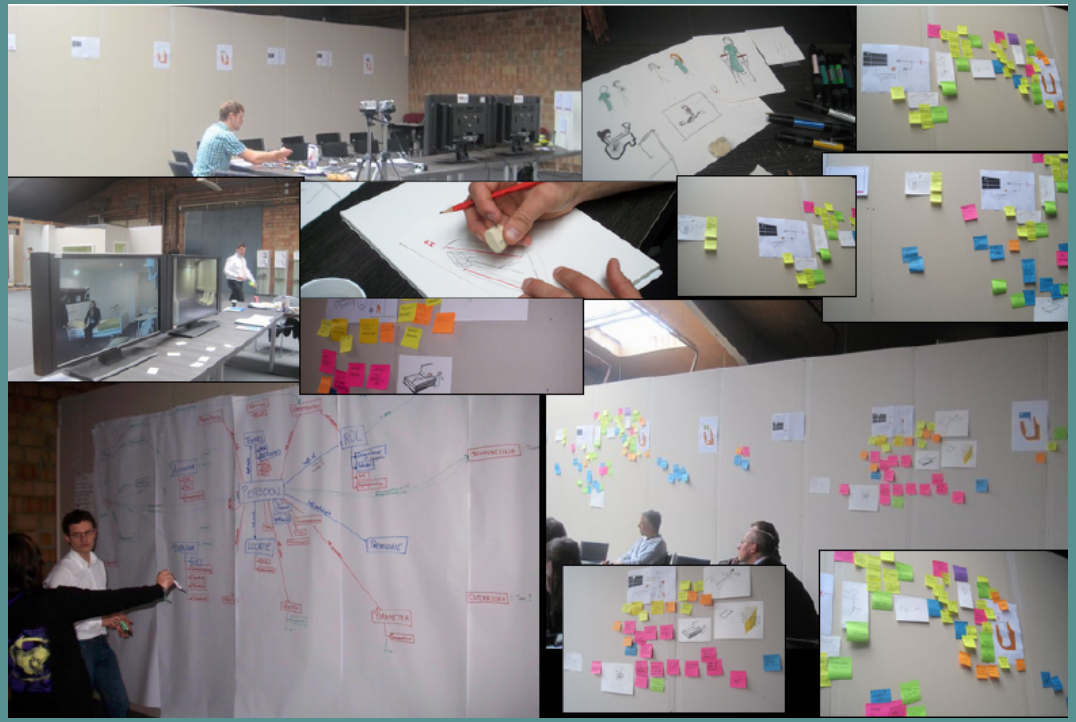


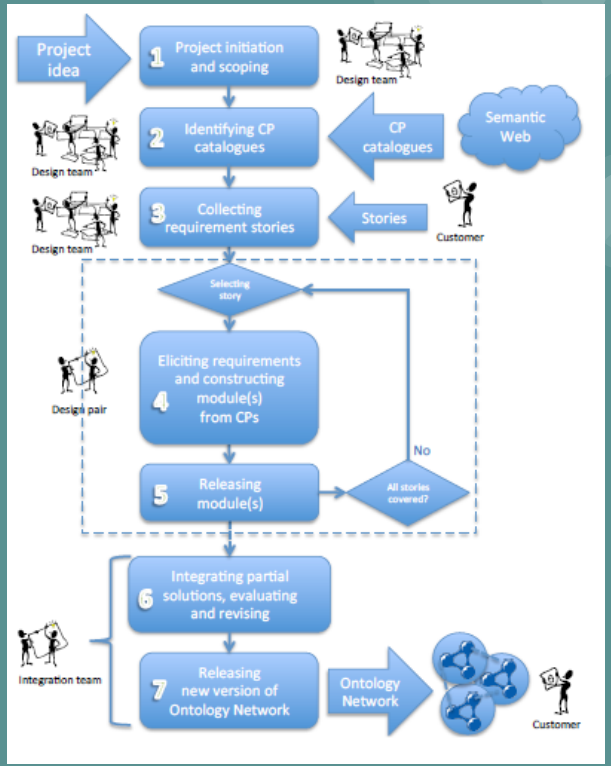
Fig. 3. Requirements workflow: an overview.

