1 Introduction

1.1 Ontology Engineering

1.1.1 Basics

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

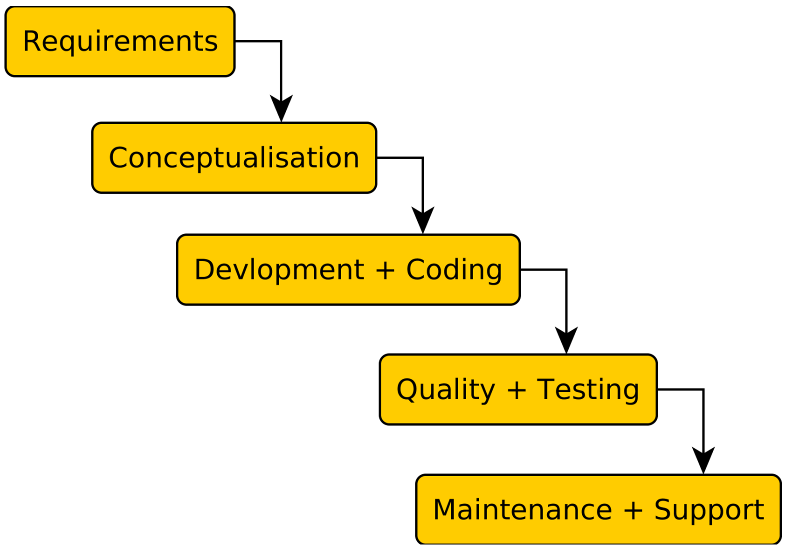
- Ontology Repair
- Ontology Modularization
- Ontology Reengineering
- Ontology Reuse
- Ontology Versioning
- Ontology Verification
- Ontology Validation
- ...

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



1.1.2 Requirements engineering

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 :

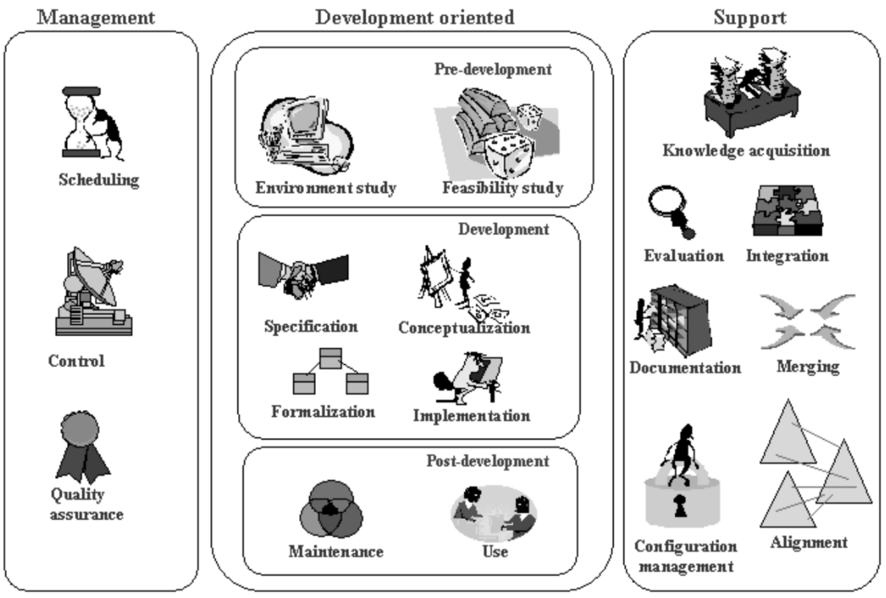
1. **Competency question** (CQ) is natural language question that the ontology should be able to answer
   - *Simple lookup queries* – Who are participant of certain event ?
   - *Queries expressing inferences and constraints* – If people have children, is a specific person grandparent or not ? Is it valid to have 3 parents ?
2. **Contextual statement** is an axiom that must hold in the ontology expressed in natural language – Every person has at 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

1.1.3 Overview of methodologies

Categories of activities covered by methodologies



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

METHONDOLOGY

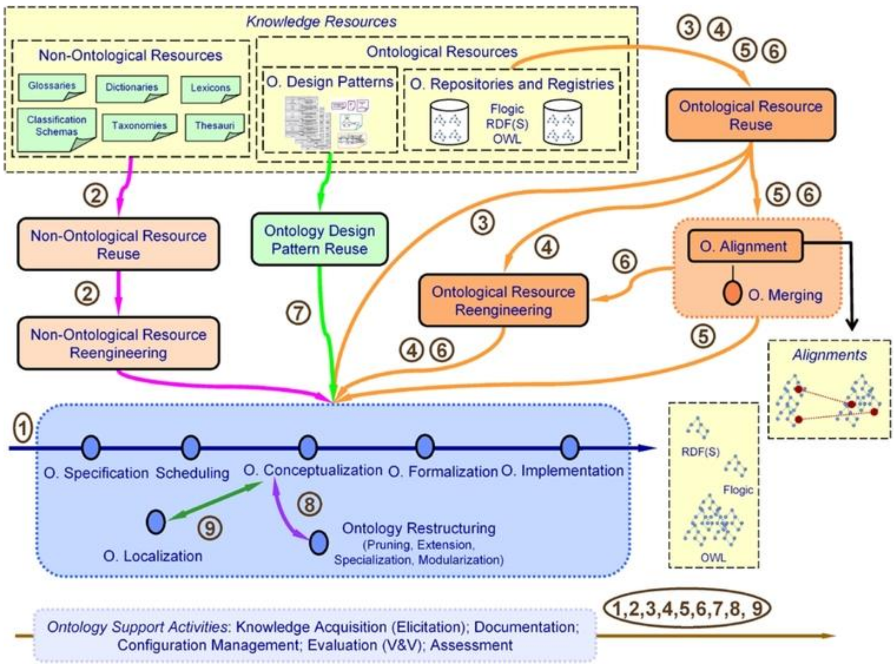
- **METHONDOLOGY** was developed at Polytechnic University of Madrid and is based on IEEE standards for Developing Software Life Cycle Processes.
- It provides guidelines for
  - **Project management process** – planning, project control, quality control etc.
  - **Ontology development process** – envisioned 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


Knowledge Resources
Non-Ontological Resources
Glossaries
Dictionaries
Lexicons
Classification Schemas
Taxonomies
Thesauri
Ontological Resources
O. Design Patterns
O. Repositories and Registries
Flogic
RDF(S)
OWL
Ontological Resource Reuse
Non-Ontological Resource Reuse
Ontology Design Pattern Reuse
Non-Ontological Resource Reengineering
Ontological Resource Reengineering
O. Alignment
O. Merging
Alignments
O. Specification
Scheduling
O. Conceptualization
O. Formalization
O. Implementation
O. Localization
Ontology Restructuring
(Pruning, Extension, Specialization, Modularization)
RDF(S)
Flogic
OWL
Ontology Support Activities: Knowledge Acquisition (Elicitation); Documentation; Configuration Management; Evaluation (V&V); Assessment
1,2,3,4,5,6,7,8, 9

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

Other methodologies

- 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.

1.1.4 Selected methodological guidelines

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.