Pattern-based ontology design

Aldo Gangemi
Valentina Presutti

Semantic Technology Lab
ISTC-CNR, Roma
aldo.gangemi@cnr.it
valentina.presutti@istc.cnr.it

Outline

- Designing Computational Ontologies
- Ontology Design Patterns
- ontologydesignpatterns.org initiative

Computational ontologies

- Ontologies as (software) components, expressed and managed in standard W3C languages like RDF, OWL, RIF, SPARQL, Fresnel, etc.
- Ontology design is the core aspect
- Quality is associated with good design
- STLab people research from 2004-5: "A formal framework for ontology evaluation and selection" [5]

Quality

- Three quality dimensions: Structural-Content-Sustainability
  - *Content* is the primary dimension
- Content compliance spans Coverage-Task-SelfExplanation
  - *Task* is the immediately measurable aspect
  - Quality is not maximal and abstract, but bound to context
  - Partial orders of problems and reusable solutions (locality)
  - Good practices (history)
- Empirical methods for evaluation (measurability)

What is ontology design? 1/3

- Computational Ontologies are artifacts
    - Have a structure (linguistic, logical, etc.)
    - Their function is to “encode” a description of the world (actual, possible, counterfactual, impossible, desired, etc.) for some purpose

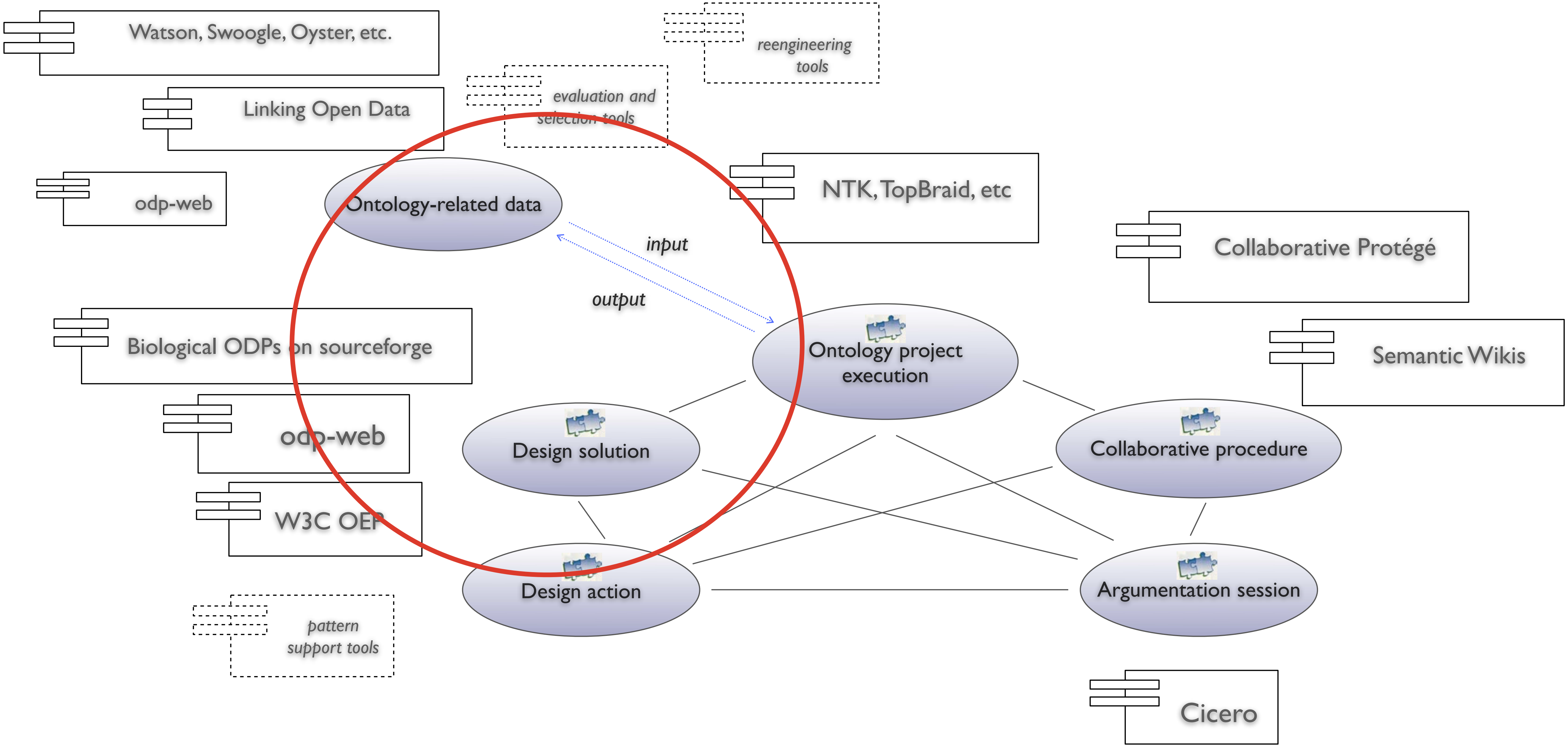
What is ontology design? 2/3

- Ontologies must match both domain and task
  - Allow the description of the entities (“domain”) whose attributes and relations are concerned because of some purpose
    - e.g. *social events* and *agents* as entities that are considered in a *legal case*, *research topics* as entities that are dealt with by a *project*, worked on by *academic staff*, and can be topic of *documents*,etc.
  - Serve a purpose (“task”), e.g. finding entities that are considered in a same legal case, finding people that work on a same topic, matching project topics to staff competencies, time left, available funds, etc.

What is ontology design? 3/3

- Ontologies have a lifecycle
  - They are created, evaluated, fixed, and exploited just like any artifact
  - Their lifecycle has some original characteristics regarding:
    - Data, Project and Workflow types, Argumentation structures, Design solutions (incl. patterns), Interaction

Design in the C-ODO key



Collaborative Ontology Design Components

Ontology-related data

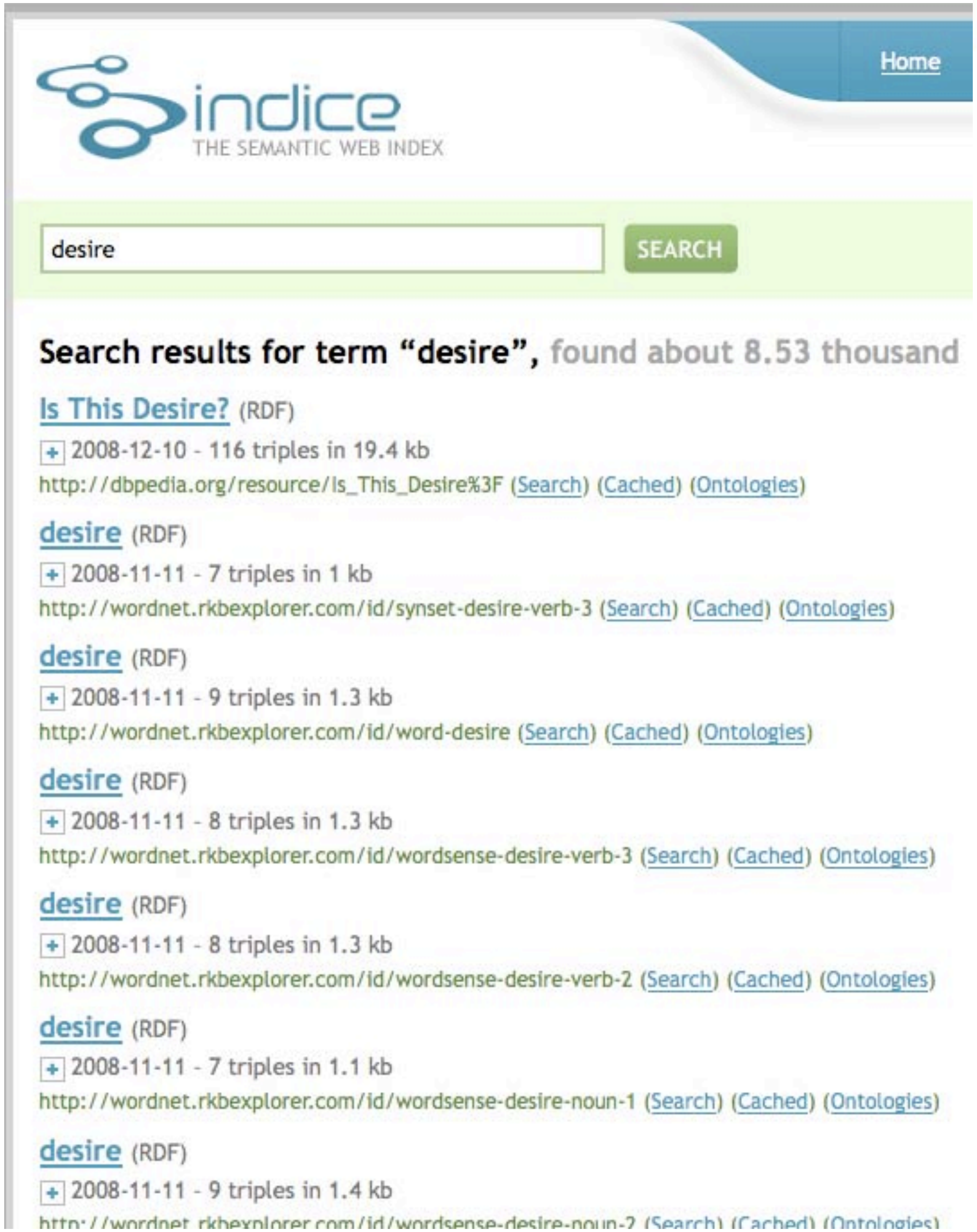
- Informal vs. formal
  - Text corpora
  - Folksonomies (tag sets, directories, topic trees, subject indexes, infoboxes)
  - Lexica (dictionaries, wordnets, terminologies, nomenclatures)
  - Knowledge organization systems (thesauri, classification schemes)
  - Frames, semantic networks
  - DB schemas
  - Linked Open Data datasets
  - (Computational) ontologies

A lot of data in the web “suq”

- Mash-ups
- Linked open data
- Wikipedia, DBpedia, Freebase, etc.
- Triplify, GRDDL, RDFa, SKOS, SIOC, etc.
- Corpora, terminologies, lexica, thesauri, “KOS”, frames, ontologies

Standard languages help

- Transform all in RDF, or even OWL
    - Cf. *Triplify* initiative
- Dataset extracted from heterogeneous sources, and triplified
- Relations are added in direct, naïve ways: Linked Open Data
    - Semantics depends on intended task of data and relations used for linking
- Then search/visualize RDF data, or make integrating applications



Search results for term “desire”, found about 8.53 thousand

Is This Desire? (RDF)
⊞ 2008-12-10 - 116 triples in 19.4 kb
http://dbpedia.org/resource/Is_This_Desire%3F (Search) (Cached) (Ontologies)

desire (RDF)
⊞ 2008-11-11 - 7 triples in 1 kb
http://wordnet.rkbexplorer.com/id/synset-desire-verb-3 (Search) (Cached) (Ontologies)