Co-design

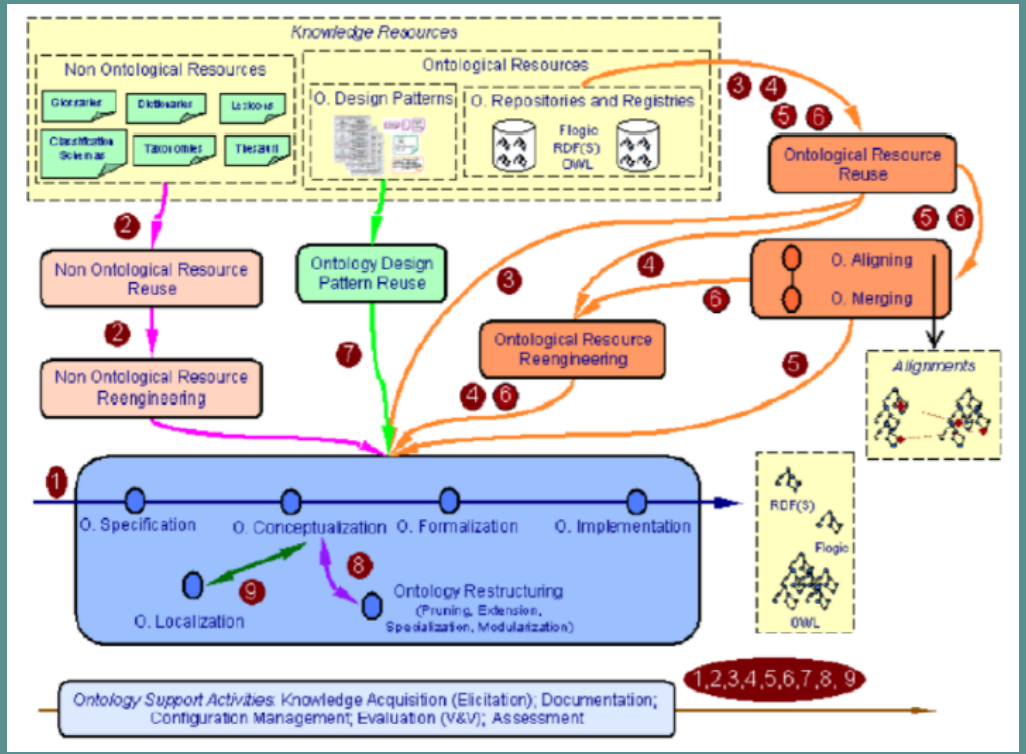


eXtreem Design (XD)

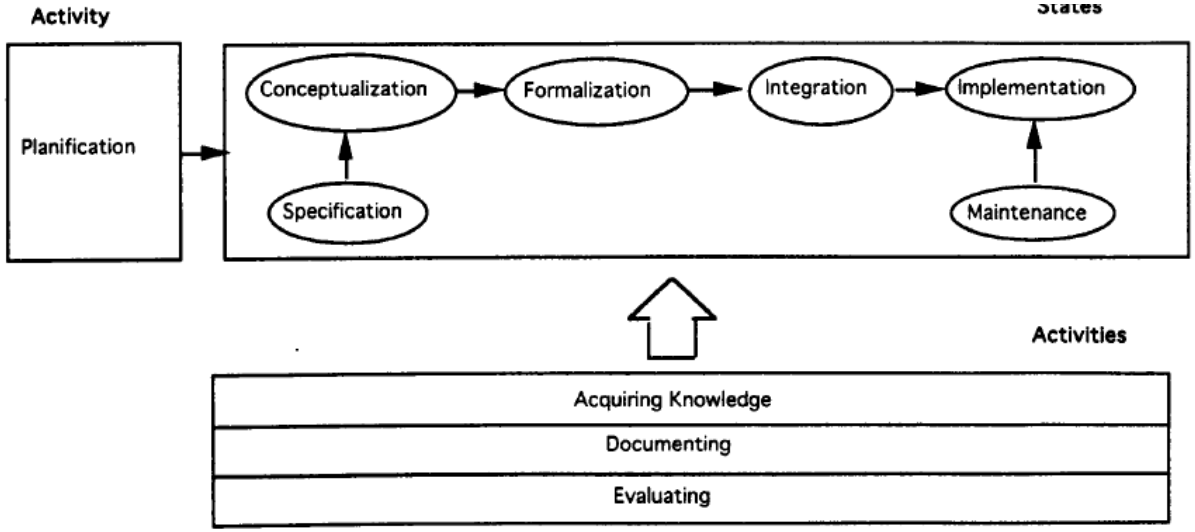
- Inspired by XP but with focus on good design
- An agile methodology for web ontology design
- Customer/domain expert involvement and feedback
- "Customer" stories to derive CQs
- ODP reuse and modular design
- Collaboration and integration
- Task-oriented design, verified by tests
- You quickly have a tangible result



