Advanced Topics in Ontology Engineering

Stepwise Methodologies

Dr. Mustafa Jarrar

Sina Institute, University of Birzeit

mjarrar@birzeit.edu
www.jarrar.info

Methodology

➔What are the phases of the ontology development life cycle? taking into account that ontologies might be built collaboratively by many people.

Let's discuss from where to start, if you want to build an ontology for:

- E-government
- E-Banking
- E-Health
- Bioinformatics
- Multilingual search engine
- …

Methodological Questions

- Which tools and techniques to use?
- Which languages should be used in which circumstances, and in which order?
- What quality measures should we care about?
- What things can be reused?
- Which people should be assigned which tasks?
- ....

- **Many Methodologies exist ! But non is good!** Because each project/application/domain is different, and the background of the people involved are also different, etc.

- We will overview some common steps in this lecture, thus try to learn smartly, and don't follow these steps literally. **You should have your own methodology for each ontology.**

Most methodologies propose these phases:

1- Identify Purpose and Scope

2- Building the Ontology

- 2.1- Ontology Capture
- 2.2- Ontology Coding

3- Integrating existing ontologies

4- Evaluation

5- Documentation

1- Purpose and Scope

- There is no one/ideal ontology of a certain domain
  - There are always alternatives, each abstracting different things, and for different usages.

- What should be included in the ontology (concepts and relations) should be smartly determined, taking into account (if possible) many application scenarios.
    - Interoperability between systems.
    - improve search quality.
    - Communication between people and organizations (important).
    - …

  - Future extensions should be anticipated.

1- Purpose and Scope

- When you specify the purpose and scope, you should specify the following:

1- What is the domain that the ontology will cover?

➔The notion of *context,* in the double articulation theory, is part of the Purpose and Scope.

That is: the scope where the vocabulary interpretation should be valid.

For example: the scope of the legal-Person ontology is the set of all laws, regulations, and repositories in the state.

2- What we are going to use the ontology for?

➔ Enough description about what application scenarios are taking into account.

Be carful with the ontology usability/reusability trade-off

2- Building the Ontology

2.1- Ontology Capture

- Identify key concepts and relationships.
- Produce clear text definitions for these concepts (i.e., **glosses**).
- Identify terms that refer to these concepts.
- Reach Consensus (Consensus is an indication of correctness).

➔ You may apply the 7 steps for building an ORM schema, somehow!

2.2- Ontology Coding/Specification/Characterization

- Explicit representation of the “conceptualization” in some formal language.

2.1- Ontology Capture: Scoping

- **Brainstorming**
  - Produce all potentially relevant terms and phrases.
    - Nouns form the basis for concept names
    - Verbs (or verb phrases) form the basis for property and names.

    This step can be semi- automated somehow, as candidate concepts and relations can be extracted automatically from relevant documents, laws, forms, DB schemes....

- **Organize candidate concepts into groups**

  Group related terms together.
  - Exclude some terms if not relevant (w.r.t., purpose and scope)
  - Keep notes of these decisions.
  - Group similar terms and potential synonyms together.

2.1- Ontology Capture: Produce Definitions

- **Use suitable meta-ontology**
  - i.e., use modeling primitives in a consistent manner (e.g. Type, role, entity, instance, relationship...)
- **When several people are involved, each might be responsible on a group of terms**
  - Semantic overlap with others must be right in the first place, otherwise lot of redundant re-working.
- **Terms: Produce definitions/glosses in a middle-out fashion**
  - Define a gloss for each term. This helps get deeper understanding of the domain.
  - These glosses will have to be revised later, after defining the relationships/ subsumptions between concepts.
  - This is called middle-out, rather than top-down or bottom up. – will be discussed later.

Define Taxonomy

- **Relevant terms must be organized in a taxonomic hierarchy (i.e., subsumptions)**
  - Opinions differ on whether it is more efficient to do this in a top-down or a bottom-up fashion.

- **Ensure that hierarchy is indeed a taxonomy:**
  - If A subsumes B, then every instance of A must also be a subsume B (compatible with semantics of **rdfs:subClassOf**)
  - Insuring the correctness of subsumptions needs philosophical thinking (apply the OntoClean Methodology).