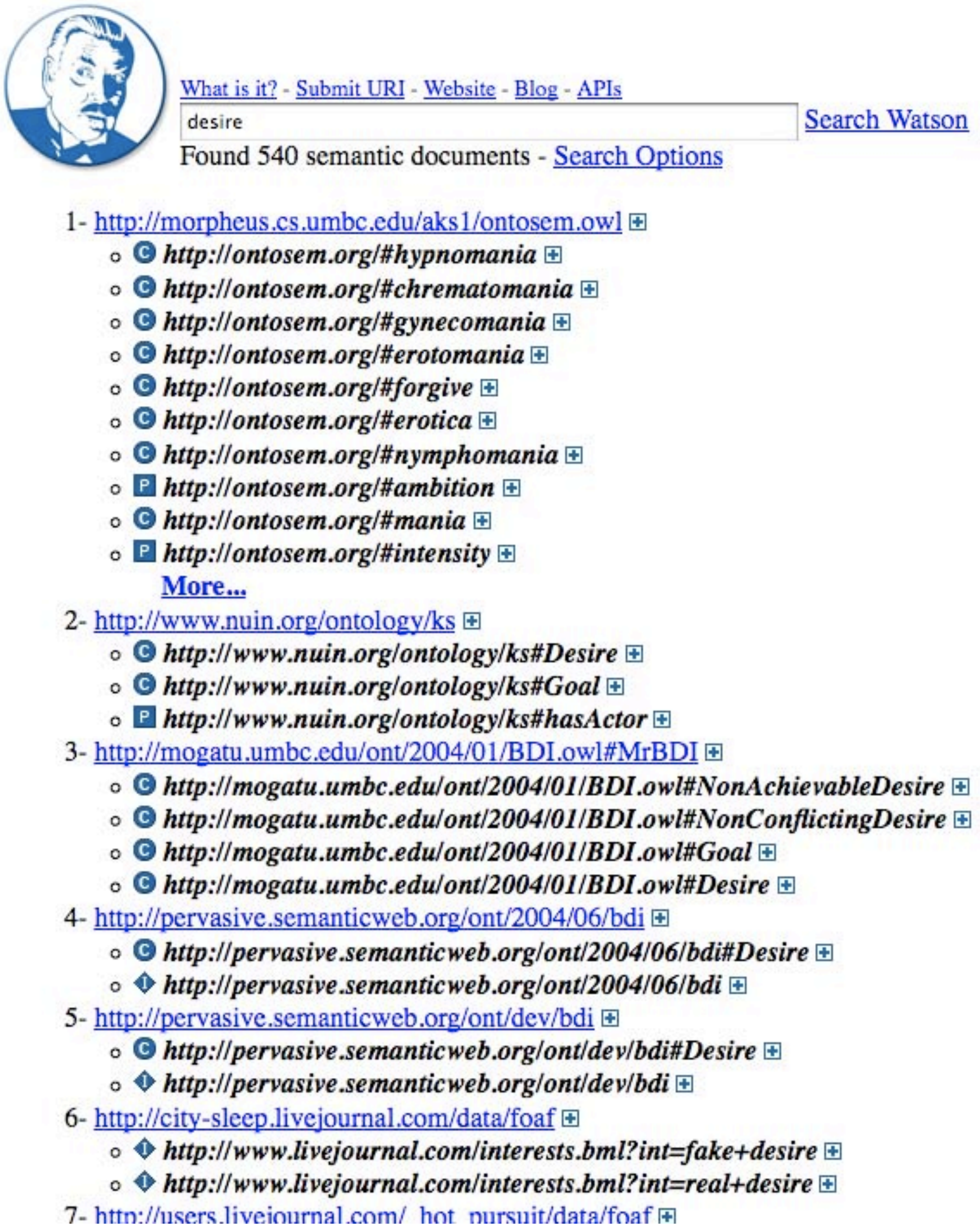
desire (RDF)
⊞ 2008-11-11 - 9 triples in 1.3 kb
http://wordnet.rkbexplorer.com/id/word-desire (Search) (Cached) (Ontologies)

desire (RDF)
⊞ 2008-11-11 - 8 triples in 1.3 kb
http://wordnet.rkbexplorer.com/id/wordsense-desire-verb-3 (Search) (Cached) (Ontologies)

desire (RDF)
⊞ 2008-11-11 - 8 triples in 1.3 kb
http://wordnet.rkbexplorer.com/id/wordsense-desire-verb-2 (Search) (Cached) (Ontologies)

desire (RDF)
⊞ 2008-11-11 - 7 triples in 1.1 kb
http://wordnet.rkbexplorer.com/id/wordsense-desire-noun-1 (Search) (Cached) (Ontologies)

desire (RDF)
⊞ 2008-11-11 - 9 triples in 1.4 kb
http://wordnet.rkbexplorer.com/id/wordsense-desire-noun-2 (Search) (Cached) (Ontologies)



Found 540 semantic documents - Search Options

1- http://morpheus.cs.umbc.edu/aks1/ontosem.owl ⊞
- C *http://ontosem.org/#hypnomania* ⊞
- C *http://ontosem.org/#chrematomania* ⊞
- C *http://ontosem.org/#gynecomania* ⊞
- C *http://ontosem.org/#erotomania* ⊞
- C *http://ontosem.org/#forgive* ⊞
- C *http://ontosem.org/#erotica* ⊞
- C *http://ontosem.org/#nymphomania* ⊞
- P *http://ontosem.org/#ambition* ⊞
- C *http://ontosem.org/#mania* ⊞
- P *http://ontosem.org/#intensity* ⊞

More...

2- http://www.nuin.org/ontology/ks ⊞
- C *http://www.nuin.org/ontology/ks#Desire* ⊞
- C *http://www.nuin.org/ontology/ks#Goal* ⊞
- P *http://www.nuin.org/ontology/ks#hasActor* ⊞

3- http://mogatu.umbc.edu/ont/2004/01/BDI.owl#MrBDI ⊞
- C *http://mogatu.umbc.edu/ont/2004/01/BDI.owl#NonAchievableDesire* ⊞
- C *http://mogatu.umbc.edu/ont/2004/01/BDI.owl#NonConflictingDesire* ⊞
- C *http://mogatu.umbc.edu/ont/2004/01/BDI.owl#Goal* ⊞
- C *http://mogatu.umbc.edu/ont/2004/01/BDI.owl#Desire* ⊞

4- http://pervasive.semanticweb.org/ont/2004/06/bdi ⊞
- C *http://pervasive.semanticweb.org/ont/2004/06/bdi#Desire* ⊞
- ◆ *http://pervasive.semanticweb.org/ont/2004/06/bdi* ⊞

5- http://pervasive.semanticweb.org/ont/dev/bdi ⊞
- C *http://pervasive.semanticweb.org/ont/dev/bdi#Desire* ⊞
- ◆ *http://pervasive.semanticweb.org/ont/dev/bdi* ⊞

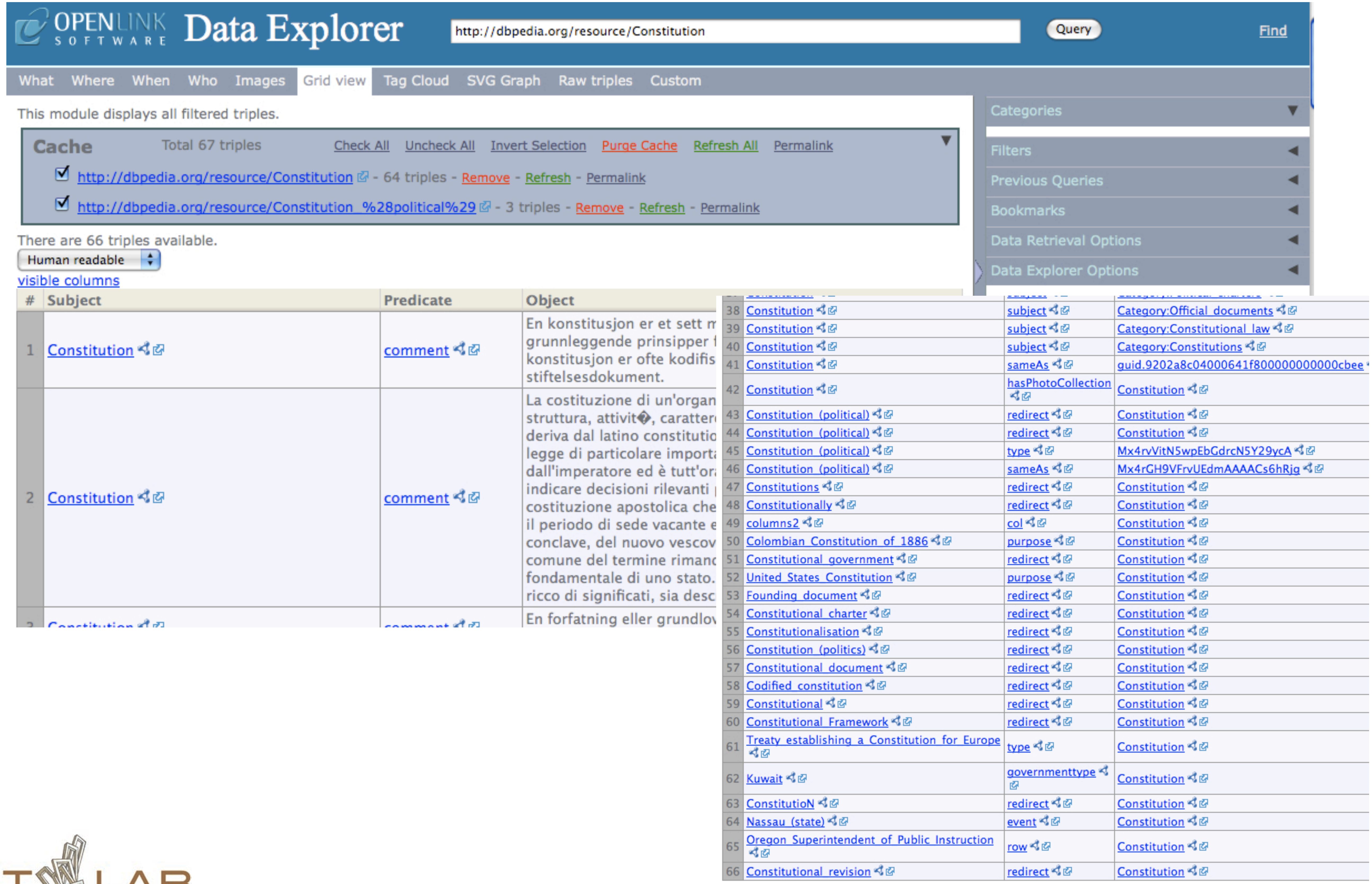
6- http://city-sleep.livejournal.com/data/foaf ⊞
- ◆ *http://www.livejournal.com/interests.bml?int=fake+desire* ⊞
- ◆ *http://www.livejournal.com/interests.bml?int=real+desire* ⊞

7- http://users.livejournal.com/_hot_pursuit/data/foaf ⊞

STLAB
SEMANTIC TECHNOLOGY LABORATORY

NeOn

Integrated knowledge search: DBpedia



This module displays all filtered triples.