NeOn Methodology



Methontology



Methontology

The most mature methodology is the METHONTOLOGY which was developed taking the IEEE 1074-1995 standard for software development process as a starting point

Phase-1 specification Phase

- Determine the domain and scope of the ontology
    - What is the domain that the ontology will cover?
    - For what we are going to use the ontology?
    - Who will use and maintain the ontology? (end users)
- Write competence Questions
    - Questions that your Knowledge base is able to answer

Methontology

Competence Questions Patterns

•Question Type:

Selection question: Which pizza characteristics should i consider when choosing a pizza?

Binary Question: Does pizza contains halal meat? (Ans will be in yes or no)

Counting Question: How many pizzas have either cheese or veg topping?

•Question Polarity:

It determines if the question is asked in a positive or negative manner

Example: "Which pizzas contain halal meet?" v.s. "Which pizza has no vegetables?"

•Predicate Arity:

Unary predicate is concerned with a single set of entities/values and its instances

Example: "Is it thin or thick bread?".

Binary predicate is concerned with the relation between 2 sets of entities/values and their instances

N-ary predicate is concerned with the relation among multiple (3) sets of entities/values and their instances

•Modifier:

Quantity modifier: restricts the number of relations among entities/values.

For example "If I have 3 ingredients, how many kinds of pizza I would make?"

"Which pizza has the most toppings?" has a quantity modifier most on the number of pizza topping relations for each pizza.

Numeric modifier: "What pizza has very little (10%) onion and/or leeks and/or green peppers?"

Methontology

Phase-2 Knowledge Acquisition

- Structured/ semi structured interviews with experts
- Study the main concepts given in books, articles etc (non ontological resources)

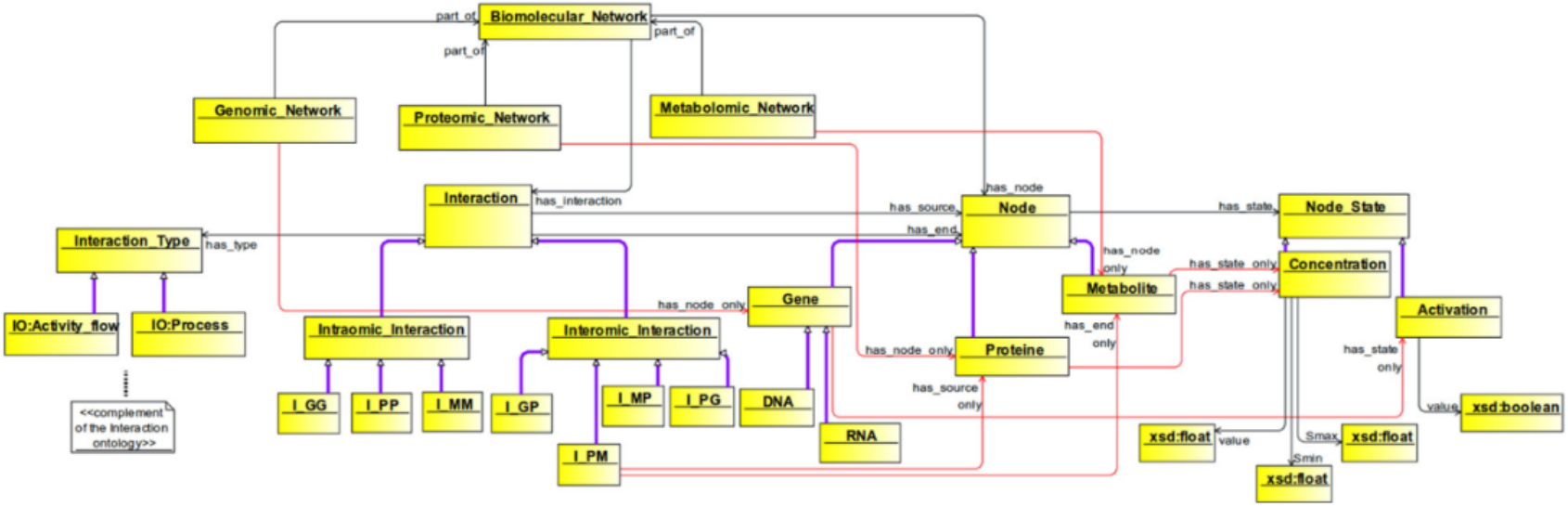
Phase-3 Conceptualization

- Build a preliminary draft of the requirements specification document
- Identify which terms are most common and frequently used (Enumerate the terms)
- Identify the classes, Instances and properties

Phase-4 Formalization

- Transforms the conceptual model into a formal or semi-computable model

Methontology



Conceptual Modeling

Methontology

Phase-5 Integration

- Ontological Resource Reuse
- Reuse of Ontology design pattern

Phase-6 Implementation

- Encoding in the formal language

Phase- 7 Evaluation

- Ontology Verification: are you producing the ontology in the right way?
- Ontology Validation : are you producing the right ontology?

