Modelling and using ontology design patterns

Authors: E. Blomqvist, E. Daga, A. Gangemi, V. Presutti

Motivation

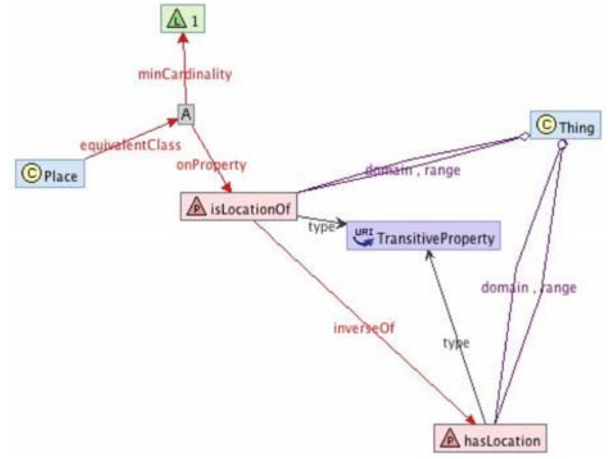
The notion of “pattern” has proved useful in design, as exemplified in diverse areas such as architecture, software engineering. Today one of the most challenging and neglected areas of ontology design is **reusability**. We propose **Ontology Design Patterns (ODPs)**, accompanied by a theoretical framework and methods to apply them in realistic ontology projects [1].

Ontology Design Patterns (ODPs)

An ODP is a modeling solution to solve a recurrent ontology design problem. It is a template that represents a schema for specific design solutions. An ODP consists of a set of “prototypical” ontology entities that constitute the “abstract form” of a pattern, and of a set of metadata about its use cases, motivations, provenance, the pros and cons of its application, the links to other patterns, etc. Design solutions based on ODPs encode ontology entities that apply, specialize, or instantiate the prototypical entities defined by the schema [2].

An example of CP: *Place*

OWL encoding of the prototypical entities from the Place CP taken from [3], which contains a catalogue of CPs expressed in OWL.

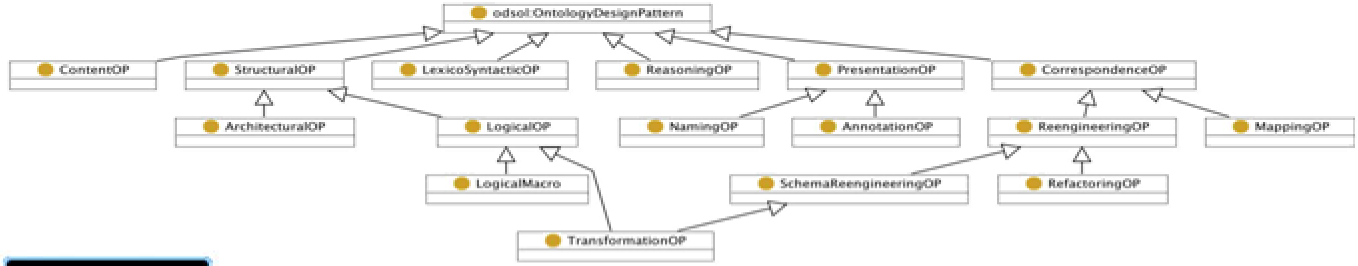


Highlight on Content ODPs (CPs)

CPs are special networked ontologies. They cover a set of competency questions (e.g., where is located an object at a certain time?), which represent the problem they provide a solution for. A CP can be extracted (cloned) from a reference ontology, reengineered from a conceptual model (e.g., a data model), and created by composition of other CPs, or by specialization of another CP [2].

ODP Pattern types

- **Structural ODPs**: include Logical ODPs - compositions of logical constructs that solve a problem of expressivity and Architectural ODPs - composition of Logical OPs.
- **Correspondence ODPs**: include Reengineering ODPs - to provide designers with solutions to the problem of transforming a conceptual model - and Mapping ODPs, for creating semantic associations between ontologies.
- **Reasoning ODPs**: applications of Logical OPs oriented to obtain certain reasoning results, based on the behavior implemented in a reasoning engine.
- **Presentation ODPs**: deal with usability and readability of ontologies from a user's perspective.
- **Lexico-Syntactic ODPs**: linguistic structures or schemas that permit to generalize and extract some conclusions about the meaning they express.
- **Content ODPs**: encode conceptual, rather than logical design patterns and propose patterns for solving design problems for the domain classes and properties that populate an ontology [2].