Cache Total 67 triples Check All Uncheck All Invert Selection Purge Cache Refresh All Permalink

☑ http://dbpedia.org/resource/Constitution - 64 triples - Remove - Refresh - Permalink

☑ http://dbpedia.org/resource/Constitution_%28political%29 - 3 triples - Remove - Refresh - Permalink

There are 66 triples available.

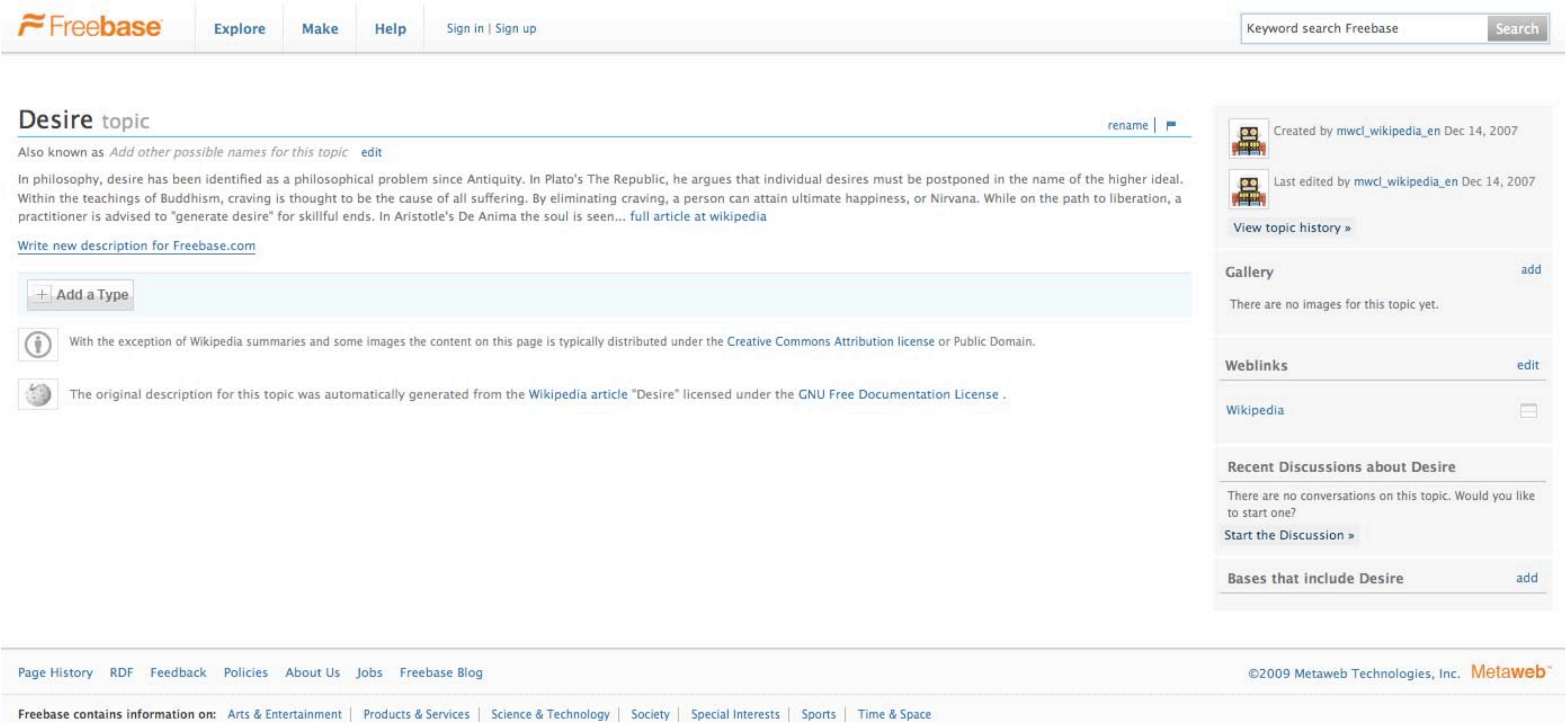
Human readable

visible columns

#	Subject	Predicate	Object
1	Constitution	comment	En konstitusjon er et sett m grunnleggende prinsipper f konstitusjon er ofte kodifis stiftelsesdokument.
2	Constitution	comment	La costituzione di un'organ struttura, attivit�, caratter deriva dal latino constitutio legge di particolare importa dall'imperatore ed è tutt'or indicare decisioni rilevanti costituzione apostolica che il periodo di sede vacante e conclave, del nuovo vescov comune del termine rimanc fondamentale di uno stato. ricco di significati, sia desc
3	Constitution	comment	En forfatning eller grundlov

#	Subject	Predicate	Object
38	Constitution	subject	Category:Official_documents
39	Constitution	subject	Category:Constitutional_law
40	Constitution	subject	Category:Constitutions
41	Constitution	sameAs	guid.9202a8c04000641f8000000000cbee
42	Constitution	hasPhotoCollection	Constitution
43	Constitution_(political)	redirect	Constitution
44	Constitution_(political)	redirect	Constitution
45	Constitution_(political)	type	Mx4rvVitN5wpEbGdrcN5Y29ycA
46	Constitution_(political)	sameAs	Mx4rGH9VFrvUEdmAAAACs6hRjg
47	Constitutions	redirect	Constitution
48	Constitutionally	redirect	Constitution
49	columns2	col	Constitution
50	Colombian_Constitution_of_1886	purpose	Constitution
51	Constitutional_government	redirect	Constitution
52	United_States_Constitution	purpose	Constitution
53	Founding_document	redirect	Constitution
54	Constitutional_charter	redirect	Constitution
55	Constitutionalisation	redirect	Constitution
56	Constitution_(politics)	redirect	Constitution
57	Constitutional_document	redirect	Constitution
58	Codified_constitution	redirect	Constitution
59	Constitutional	redirect	Constitution
60	Constitutional_Framework	redirect	Constitution
61	Treaty_establishing_a_Constitution_for_Europe	type	Constitution
62	Kuwait	governmenttype	Constitution
63	ConstitutioN	redirect	Constitution
64	Nassau_(state)	event	Constitution
65	Oregon_Superintendent_of_Public_Instruction	row	Constitution
66	Constitutional_revision	redirect	Constitution

Integrated knowledge search: Freebase



Desire topic

Also known as *Add other possible names for this topic* edit

In philosophy, desire has been identified as a philosophical problem since Antiquity. In Plato's The Republic, he argues that individual desires must be postponed in the name of the higher ideal. Within the teachings of Buddhism, craving is thought to be the cause of all suffering. By eliminating craving, a person can attain ultimate happiness, or Nirvana. While on the path to liberation, a practitioner is advised to "generate desire" for skillful ends. In Aristotle's De Anima the soul is seen... full article at wikipedia

Write new description for Freebase.com

With the exception of Wikipedia summaries and some images the content on this page is typically distributed under the Creative Commons Attribution license or Public Domain.

The original description for this topic was automatically generated from the Wikipedia article "Desire" licensed under the GNU Free Documentation License .

Created by mwcl_wikipedia_en Dec 14, 2007

Last edited by mwcl_wikipedia_en Dec 14, 2007

View topic history »

Gallery add

There are no images for this topic yet.

Weblinks edit

Wikipedia

Recent Discussions about Desire

There are no conversations on this topic. Would you like to start one?

Start the Discussion »

Bases that include Desire add

©2009 Metaweb Technologies, Inc.

Freebase contains information on: Arts & Entertainment | Products & Services | Science & Technology | Society | Special Interests | Sports | Time & Space

Now we have all those data expressed in a language that allows semantic interoperability ...

What we can do with OWL

- ... (maybe) we can check the consistency, classify, and query all this knowledge
- this is great, but ...
- ... when I locally reuse parts of such a big bunch of knowledge, inferences sometimes produce strange results:
  - a web page same as an email address (e.g. http://.../Aldo owl:sameAs mailto://aldo@...)
  - a person same as a wikipedia article (e.g. Aldo owl:sameAs http://en.wikipedia.org/Aldo)
  - Italy is a continent (e.g. (Italy rdf:type (Country) rdfs:subClassOf Continent))
  - ...
- ... and problems are hardly fixable on a large scale
- Logical consistency is not the main problem
  - e.g. owl:sameAs can be wrongly used and still we have consistency
- Why OWL is not enough?

When to use
owl:Individual, Class, ObjectProperty, DatatypeProperty?

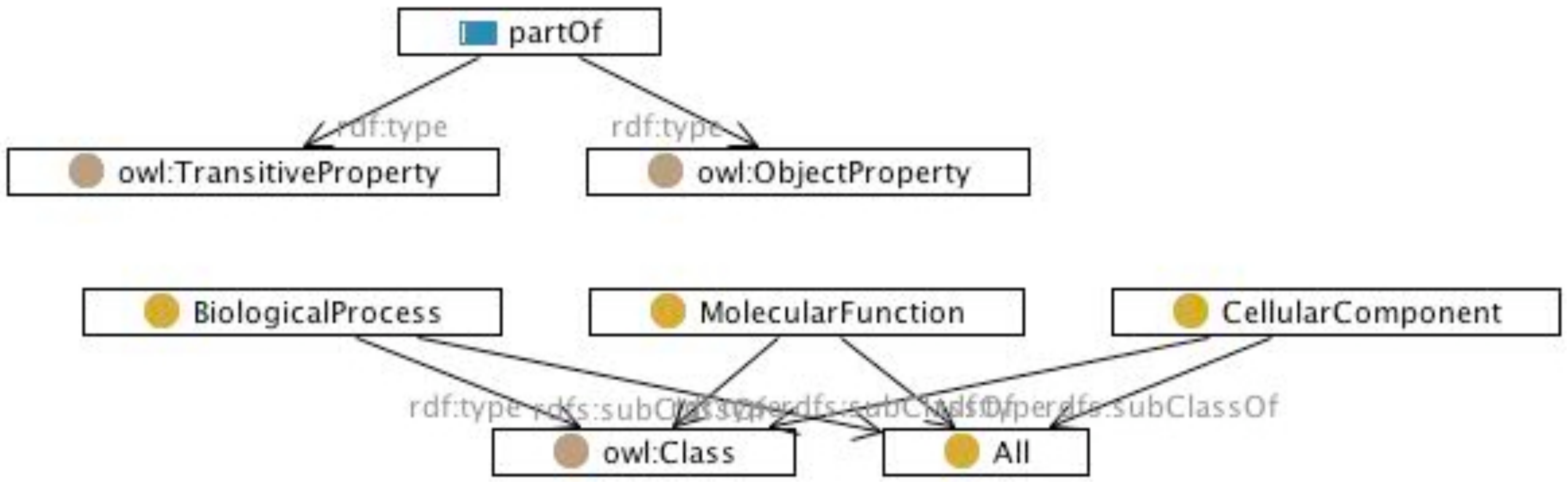
- OWL gives us logical language constructs, but does not give us any guidelines on how to use them in order to solve our tasks.
- E.g. modeling something as an individual, a class, or an object property can be quite arbitrary

New problems arising on the Web...