Comparison of OD101 & Methontology

Step 1. Determine the domain and scope of the ontology
Step 2. Consider reusing existing ontologies
Step 3. Enumerate important terms in the ontology
Step 4. Define the classes and the class hierarchy
Step 5. Define the properties of classes–slots
Step 6. Define the facets of the slots
Step 7. Create instances

Non ontological resource reuse not mentioned
No collaboration

Step 1. Specification

Step 2. Knowledge Acquisition

Step 3. Conceptualization

Step 4. Formalization

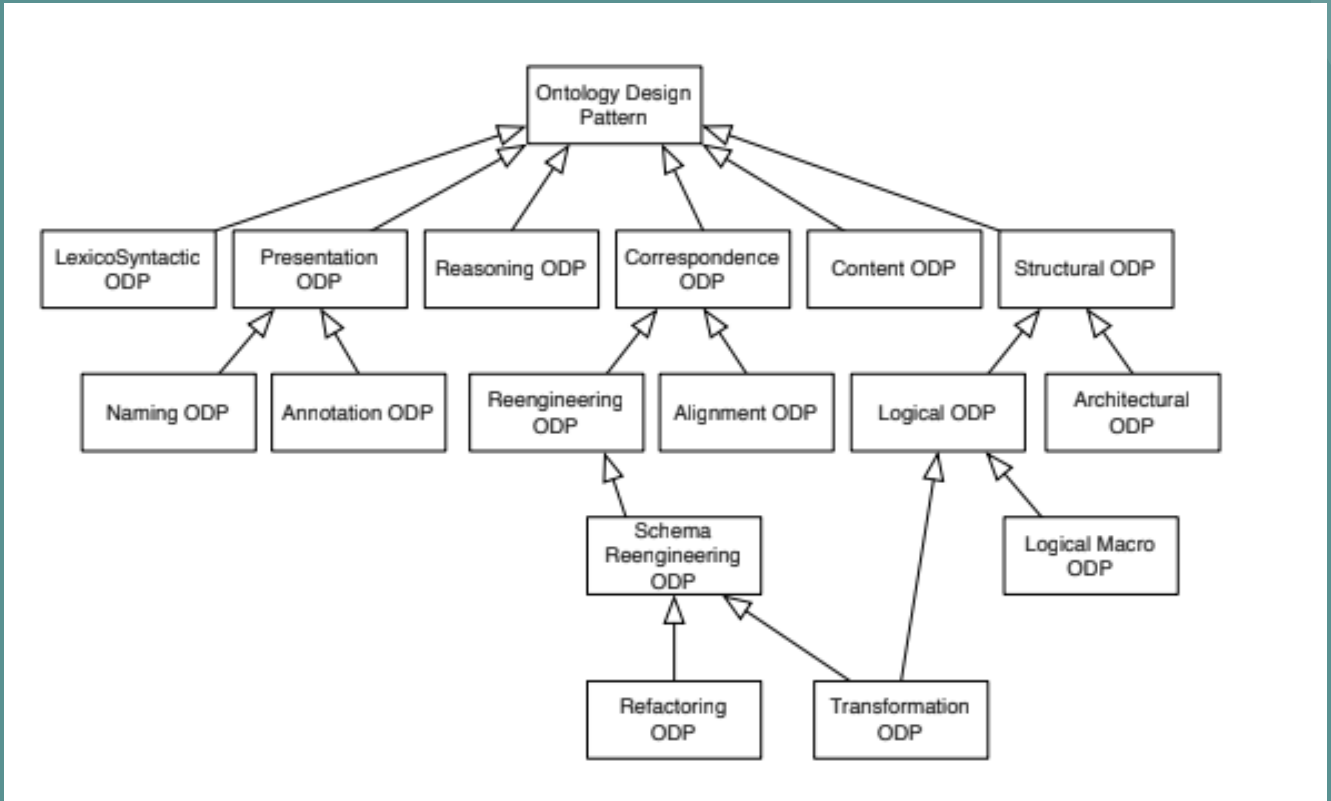
Step 5. Integration

Step 6. Implementation

Step 7. Evaluation

Domain Experts involvement except in Implementation and integration

