

# Ontology Engineering and Upper Ontologies

Petr Křemen, Miroslav Blaško

petr.kremen@cvut.cz

December 2, 2021



# Outline

## 1 Ontology Engineering

- Basics
- Overview of methodologies
- Requirements engineering
- Selected methodological guidelines

## 2 Upper Ontologies

- Basics
- Overview of existing Upper Ontologies



- 1 **Ontology Engineering**
  - Basics
  - Overview of methodologies
  - Requirements engineering
  - Selected methodological guidelines

- 2 **Upper Ontologies**
  - Basics
  - Overview of existing Upper Ontologies

# Ontology Engineering



# Basics

## 1 Ontology Engineering

### ● Basics

- Overview of methodologies
- Requirements engineering
- Selected methodological guidelines

## 2 Upper Ontologies

### ● Basics

- Overview of existing Upper Ontologies



## Basic terms

- **Ontology Engineering** is research field that covers a set of activities appearing during conceptualization, design, implementation and deployment of ontologies. It includes topics such as philosophy, metaphysics, knowledge representation formalisms, development methodology, knowledge sharing and reuse, knowledge management, systemization of domain knowledge, standardization, evaluation etc.
- **Ontology life cycle** is the specific sequence of activities that are carried out by ontology practitioners for developing an ontology.
- **Ontology life cycle model** is a framework that provides general structure on which activities of the ontology life cycle can be mapped.



# Ontology Engineering activities

List of Ontology Engineering activities :

- Ontology Conceptualization
- Ontology Integration
- Ontology Population
- Ontology Elicitation
- Ontology Learning
- Ontology Localization
- Ontology Matching
- Ontology Search
- O. Requirements Specification
- Ontology Specialization
- Ontology Diagnosis
- Ontology Repair
- Ontology Modularization
- Ontology Reengineering
- Ontology Reuse
- Ontology Versioning
- Ontology Verification
- Ontology Validation
- ...

Comprehensive glossary of Ontology Engineering activities can be found at

<http://mayor2.dia.fi.upm.es/oeg-upm/files/pdf/NeOnGlossary.pdf>



# Ontology life cycle models

There are multiple models to sequence activities for development of ontology :

- Waterfall Model
- "V" Model
- Incremental Model
- Iterative Model
- Evolutionary Prototyping
- Rapid Throwaway Prototyping
- Spiral Model



# Overview of methodologies

- 1 **Ontology Engineering**
  - Basics
  - **Overview of methodologies**
  - Requirements engineering
  - Selected methodological guidelines

- 2 **Upper Ontologies**
  - Basics
  - Overview of existing Upper Ontologies





# Categories of activities covered by methodologies

Taken from book Gómez-Pérez et.al, Methodologies and methods for building ontologies,2004.



# METHONDOLOGY

- **METHONDOLOGY** was developed at Polytechnic University of Madrid and is based on IEEE standards for Developing Software Life Cycle Processes.
- It provides guidelines for
  - **Project management process** – planning, project control, quality control etc.
  - **Ontology development process** – envisioned use of the ontology, explication of the envisioned users, conceptualization of target domain, formalization of ontology, implementation etc.
  - **Support activities** – knowledge acquisition, evaluation, ontology integration, documentation, version management etc.



# The NeOn Methodology – introduction

- **The NeOn Methodology** [suarez2010neon] is scenario-based methodology with emphasis on
  - reuse and reengineering of knowledge aware resources
  - collaborative and argumentative ontology development
  - building of ontology networks, as opposed to custom-building new ontologies from scratch.
- It defines following scenarios
  - 1 From specification to implementation
  - 2 Reusing and re-engineering non-ontological resources (NORs)
  - 3 Reusing ontological resources
  - 4 Reusing and re-engineering ontological resources
  - 5 Reusing and merging ontological resources
  - 6 Reusing, merging and re-engineering ontological resources
  - 7 Reusing ontology design patterns (ODPs)
  - 8 Restructuring ontological resources
  - 9 Localizing ontological resources



# The NeOn Methodology – overall architecture

Available at <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/methodologies/59-neon-methodology>



## Other methodologies

- On-To-Knowledge
- Cyc methodology
- Grüninger and Fox's methodology
- Uschold and King's methodology
- KACTUS methodology
- SENSUS methodology
- DILIGENT methodology
- TOVE methodology
- Activity-First Method (AFM) in Hozo



## Three-layer model of guidelines

An ontology engineering methodology can be composed of three-layers guidelines [mizoguchi2004tutorial] :

- **Top-layer** – The whole building process compliant with the conventional software development process (The methodologies described so far includes mostly only this level guidelines and partially the next level)
- **Middle-layer** – Generic constraints and guidelines which specify major steps (AFM is example of a methodology mainly concerned with this level)
- **Bottom-layer** – The most fine-grain guidelines such as those for class identification process, etc. (Ontology development tutorial by Noy [noy2001ontology] have few guidelines at this level)

### Note

Middle-layer and bottom-layer guidelines are essential for novices to develop "correct ontology" as they directly influence the quality of the ontology developed.