- cf. Semantic Web Interest Group post May 27th, 2008 by Zille Huma:
  "I have been wondering for sometime now that why isn't it a popular trend to store standard activities of a domain in the ontology and not only the concepts, e.g., for the tourism domain, ontologies normally contain concepts like Tourist, Resort, etc. but I have not so far come across an ontology that also contains the standard activities like searchResort, bookHotel, etc. Why is it so? What support is provided in the ontology langauges to model the standard activities of the domain as well?"
- (1) a functionality for searching resorts is implemented in our web service
  - owl:Individual(searchResort) rdf:type(Functionality)
- (2) searching resorts is a type of functionality required for this kind of services
  - owl:Class(searchResort) rdfs:subClassOf(Functionality)
- (3) who has been searching for what resorts in our web service?
  - owl:ObjectProperty(searchResort) rdfs:range(Resort)
- (4) how many users have been using our resort searching functionality?
  - owl:DatatypeProperty(searchResort) rdfs:range(xsd:boolean)

Solutions?

- ... OWL is not enough for building a good ontology, and we cannot ask all web users either to learn logic, or to study ontology design
- Reusable solutions are described as Ontology Design Patterns, which help reducing arbitrariness without asking for sophisticated skills ...
- ... provided that tools are built for any user :)



An ontology designer's world

- Requirements (e.g."*I want to attend my ideal talk"*)
- Logical constructs (rdfs:subClassOf, owl:Restriction, ...)
- Existing ontologies (FOAF, BibTex, SWC, DOLCE, ...)
- Informal knowledge resources (CiteSeer,ACM topic catalog)
- Conventions and practices (e.g. naming, URI making, XML2OWL, SKOS, disjoint covering, reification methods, transitive partOf, role-task, ...)
- Tools: editors, reasoners, translators, etc. (Protégé, NeOn Toolkit, TBC, FaCT++, Pellet, SMW, Jena, AllegroGraph, Virtuoso, ...)

A well-designed ontology ...

- Obeys to “capital questions”:
  - What are we talking about?
  - Why do we want to talk about it?
  - Where to find reusable knowledge?
  - Do we have the resources to maintain it?
- Whats, whys and wheres constitute the *Problem Space* of an ontology project
- Ontology designers need to find solutions from a *Solution Space*
- Matching problems to solutions is not trivial

Outline

- Designing Computational Ontologies
- Ontology Design Patterns
- ontologydesignpatterns.org initiative

Ontology Design Pattern

- *An ontology design pattern is a successful reusable solution to a recurrent modeling problem*

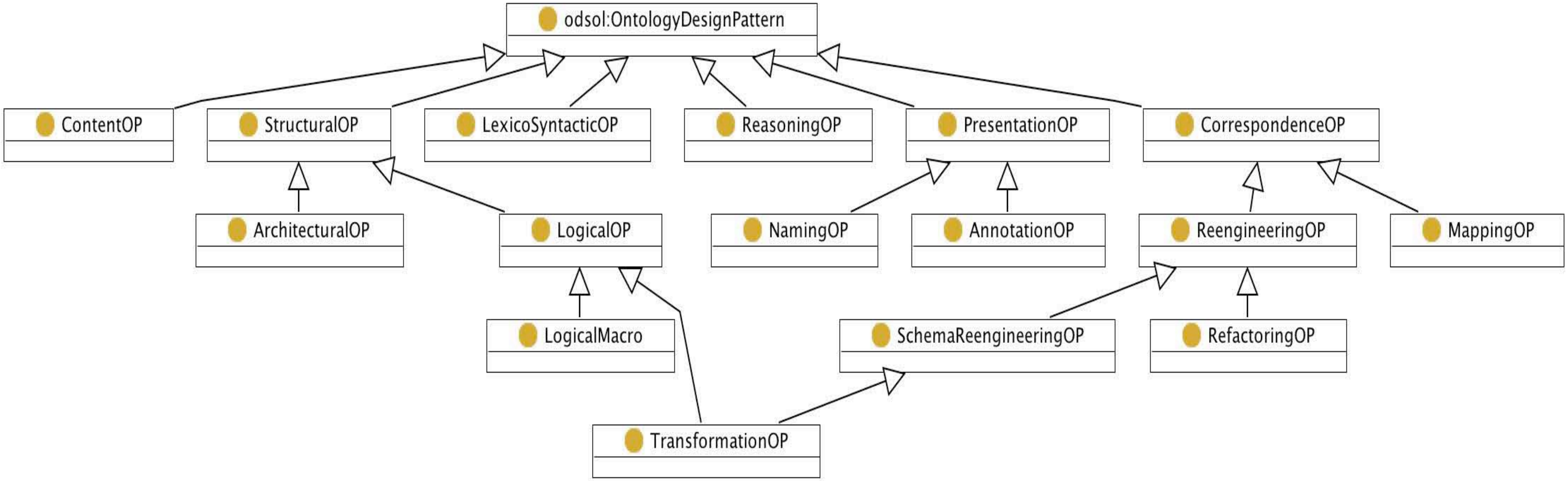
Pattern-based design

- Pattern-based ontology design is the activity of searching, selecting, and composing different patterns
  - Logical, Reasoning, Architectural, Naming, Reengineering, Content
  - Common framework to understand modelling choices (the "solution space") wrt task- and domain-oriented requirements (the "problem space")
  - http://www.ontologydesignpatterns.org

Pattern-based design aka e*Xtreme Design* (XD)

- Pattern-based ontology design is the activity of searching, selecting, and composing different patterns
  - Logical, Reasoning, Architectural, Naming, Correspondence, Reengineering, Content
  - Common framework to understand modeling choices (the “solution space”) wrt task- and domain-oriented requirements (the “problem space”)
  - http://www.ontologydesignpatterns.org

Types of Ontology Design Patterns (OPs)



- We also distinguish between ontological resources that are not OPs and Ontology Design Anti-Patterns (AntiOP)

Examples of Presentation OPs

- Class names should not contain plurals, unless explicitly required by the context
  - Names like Areas is considered bad practice, if e.g. an instance of the class Areas is a single area, not a collection of areas
- It is useful to include the name of the parent class as a suffix of the class name
  - e.g. MarineArea rdfs:subClassOf Area
- Class names conventionally start with a capital letter
  - e.g. Area instead of area

Examples of Reasoning OPs

- Precise
  - Classification
  - Subsumption
  - Inheritance
  - Materialization
  - De-anonymizing
  - ...
- Approximate
  - Approximate classification
  - Similarity induction
  - Taxonomy induction
  - Relevance detection
  - Latent semantic indexing
  - Automatic alignment
  - ...

or some workflow of them, cf.TBC

Example of Schema Reengineering OP: kos2skosABox

$$\texttt{KOS} \mapsto \texttt{skos:ConceptSchema} \tag{2.1}$$

$$\texttt{Descriptor} \mapsto \texttt{skos:Concept} \tag{2.2}$$

$$\texttt{Broader Term} \mapsto \texttt{skos:broader} \tag{2.3}$$

$$\texttt{Related Term} \mapsto \texttt{skos:related} \tag{2.4}$$

Example of Mapping OPs

- Also called “correspondence patterns” in [16]
  - equivalent to, (not equivalent to)
    - foaf:Agent ≡ wn16:Agent-3
  - contained in, (not contained in)
    - foaf:Person ⊑ geo:SpatialThing
  - overlap with
    - foaf:Person ⊓ dul:Person
  - disjoint with
    - (dul:PhysicalPerson ⊓ dul:SocialPerson) = ∅
  - logically heterogeneous mapping
    - dul:PhysicalPerson (owl:Class) ≈ p1:PhysicalPersonRole (owl:Individual)
- We also consider an additional semantic relation, *cloned from*
  - ontology element oe_1 in one ontology is the clone of an ontology element oe_2 in another ontology

Example of Logical Macro

- Logical macros provide a shortcut to model a recurrent intuitive logical expression

Example:

the macro: $\nabla R.C$ [7]

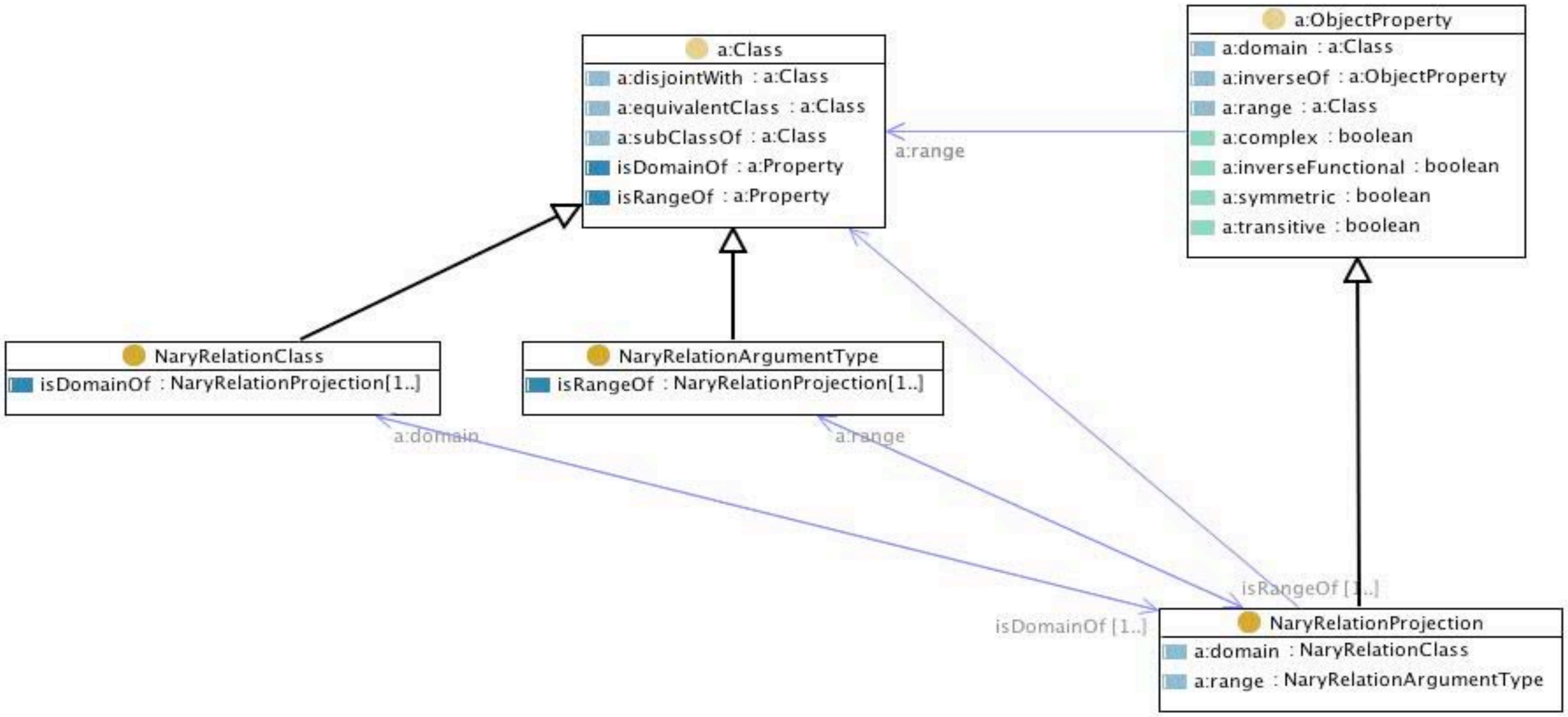
colloquially means "every R must be a C"

formally: $\exists R.\top \sqcap \forall R.C$

in OWL:

the combination of an owl:allValuesFrom restriction with an owl:someValuesFrom restriction.