- The semantics of subsumption demands that whenever A subsumes B, every property that holds for instances of B must also apply to instances of A (called **inheritance**).
  - It makes sense to attach properties to the highest class in the hierarchy to which they apply.

Define Properties

- Determine the relevant properties for each concept. Such properties must be essential –to describe the meaning-, or relevant to the applications.

- While attaching properties to concepts, it is useful to determine its range (its datatype/value, or relations with other concepts).

Add Rules and Restrictions

- Cardinality Restrictions
- Which properties should be unique, mandatory, disjunctions, restricted values…etc.
- Relational Characteristics
  - symmetry, transitivity, inverse properties, functional values

➔ You must avoid the situation that the added rules are DB integrity constraints.

➔ Some/all rules should be verbalized –in pseudo natural language sentences- so to enable other people review it and give feedback.

Define Some Important Instances

- Some important instances (might) be added to the ontology, if needed. Such entities can be:
  - Country: Palestine
  - Person: Arafat
  - Capital: Jerusalem

- ➔ in case of a large instances, it is more convenient to have them separately .

  - See the Entity and Address servers in Zinnar

Advantages of the Middle-out Approaches

- A bottom-up approach results in a high degree of detail
  - increases overall effort
  - makes it difficult to spot commonality between related concepts.
  - increases risk of inconsistencies and re-work.
- Top-down allow better control of degree of detail
  - risk of arbitrary high-level categories
  - risk of limited stability
- Middle-out strikes is a compromise, but it allow the ontology evolve gradually, you need to come back to some steps.
- The higher level concepts naturally arise and are thus more likely to be stable.

Reaching Agreement: Some suggestions

➔ Ontologies are made to be agreed and shared, thus it is VERY important to make sure that people agree on them.

➔ How to facilitate reaching agreement?

• Produce a natural language text definitions.

- Ask domain experts to review the context, glosses, verbalized rules, and the ontology itself in a graphical/diagramatic form.