## Summary of methodologies

- *Cyc, Uschold and King, Grüninger and Fox, KACTUS, METHONDOLOGY, SENSUS, On-To-Knowledge* methodologies can be used to build ontologies from scratch.
- *METHONDOLOGY, On-To-Knowledge methodology* are very useful in development process management.
- *Uschold and King methodology* is useful for obtaining informal ontology in early phase of development.
- *Activity-First Method methodology* is useful for building task and domain ontologies.



# Requirements engineering

- 1 **Ontology Engineering**
  - Basics
  - Overview of methodologies
  - **Requirements engineering**
  - Selected methodological guidelines

- 2 **Upper Ontologies**
  - Basics
  - Overview of existing Upper Ontologies





## Requirements engineering basics

- **Requirement engineering** is an activity in which we view ontology as black box while trying to describe requirements that the ontology should fulfill.
- Requirements are of two types :
  - **Functional requirements** describing internal structure and content
    - Query results
    - Inferences
    - Error checking
    - ...
  - **Non-functional requirements** describing overall structure, acceptance, guidelines and rules for development etc.
    - Coverage
    - Efficiency
    - Documentation
    - ...
- We will focus here rather on "computational ontologies" (as part of a software system performing some task)



## Requirements engineering – Functional requirements

There are three types of functional requirements that are used within requirement engineering :

- 1 **Competency question (CQ)** is natural language question that the ontology should be able to answer
  - *Simple lookup queries* – Who are participant of certain event ?
  - *Queries expressing inferences and constraints* – If people have children, is a specific person grandparent or not ? Is it valid to have 3 parents ?
- 2 **Contextual statement** is an axiom that must hold in the ontology expressed in natural language – Every person has exactly 2 parents. Grandparent is person who has a child who has a child.
- 3 **Reasoning (inference) requirement** specify the input and output data for a reasoning task – we need to query directly for all the grandparents

### Note

Contextual statements and reasoning requirements are used to clarify/remove complex CQs

# Selected methodological guidelines

- 1 **Ontology Engineering**
  - Basics
  - Overview of methodologies
  - Requirements engineering
  - **Selected methodological guidelines**

- 2 **Upper Ontologies**
  - Basics
  - Overview of existing Upper Ontologies



## Guideline to ontology requirement specification

Based on NeOn methodology ontology requirement specification can be formed from a set of ontological needs as follows :

- 1 Identify the purpose, scope and implementation language
- 2 Identify the intended end-users
- 3 Identify the intended uses
- 4 Identify functional and non-functional requirements
- 5 Group requirements (the list of CQs)
- 6 Validate set of requirements – *if not valid jump to step 4*
- 7 Prioritize requirements (optional)
- 8 Extract terminology and its frequency



## Guideline for ontology life cycle models

Based on assumption about ontology requirements we can use following ontology life cycle models :

- Ontology requirements are known from the beginning
  - Waterfall Model
  - "V" Model
  - Incremental Model
  - Iterative Model
- Ontology requirements are not known from the beginning and can change during the project
  - Evolutionary Prototyping
  - Rapid Throwaway Prototyping
- Uncertainties in the ontology requirements can derive into risks in the project
  - Spiral model



# Guidelines for taxonomy construction direction

- There are three approaches to construct taxonomies :
  - top-down approach
  - bottom-up approach
  - middle-out approach
- Try to combine the approaches as much as possible !



# Guideline for reusing ontological resource

- There are multiple ways how to reuse ontological resources :
  - ontologies as wholes
  - ontology modules
  - ontology design patterns
  - ontology statements
- In addition to reuse, reconstruction, re-engineering, and merging can be used.



- 1 Ontology Engineering
  - Basics
  - Overview of methodologies
  - Requirements engineering
  - Selected methodological guidelines

- 2 Upper Ontologies
  - Basics
  - Overview of existing Upper Ontologies

# Upper Ontologies





# Motivation – Reuse of Ontological Resources

- Types of ontologies:
  - top-level (upper) ontologies
  - domain ontologies and task ontologies
  - application ontologies
- Ways to reuse ontological resources:
  - ontologies as wholes
  - syntactic/semantic ontology modules
  - ontology design patterns
  - ontology statements



# Basics

- 1 **Ontology Engineering**
  - Basics
  - Overview of methodologies
  - Requirements engineering
  - Selected methodological guidelines
- 2 **Upper Ontologies**
  - **Basics**
  - Overview of existing Upper Ontologies



# What Are Upper Ontologies ?

- **Upper ontologies** (sometimes also called *top-level* or *foundational* ontologies) describe very general concepts that are independent of particular problem or domain.
- They provide categories of kinds of things and relations that can provide a basic structure for “lower-level“ ontologies such as domain ontology.