Example of Transformation pattern: N-ary relation (1/2)



a:Class
a:disjointWith : a:Class
a:equivalentClass : a:Class
a:subClassOf : a:Class
isDomainOf : a:Property
isRangeOf : a:Property

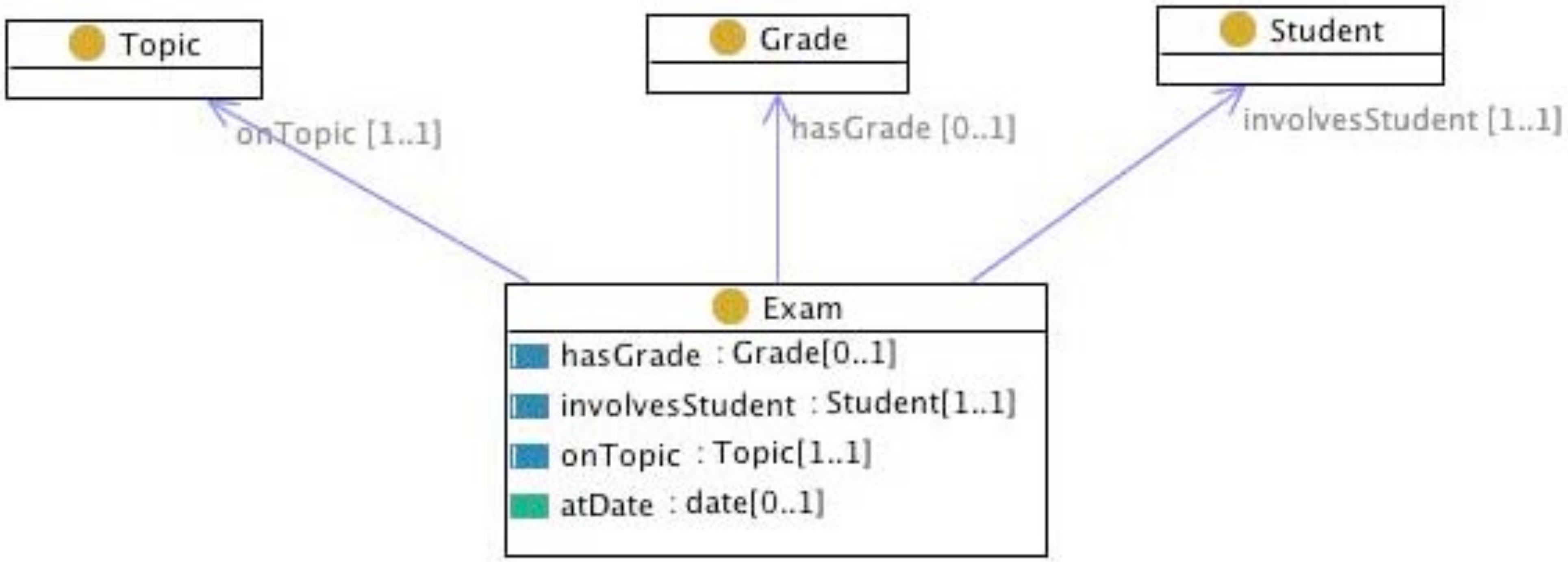
a:ObjectProperty
a:domain : a:Class
a:inverseOf : a:ObjectProperty
a:range : a:Class
a:complex : boolean
a:inverseFunctional : boolean
a:symmetric : boolean
a:transitive : boolean

NaryRelationClass
isDomainOf : NaryRelationProjection[1..]

NaryRelationArgumentType
isRangeOf : NaryRelationProjection[1..]

NaryRelationProjection
a:domain : NaryRelationClass
a:range : NaryRelationArgumentType

Example of Transformation pattern: N-ary relation (2/2)



But beware of identification constraints! [15]

Content Ontology Design Patterns

Some theory

Reusable ontologies?

- How many cases of reusability?
- What kind of ontologies are mostly reusable?
- How many ontologies have been actually specialized in more than one domain?
- How many studies in comparing the cost of reusing vs. developing from scratch?
- How many studies in evaluating/facilitating reusability?

- Let's face it: reusing, when applied, is an art, not a communicable/ manageable know-how
- Started with W3C SWBPD: OWL modelling best practices, semantic SE patterns, techniques to vocabulary porting and migration to the SW
- Then in EU NeOn project: ontology design for networked and contextualized ontologies: Watson, ODP Portal, Modularization plugin, Pattern-based design plugin

From the lessons learnt ...

- We envision small ontologies with *explicit* documentation of *design rationales*, and best *reengineering practices*
  - components supported by specific functionalities
    - selection, matching, composition, etc.
  - implemented in repositories, registries, catalogues, open discussion and evaluation forums, and in new-generation ontology design tools
    - ontologydesignpattern.org
    - ODP and Watson APIs
    - NeOn ODP Plugin
    - etc.

Content OPs (CPs) 1/2

- CPs encode conceptual, rather than logical design patterns.
  - Logical OPs solve design problems independently of a particular conceptualization
  - CPs are patterns for solving design problems for the domain classes and properties that populate an ontology, therefore they address content problems

- CPs are instantiations of Logical OPs (or of compositions of Logical OPs), featuring a non-empty signature
  - Hence, they have an explicit non-logical vocabulary for a specific domain of interest, i.e. they are content-dependent

Content OPs (CPs) 2/2

- CPs are instantiations of Logical OPs (or of compositions of Logical OPs), featuring a non-empty signature
  - Hence, they have an explicit non-logical vocabulary for a specific domain of interest, i.e. they are content-dependent
- Modeling problems solved by CPs have two components: domain and requirements.
  - A same domain can have many requirements (e.g. different scenarios in a clinical information context)
  - A same requirement can be found in different domains (e.g. different domains with a same "expert finding" scenario)
  - A typical way of capturing requirements is by means of competency questions [11]

Peter Clark’s idea

- A pattern is a theory template. It denotes a structure that is invariant under signature transformation (morphism). Pattern validity in an application is then left to a subjective decision.
  - E.g. the axiom:
- [If a consumer is connected to a producer, then it is supplied]
- $\forall c((\text{consumer}(c) \land \exists p(\text{producer}(p) \land \text{connects}(c,p))) \rightarrow \text{supplied}(c))$
  - via signature morphism becomes e.g. in an application:
- [If a light is connected to a battery, then it is powered]
- $\forall c((\text{light}(c) \land \exists p(\text{battery}(p) \land \text{connects}(c,p))) \rightarrow \text{powered}(c))$
- But if a pattern is just an untyped structure, there are no ways to distinguish a Logical OP vs. a CP

CPs vs. Logical OPs

- $\forall c((\mathrm{consumer}(c) \wedge \exists p(\mathrm{producer}(p) \wedge \mathrm{connects}(c,p))) \rightarrow \mathrm{supplied}(c))$

- SubClassOf
- ((intersectionOf
- Consumer
- (restriction(connects someValuesFrom(Producer))))
- Supplied)

CP: specific (non-logical) vocabulary

- $\forall c((\varphi(c) \wedge \exists p(\psi(p) \wedge \rho(c,p))) \rightarrow \chi(c))$

- SubClassOf
- ((intersectionOf
- owl:Class:φ
- (restriction(owl:ObjectProperty:ρ someValuesFrom(owl:Class:ψ))))
- owl:Class:χ)

Logical OP: no specific vocabulary

- In OWL, this is a GCI (General Concept Inclusion) axiom. Not a typical LP

Formal characteristics of OWL CPs

- (Small) ontology morphing
  - “being a part of something at some time”
- Downward subsumption of at least one element
  - “being a component of a system at some time”
- Only rarely GCI axioms like in Clark’s example
- Mostly graphs of classes and properties that are self-connected through axioms (subClassOf, equivalentClass, domain, range, disjointFrom)
  - ObjectProperty(component domain(System))
- Usually there is an underlying n-ary relation (sometimes polymorphic)
  - $\text{component}(s,e,t) \rightarrow \text{System}(s) \wedge \text{Entity}(e) \wedge \text{Time}(t)$
  - ? $\text{component}(s,e,t,...) \rightarrow \text{System}(s) \wedge \text{Entity}(e) \wedge \text{Time}(t) \wedge ...(...)$

Formal characteristics of OWL CPs

- Mostly graphs of classes and properties, self-connected through axioms (subClassOf, equivalentClass, domain, range, disjointFrom)
  - ObjectProperty(component domain(System))
- Usually applied through downward subsumption of at least one element
  - "being a part of something at some time"
    - "being a component of a system at some time"
      - "being a section in a law at some time"
- Or through composition
      - "being a section in a law at some time" ⊗ "being expressed in a legal text"
- Usually there is an underlying n-ary relation (sometimes polymorphic)
  - component(s,e,t) → System(s) $\wedge$ Entity(e) $\wedge$ Time(t)
  - ? component(s,e,t,...) → System(s) $\wedge$ Entity(e) $\wedge$ Time(t) $\wedge$ Function(...) ...