Ontology Design Pattern

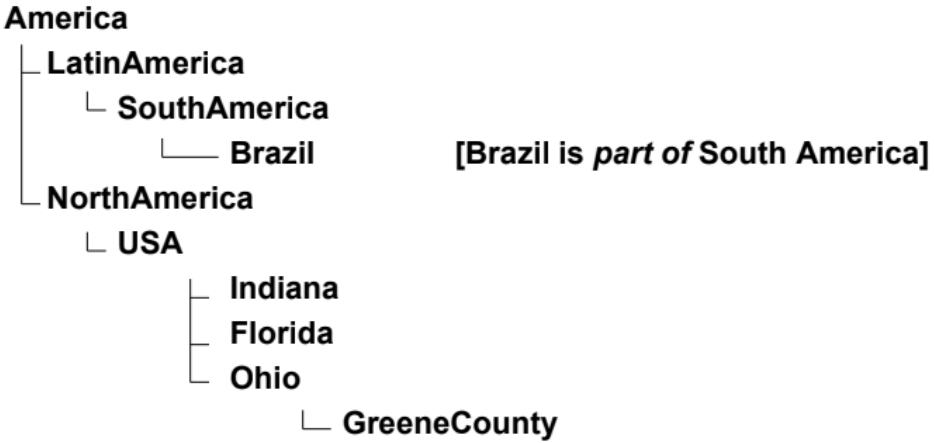


Ontology Design Pattern

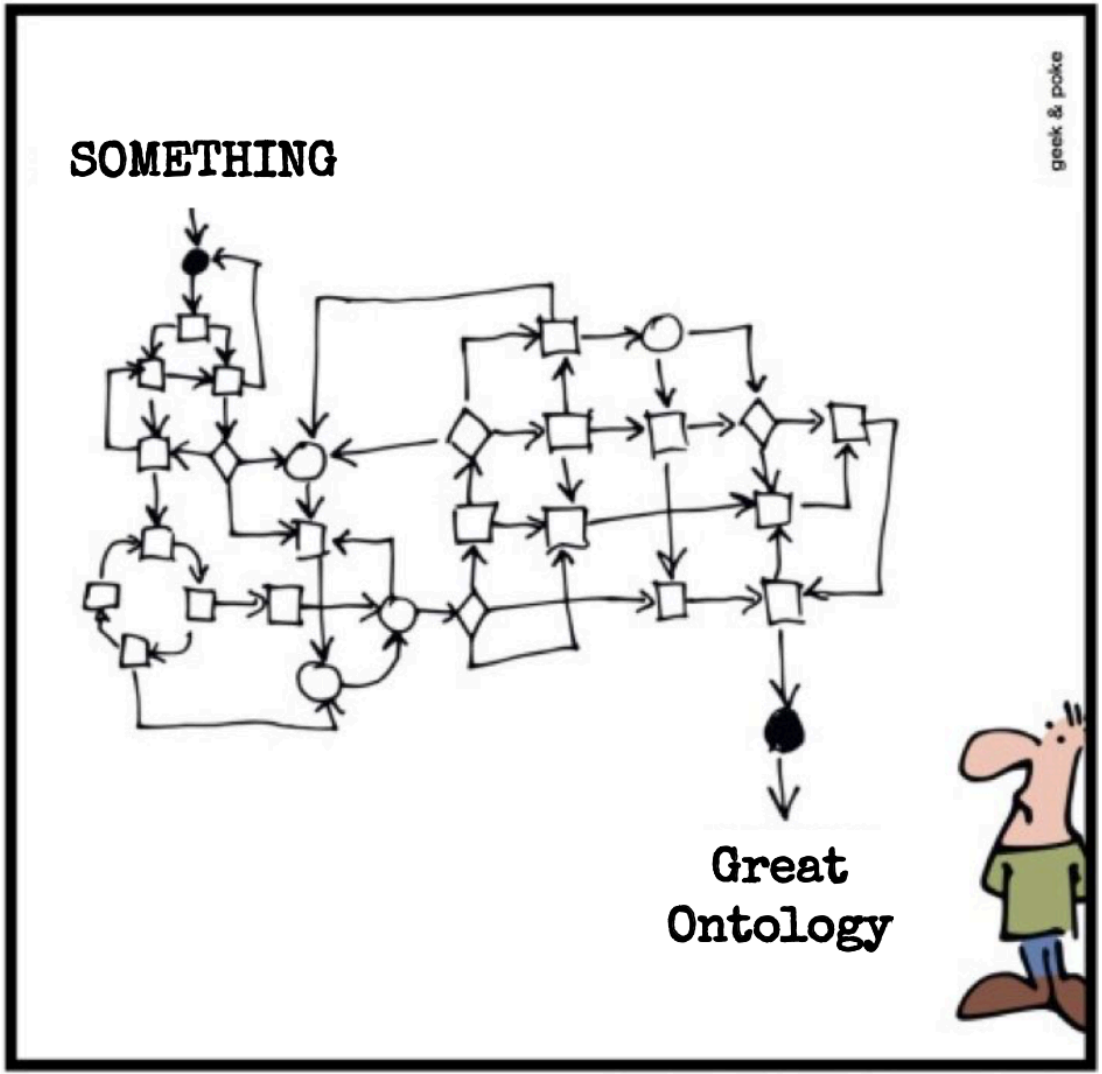
- An ontology design pattern is a reusable successful solution to a recurrent modeling problem
- Ontology Design Patterns (ODPs) help domain experts in ontology engineering work, by allowing them to reusable best practices.
- One of the first patterns published was about how to model part–whole relationships in Web Ontology Language (OWL).
- Another pattern which is frequently used is how to represent n-ary relations in OWL
- Others include how to represent classes as property values as well as specified values and value sets