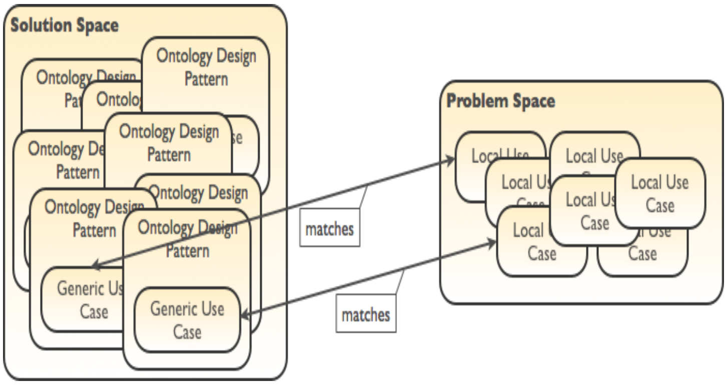


IST-2005-027595
NeOn-project.org

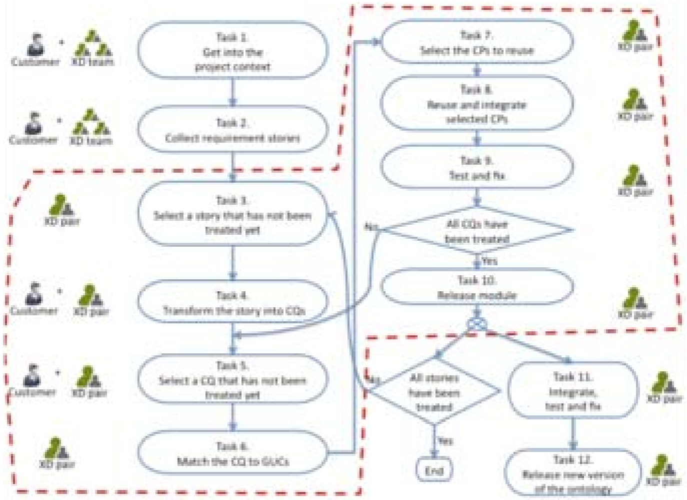
Modelling and using ontology design patterns

eXtreme Design (XD): pattern-based methodology for ontology development

XD is a family of methods based on the application, exploitation, and detection of ODPs, for solving problems related to an ontology design project. The methods are organized around a problem space, which is composed of the actual problems that have to be addressed during a project, and a solution space, which is made up of successful reusable solutions [4].



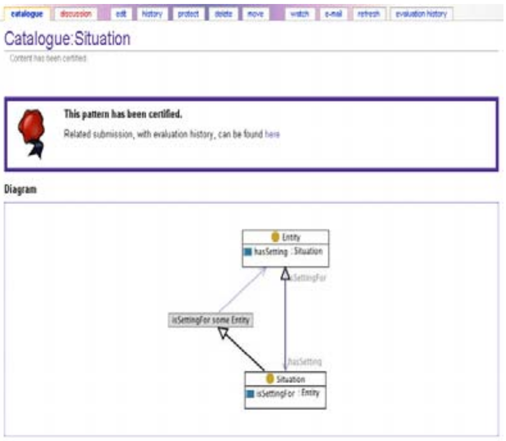
XD iteration



eXtreme Design principles

- Customer involvement and feedback
- Customer stories, competency questions
- ODP reuse and modular design
- Collaboration and Integration
- Task-oriented design
- Test-driven design
- Pair design

An ODP entry



Catalogue:Situation

Content has been certified

This pattern has been certified.

Related submission, with evaluation history, can be found here

Diagram

A certified pattern [4]

The ODP Portal: http://ontologydesignpatterns.org

”Community, shared knowledge and resources”

The ODP Portal is a portal dedicated to ontology design best practices for the semantic web, with particular focus on ODPs. The ODP Portal is a semantic wiki supporting the life cycle of ODPs, from proposals to evaluation and certification. It supports automatic forms for uploading CP submissions, semantic forms, and a unique semantic workflow.

Additional information:

Contact person: enrico.daga@cnr.it

❑ *NeOn Deliverable D2.1.1: Design Rationales for Networked Ontologies. [1].*

❑ *NeOn Deliverable D2.5.1: A Library of Ontology Design Patterns: reusable solutions for collaborative design of networked ontologies. [2].*

❑ *http://ontologydesignpatterns.org [3].*

❑ *NeOn Deliverable D2.5.2: Pattern-based ontology design: methodology and software support. [4].*