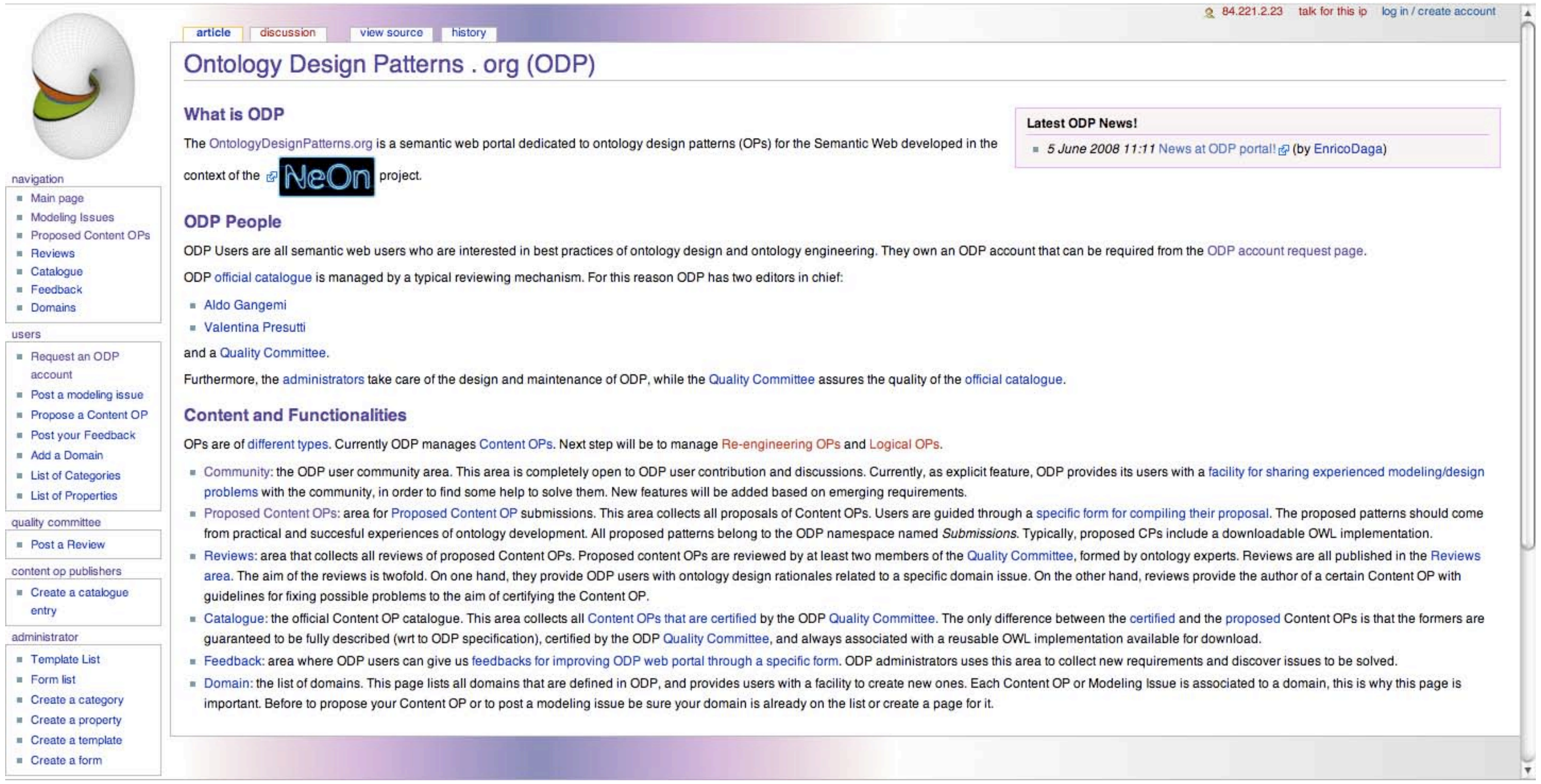
Pragmatic characteristics of CPs

- Domain-dependent
  - Expressed with a domain-specific (non-logical) vocabulary
- Requirement-covering
  - Solve domain modelling problems (expressible as use-cases, tasks or "competency questions"), at a typical maximum size (cf. blink)
- Reasoning-relevant components
  - Allow some form of inference (minimal axiomatization, e.g. not an isolated class)
- Cognitively-relevant components
  - Catch relevant core notions of a domain and the related expertise -- blink knowledge
- Linguistically-relevant components
  - Are lexically grounded, e.g. they match linguistic frames, or at least a domain terminology
- Examples:
  - PartOf, Participation, Plan, Legal Norm, Legal Fact, Sales Order, Research Topic, Legal Contract, Inflammation, Medical Guideline, Gene Ontology Top, Situation, TimeInterval, etc.

Generic ontology requirements (GCQ)

Generic Competency Questions	Specific Modelling Use Case
Who does what, when and where?	Production reports, schedules
Which objects take part in a certain event?	Resource allocation, biochemical pathways
What are the parts of something?	Component schemas, warehouse management
What’s an object made of?	Drug and food composition, e.g. for safety (comp.)
What’s the place of something?	Geographic systems, resource allocation
What’s the time frame of something?	Dynamic knowledge bases
What technique, method, practice is being used?	Instructions, enterprise know-how database
Which tasks should be executed in order to achieve a certain goal?	Planning, workflow management
Does this behaviour conform to a certain rule?	Control systems, legal reasoning services
What’s the function of that artifact?	System description
How is that object built?	Control systems, quality check
What’s the design of that artifact?	Project assistants, catalogues
How did that phenomenon happen?	Diagnostic systems, physical models
What’s your role in that transaction?	Activity diagrams, planning, organizational models
What that information is about? How is it realized?	Information and content modelling, computational models, subject directories
What argumentation model are you adopting for negotiating an agreement?	Cooperation systems
What’s the degree of confidence that you give to this axiom?	Ontology engineering tools

Presentation



Ontology Design Patterns . org (ODP)

What is ODP

The OntologyDesignPatterns.org is a semantic web portal dedicated to ontology design patterns (OPs) for the Semantic Web developed in the context of the NeOn project.

Latest ODP News!

- *5 June 2008 11:11* News at ODP portal! (by EnricoDaga)

ODP People

ODP Users are all semantic web users who are interested in best practices of ontology design and ontology engineering. They own an ODP account that can be required from the ODP account request page.

ODP official catalogue is managed by a typical reviewing mechanism. For this reason ODP has two editors in chief:

- Aldo Gangemi
- Valentina Presutti

and a Quality Committee.

Furthermore, the administrators take care of the design and maintenance of ODP, while the Quality Committee assures the quality of the official catalogue.

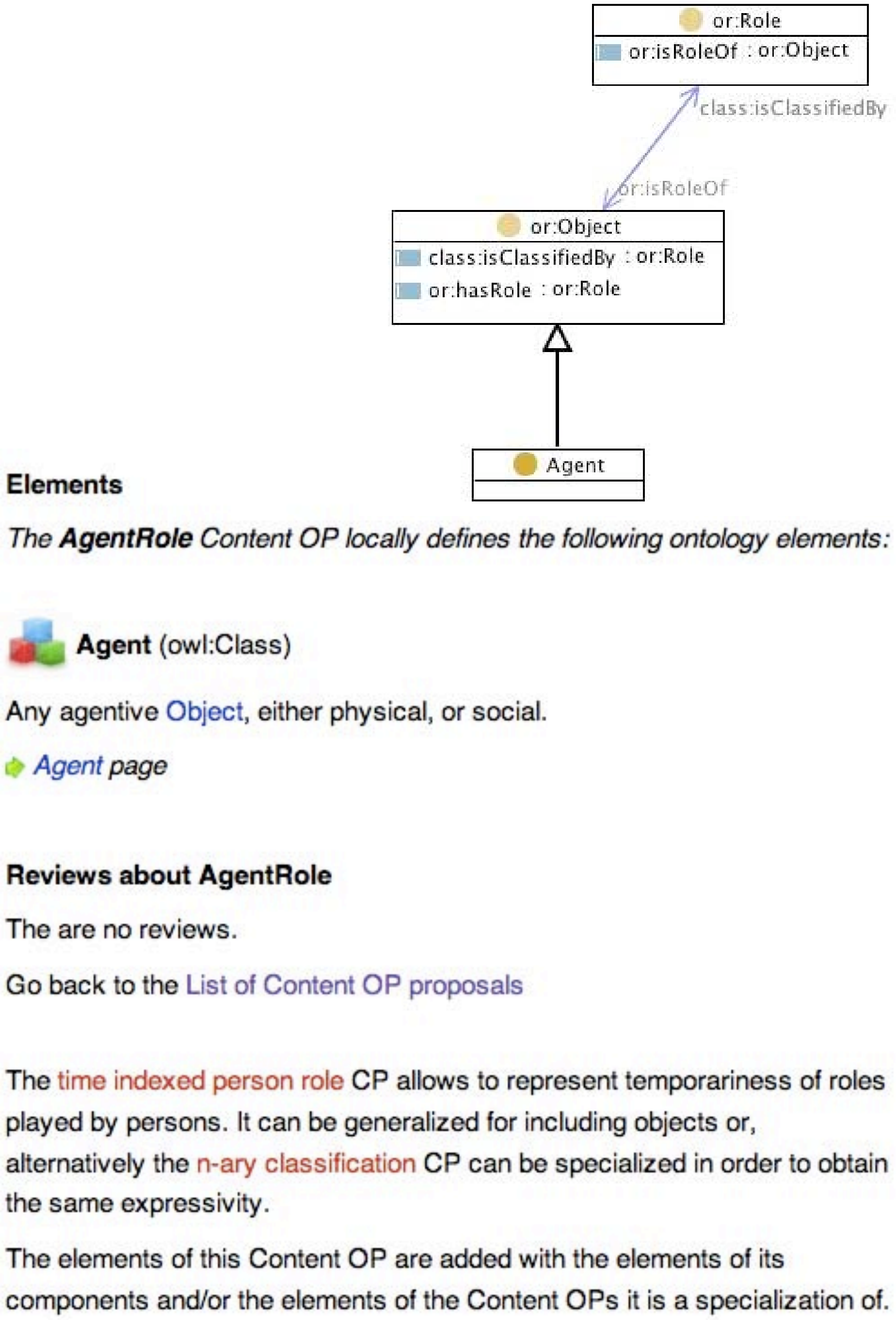
Content and Functionalities

OPs are of different types. Currently ODP manages Content OPs. Next step will be to manage Re-engineering OPs and Logical OPs.

- Community: the ODP user community area. This area is completely open to ODP user contribution and discussions. Currently, as explicit feature, ODP provides its users with a facility for sharing experienced modeling/design problems with the community, in order to find some help to solve them. New features will be added based on emerging requirements.
- Proposed Content OPs: area for Proposed Content OP submissions. This area collects all proposals of Content OPs. Users are guided through a specific form for compiling their proposal. The proposed patterns should come from practical and succesful experiences of ontology development. All proposed patterns belong to the ODP namespace named *Submissions*. Typically, proposed CPs include a downloadable OWL implementation.
- Reviews: area that collects all reviews of proposed Content OPs. Proposed content OPs are reviewed by at least two members of the Quality Committee, formed by ontology experts. Reviews are all published in the Reviews area. The aim of the reviews is twofold. On one hand, they provide ODP users with ontology design rationales related to a specific domain issue. On the other hand, reviews provide the author of a certain Content OP with guidelines for fixing possible problems to the aim of certifying the Content OP.
- Catalogue: the official Content OP catalogue. This area collects all Content OPs that are certified by the ODP Quality Committee. The only difference between the certified and the proposed Content OPs is that the formers are guaranteed to be fully described (wrt to ODP specification), certified by the ODP Quality Committee, and always associated with a reusable OWL implementation available for download.
- Feedback: area where ODP users can give us feedbacks for improving ODP web portal through a specific form. ODP administrators uses this area to collect new requirements and discover issues to be solved.
- Domain: the list of domains. This page lists all domains that are defined in ODP, and provides users with a facility to create new ones. Each Content OP or Modeling Issue is associated to a domain, this is why this page is important. Before to propose your Content OP or to post a modeling issue be sure your domain is already on the list or create a page for it.