Ontology Design Pattern (Part-Whole Relation)

– classes of things, arranged in a hierarchy of “part-of” relationships



Example


SOMETHING
geek & poke
Great
Ontology

Ontology Development 101

- in practice 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

„There is no one correct way to model a domain.
There are always viable alternatives.“

Determine Domain & Focus

- Which **domain** should be covered by the ontology?
- **What** should the ontology be used **for**?
- What types of **Questions** should be answered by the knowledge represented i the ontology?
- **Who** will use and maintain the ontology?
- Formulation of **Competence Questions**

Determine Domain & Focus

Competence Questions *(Wine Ontology)*

- *Which properties of the wine should be considered for modelling?*
- *Is Bordeaux a white wine or a red wine?*
- *Does a Sauvignon Blanc match with fish?*
- *Which wine matches best for grilled vegetables?*
- *Which properties of a wine do influence whether it matches with a specific dish?*
- *Does the bouquet of a wine ch*
- *Does th*
- *...*

These Questions might change within the ontology life cycle

Consider Reuse

- Why should we consider reuse?
  - in order to save **cost**
  - in order to apply **tools** that are applied for other existing ontologies also for our own ontology
  - in order to reuse ontologies that have been validated by their application

If you don't find a suitable ontology or if the adaption is too complex then create a new ontology!

Develop a Terminology (Vocabulary)

- About which **concepts** are we talking?
- Which **properties** have these concepts?
- **What** do we want **to say** about these concepts?

Example: ***Wine Ontology***

- *wine, grape, winery, location,...*
- *a wine's color, body, flavor, sugar content,...*
- *subtypes of wine: white wine, red wine, Bordeaux wine,...*
- *types of food: seafood, fish, meat, vegetables, cheese,...*
- *...*

Develop Classes & Hierarchies

- **Classes** are concepts in the designated domain
  - *class of wines*
  - *class of wineries*
  - *class of red wines*
  - ...
- Classes are collections of objects with **similar properties**
- Choose a **top-down / bottom-up / middle-out** approach to model class hierarchies

Develop Classes & Hierarchies (Top Down)

Develop Classes & Hierarchies (Bottom Up)

Develop Classes & Hierarchies (Middle Out)

Define Properties

- **Properties** in a class definition describe attributes of instances
  - *every wine has a color, residual sugar, producer, etc...*

Define Property Constraints

- **Property constraints** (restrictions) describe or restrict the set of possible property values
  - *The name of a wine is a String*
  - *The producer is an instance of Winemaker...*

Definition of Instances

- Create **instances for the classes**
  - Every **class** directly becomes the **type** of its instances
  - Every **superclass** of a direct type is also type of its instances
- Create **instances for properties**, i.e. assignment of property values for the instances according to the given constraints

- *„the glass of red wine that I drank last supper...“*

Ontology Development 101

- Ontology development in practice is an **iterative Process** that **repeats continuously** and improves the ontology