• Ensure consistency with terms already in use

– use existing thesauri and dictionaries

– avoid introducing new terms in the definitions

• Indicate relationships with other commonly used terms

– synonyms, variants, such referring to different dimensions

• Give examples

Integrating Existing Ontologies

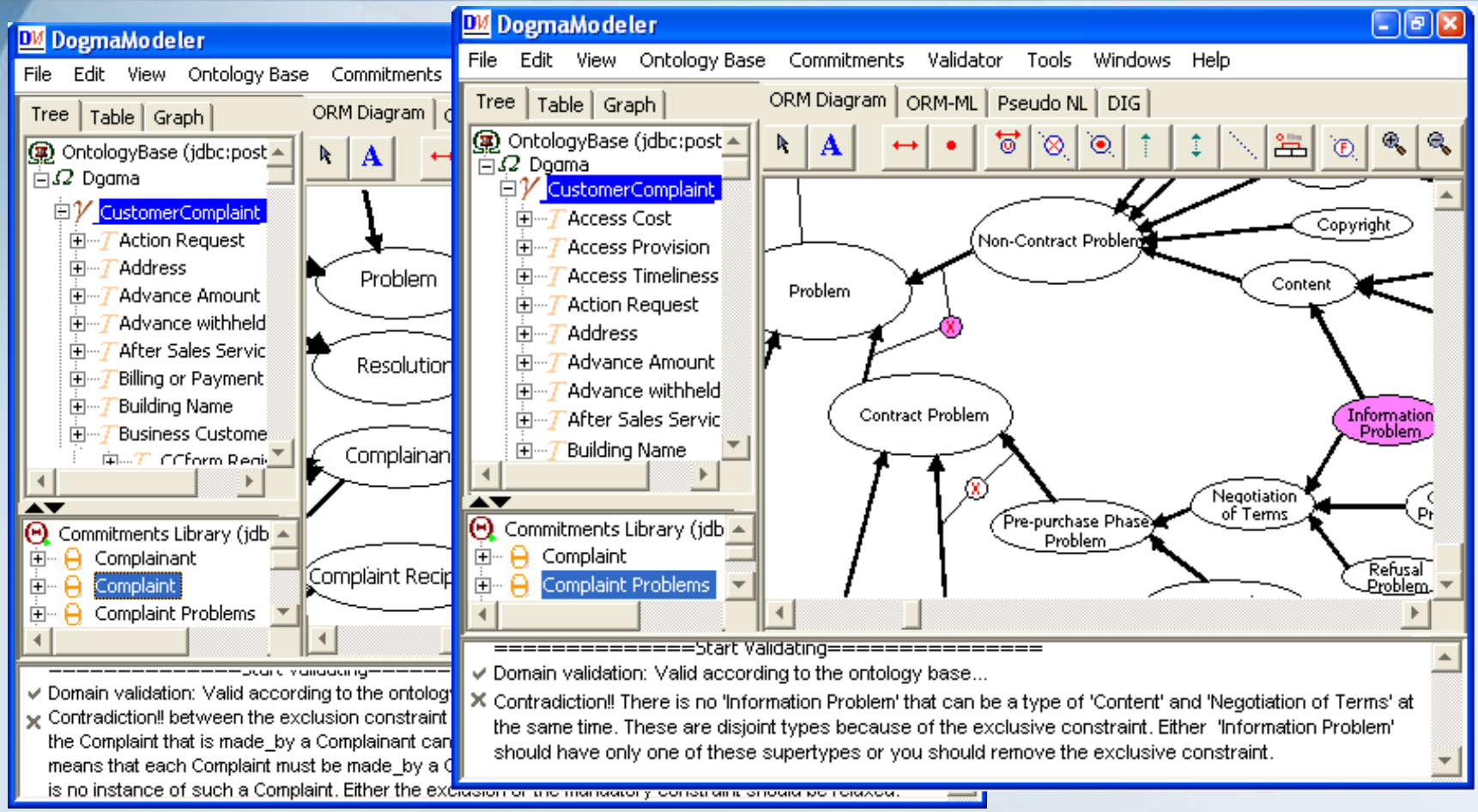
- Check overlap with existing ontologies
- Establish formal links
  - Produce mappings to existing concept definitions
  - Import and extend existing ontologies
- Avoid re-inventing the wheel!

Ontology Evaluation

Several Type of evaluations:

1. **Usability Evaluation**: Validate whether the ontology produced satisfies (at least) the intended applications' requirements.
2. **Syntax evaluation**: Validate whether the ontology is well-formed w.r.t the used language.
3. **Logical evaluation**: Validate whether the ontology has axioms contradicting or implying each other.
4. **Ontological Evaluation**: Validate whether the ontology has concepts that should be instances, sub-concepts that should be roles, etc. (The OntoClean methodology is very good for this evaluation)

Check for Implications and Contradictions



Some tools exist to automatically detect logical correctness (contradictions and implications), depending on the used ontology language (Such as ORM: DogmaModeler, OWL: Racer)

Some Guidelines

Clarity: The ontology engineer should communicate effectively with the domain experts (= ask the *right* *questions*):

- Natural language definitions.
- Give examples, alternatives, and contradictions, elicit knowledge.
- emphasize distinctions.

Coherence: The ontology should be internally consistent

- Syntactically correct.
- Logically consistent.
- Ontologically consistent.

Extensibility: modularize the ontology in a way it is easy to build, understand, and maintain. What should be in a module?

Reusability and Usability: be innovative to tradeoff this smartly.

References

Mike Uschol: **Building Ontologies: Towards a Unified Methodology.** Proceedings of Expert Systems th Annual Conference of the British Computer Society Specialist Group on Expert Systems. 1996 http://www.imamu.edu.sa/Scientific_selections/Documents/IT/96-es96-unified-method.pdf

Mustafa Jarrar: **Towards methodological principles for ontology engineering**. PhD Thesis. Vrije Universiteit Brussel. (May 2005) http://www.jarrar.info/phd-thesis/

Fernández López: **Overview Of Methodologies For Building Ontologies**. Proceedings of the IJCAI99 Workshop on Ontologies and ProblemSolvingMethods Lessons Learned and Future Trends CEUR Publications. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.6002