- A catalogue of CPs
  - http://www.ontologydesignpatterns.org (odp-web)
  - catalogue entry
- Annotation properties:
  - http://www.ontologydesignpatterns.org/schemas/cpannotationschema.owl
  - annotation of OWL implementation of CPs

Example 1: AgentRole



Elements

*The **AgentRole** Content OP locally defines the following ontology elements:*

Agent (owl:Class)

Any agentive Object, either physical, or social.

Agent page

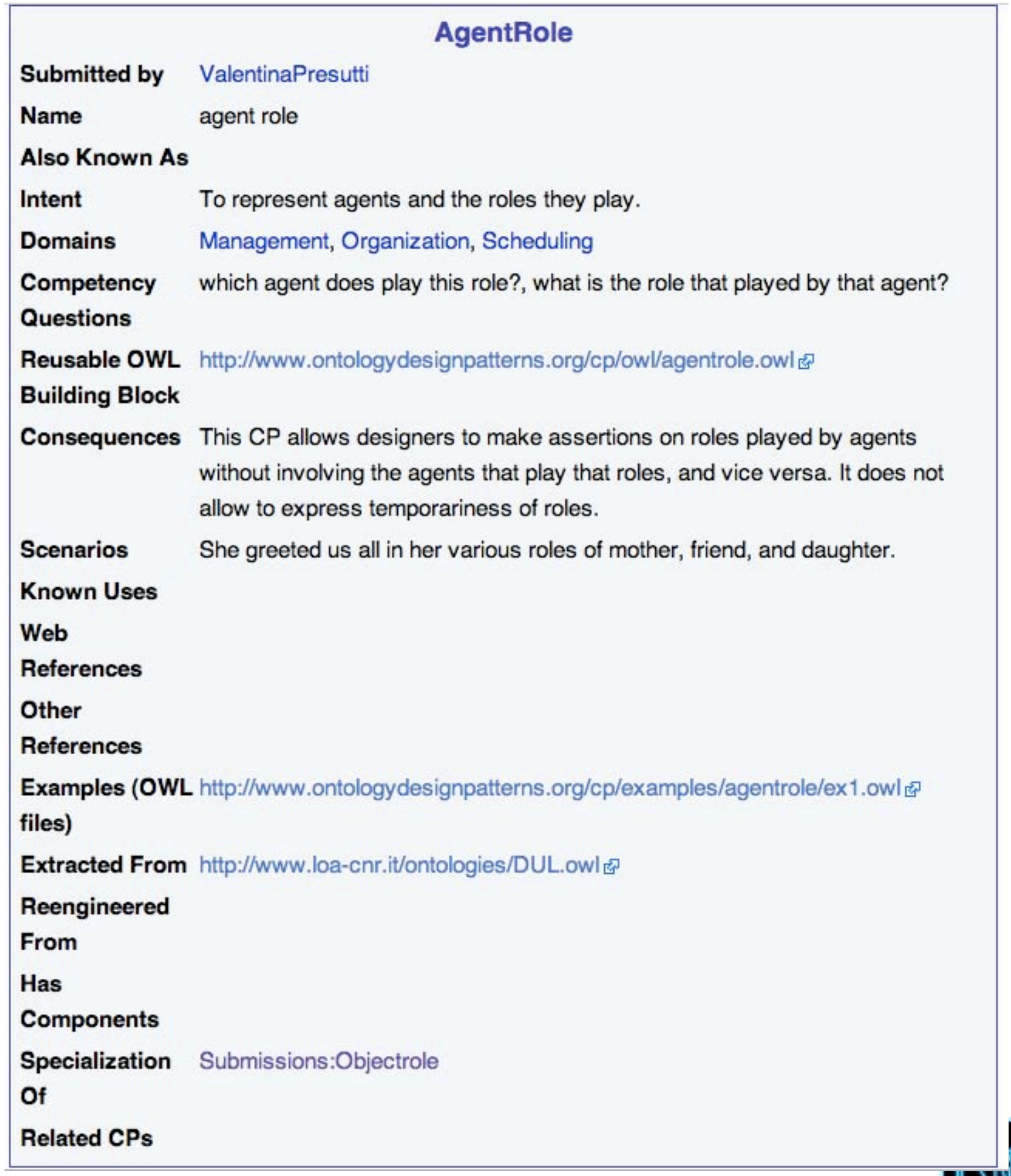
Reviews about AgentRole

The are no reviews.

Go back to the List of Content OP proposals

The time indexed person role CP allows to represent temporariness of roles played by persons. It can be generalized for including objects or, alternatively the n-ary classification CP can be specialized in order to obtain the same expressivity.

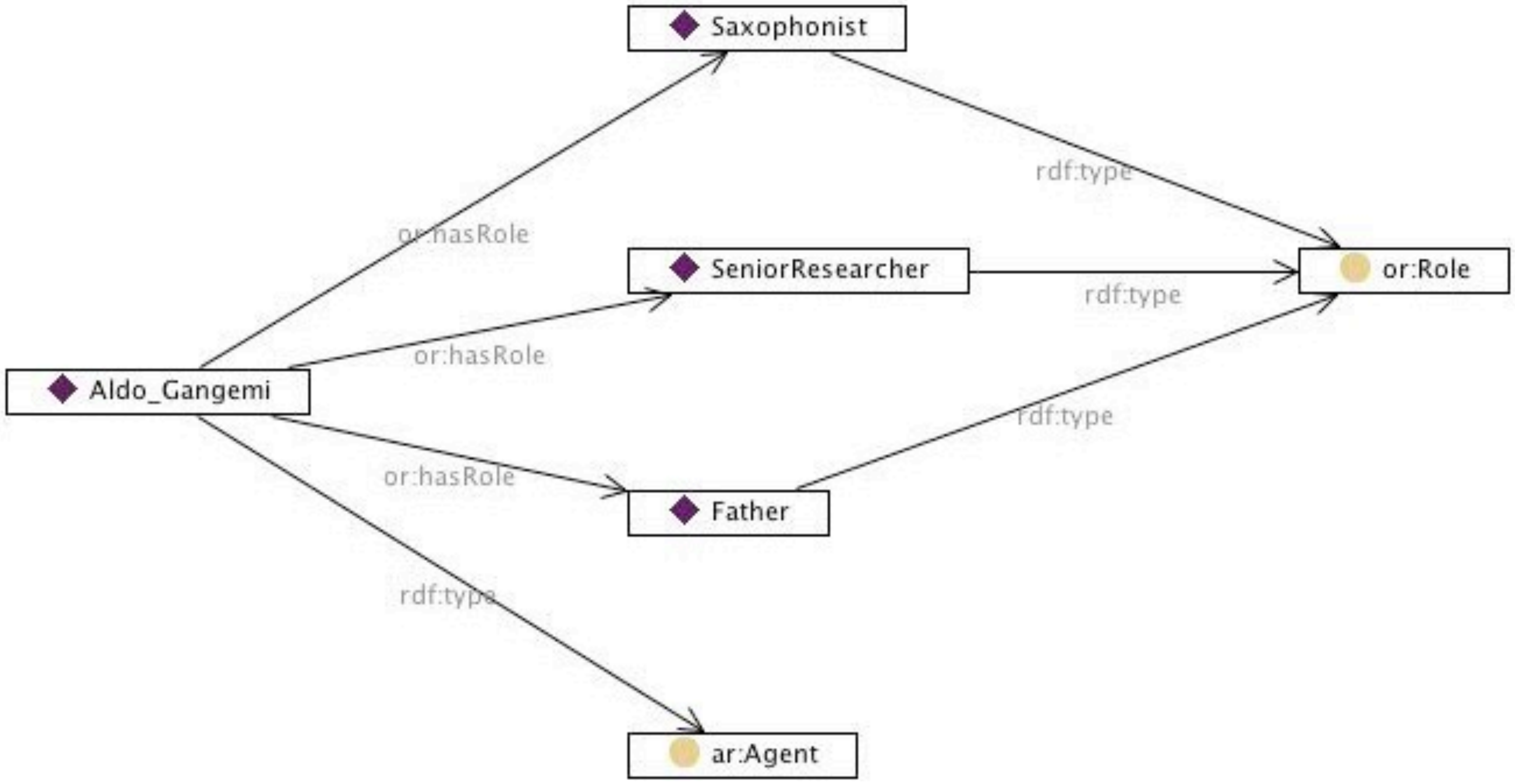
The elements of this Content OP are added with the elements of its components and/or the elements of the Content OPs it is a specialization of.



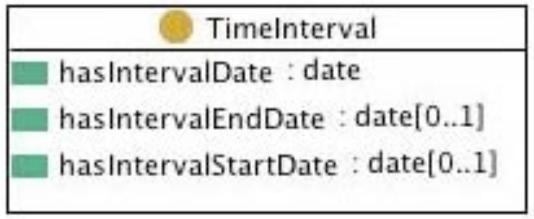
AgentRole

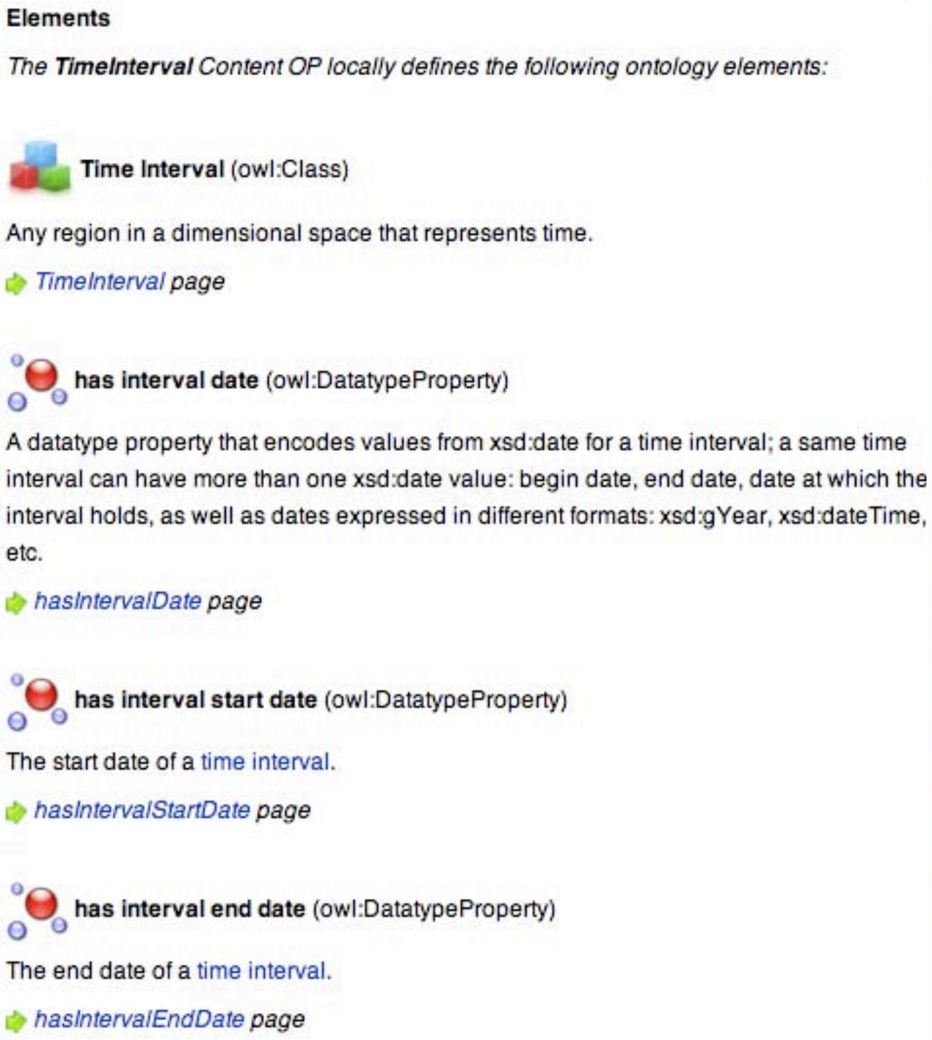
Submitted by	ValentinaPresutti
Name	agent role
Also Known As	
Intent	To represent agents and the roles they play.
Domains	Management, Organization, Scheduling
Competency Questions	which agent does play this role?, what is the role that played by that agent?
Reusable OWL Building Block	http://www.ontologydesignpatterns.org/cp/owl/agentrole.owl
Consequences	This CP allows designers to make assertions on roles played by agents without involving the agents that play that roles, and vice versa. It does not allow to express temporariness of roles.
Scenarios	She greeted us all in her various roles of mother, friend, and daughter.
Known Uses	
Web References	
Other References	
Examples (OWL files)	http://www.ontologydesignpatterns.org/cp/examples/agentrole/ex1.owl
Extracted From	http://www.loa-cnr.it/ontologies/DUL.owl
Reengineered From	
Has Components	
Specialization Of	Submissions:Objectrole
Related CPs	