# Why Should We Use Upper Ontologies ?

- Pros:
  - "top-down approach" and modelling guidance for ontology development
  - basic categories and relations that we don't need to reinvent again
  - interoperability among ontologies
- Cons:
  - a lot of effort needed to understand
  - too abstract



# Basic Ontological Commitments

- Universals vs. Particulars – Universals can have instances, while Particulars don't
- Descriptive vs. Realist – represent world using natural language and common sense vs. represent it as is
- Multiplicative vs. Reductionist – different objects can be co-located at the same time vs. only one object may be located at the same region at one time
- Endurantism vs. Perdurantism – an object is wholly present at all times vs. an object has temporal parts
- Actualism vs. Possibilism – everything that exists in the ontology is real vs. objects are allowed independent of their actual existence
- Concrete & Abstract entities – entities that exist in space and time & entities that exist neither in space nor time



# Overview of existing Upper Ontologies

- 1 **Ontology Engineering**
  - Basics
  - Overview of methodologies
  - Requirements engineering
  - Selected methodological guidelines

- 2 **Upper Ontologies**
  - Basics
  - Overview of existing Upper Ontologies



# Existing Upper Ontologies

- UFO (Unified Foundational Ontology)
- BFO (Basic Formal Ontology)
- DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)
- SUMO (Suggested Upper Merged Ontology)
- YOMATO (Yet Another More Advanced Top-level Ontology)
- GFO (General Formal Ontology)
- PROTON (PROTo ONtology)
- Cyc
- ?WordNet



# Comparison of Ontological Commitments

Comparison of ontological commitments within selected upper ontologies taken from  
<http://www.thezfiles.co.za/ROMULUS/ontologicalCommitments.html>





## DOLCE overview

- **D**escriptive **O**ntology for **L**inguistic and **C**ognitive **E**ngineering<sup>1</sup>
- developed by researchers from the Laboratory of Applied Ontology, headed by N. Guarino
- first module of the WonderWeb Foundational Ontologies Library
- ontology of particulars, multiplicative, possibilism
- strong cognitive/linguistic bias – descriptive attitude with categories mirroring cognition, common sense, and the lexical structure of natural language

---

<sup>1</sup>Home page – <http://www.loa.istc.cnr.it/old/DOLCE.html>,  
online term search –

[https://www.w3.org/2001/sw/BestPractices/WNET/DLP3941\\_daml.html](https://www.w3.org/2001/sw/BestPractices/WNET/DLP3941_daml.html)



# DOLCE's Taxonomy of Basic Categories



# DOLCE's Examples of "Leaf" Basic Categories

<b>"Leaf" Basic Category</b>	<b>Examples</b>
<i>Abstract Quality</i>	the value of an asset
<i>Abstract Region</i>	the (conventional) value of 1 Euro
<i>Accomplishment</i>	a conference, an ascent, a performance
<i>Achievement</i>	reaching the summit of K2, a departure, a death
<i>Agentive Physical Object</i>	a human person (as opposed to legal person)
<i>Amount of Matter</i>	some air, some gold, some cement
<i>Arbitrary Sum</i>	my left foot and my car
<i>Feature</i>	a hole, a gulf, an opening, a boundary
<i>Mental Object</i>	a percept, a sense datum
<i>Non-agentive Physical Object</i>	a hammer, a house, a computer, a human body
<i>Non-agentive Social Object</i>	a law, an economic system, a currency, an asset
<i>Physical Quality</i>	the weight of a pen, the color of an apple
<i>Physical Region</i>	the physical space, an area in the color spectrum, 80Kg
<i>Process</i>	running, writing
<i>Social Agent</i>	a (legal) person, a contractant
<i>Society</i>	Fiat, Apple, the Bank of Italy
<i>State</i>	being sitting, being open, being happy, being red
<i>Temporal Quality</i>	the duration of World War I, the starting time of the 2000 Olympics
<i>Temporal Region</i>	the time axis, 22 june 2002, one second



# DOLCE Basic Relations

- parthood (immediate vs. temporary)
- constitution
- participation
- representation
- specific/generic constant dependence
- inherence (between a quality and its host)
- quale (immediate vs. temporary)



# DOLCE's Primitive Relations Between Basic Categories



# DOLCE's Relations About Qualities

Example of quality and quality region – there is a difference between “this rose is red” (red-thing) and “the color of this rose is red” (red-color)



# BFO Overview

- **B**asic **F**ormal **O**ntology<sup>2</sup>
- developed in Saarland University mainly by B.Smith, P.Grenon
- designed for use in supporting information retrieval, analysis and integration in scientific and other domains
- realistic and reductionist view of the world, actualism
- limited granularity
- contains both SNAP (endurants) and SPAN (perdurants) sub-ontologies

---

<sup>2</sup><http://ifomis.uni-saarland.de/bfo/>



# BFO's Taxonomy of Basic Categories





# BFO's realizable entity example



## Other interesting upper ontology resources

- ONSET: the foundational ONTology Selection and Explanation Tool – <http://www.meteck.org/files/onset>
- Ontology browser integrated mainly through BFO – <http://www.ontobee.org>
- SUMO Concept hierarchy and search – <http://virtual.cvut.cz/kifb/en/toc/root.html>
- YOMATO ontology description – [http://download.hozo.jp/onto\\_library/YAMATO101216.pdf](http://download.hozo.jp/onto_library/YAMATO101216.pdf)
- UFO related community portal – <https://ontouml.org>
- DOLCE ontology descripton – <http://www.loa.istc.cnr.it/old/Papers/DOLCE2.1-FOL.pdf>