Agent Role Instantiation



Example 2: Time Interval


TimeInterval
hasIntervalDate : date
hasIntervalEndDate : date[0..1]
hasIntervalStartDate : date[0..1]



Elements

*The **TimeInterval** Content OP locally defines the following ontology elements:*

Time Interval (owl:Class)

Any region in a dimensional space that represents time.

TimeInterval page

has interval date (owl:DatatypeProperty)

A datatype property that encodes values from xsd:date for a time interval; a same time interval can have more than one xsd:date value: begin date, end date, date at which the interval holds, as well as dates expressed in different formats: xsd:gYear, xsd:dateTime, etc.

hasIntervalDate page

has interval start date (owl:DatatypeProperty)

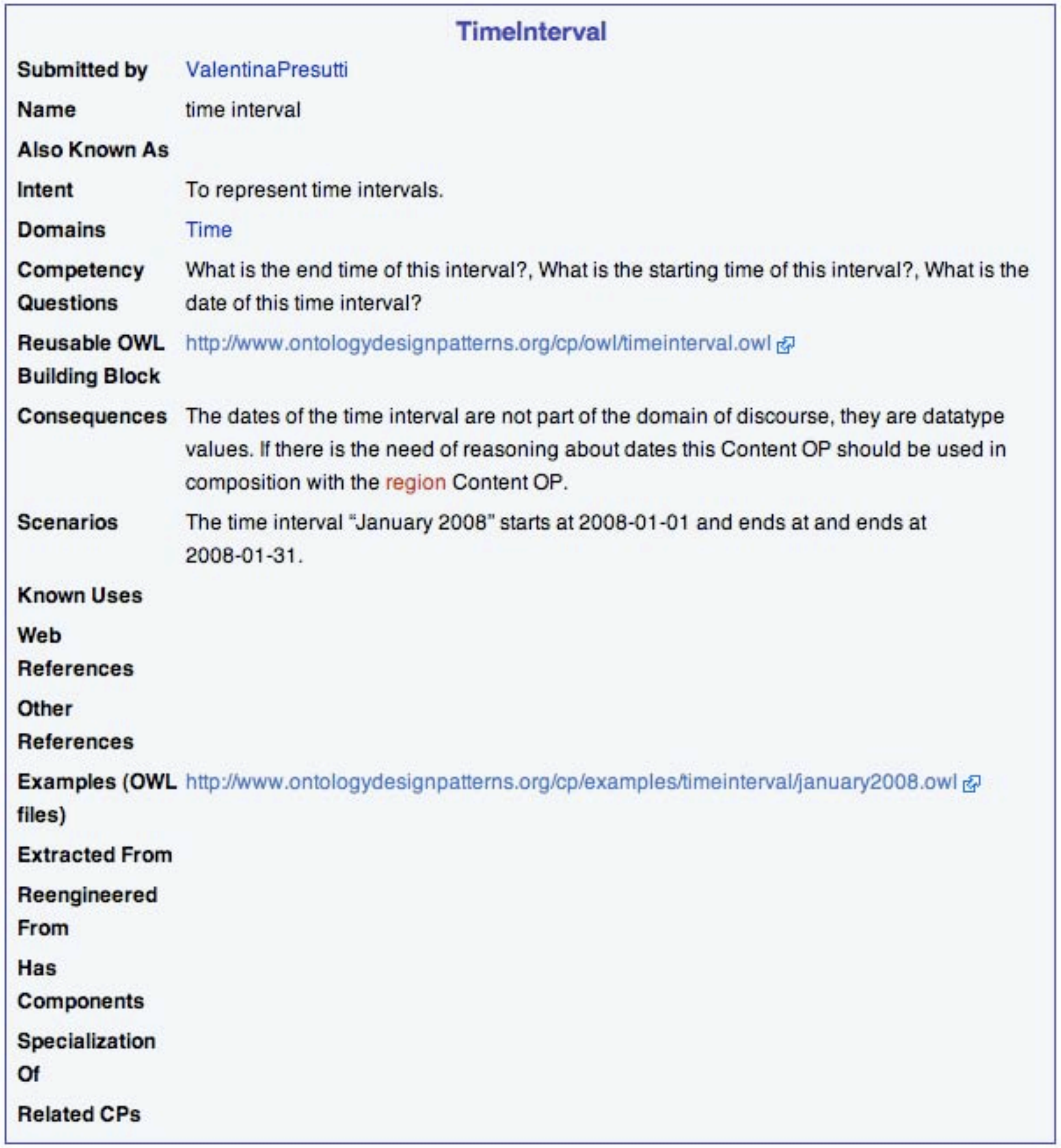
The start date of a time interval.

hasIntervalStartDate page

has interval end date (owl:DatatypeProperty)

The end date of a time interval.

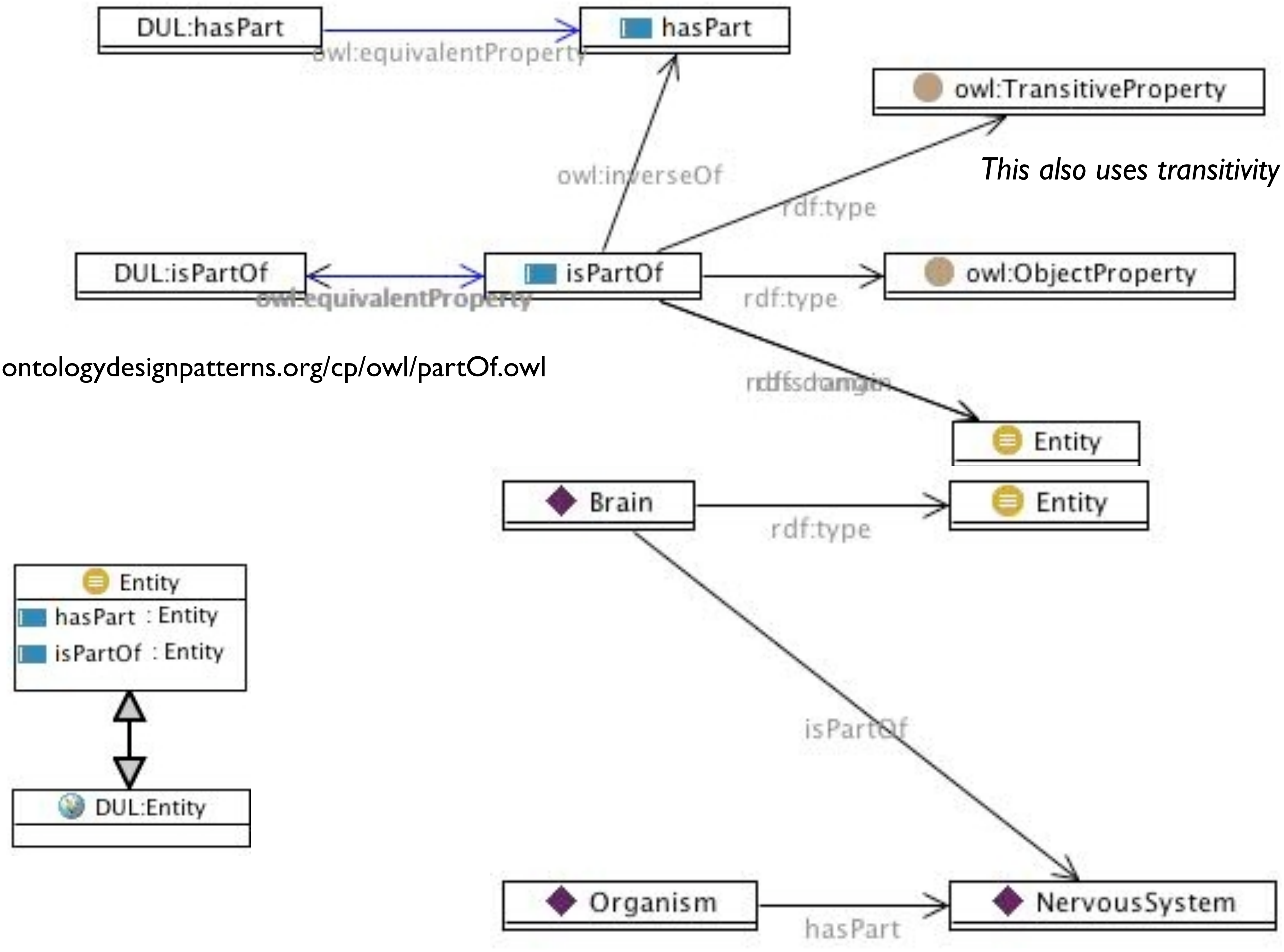
hasIntervalEndDate page



TimeInterval

Submitted by	ValentinaPresutti
Name	time interval
Also Known As	
Intent	To represent time intervals.
Domains	Time
Competency Questions	What is the end time of this interval?, What is the starting time of this interval?, What is the date of this time interval?
Reusable OWL Building Block	http://www.ontologydesignpatterns.org/cp/owl/timeinterval.owl
Consequences	The dates of the time interval are not part of the domain of discourse, they are datatype values. If there is the need of reasoning about dates this Content OP should be used in composition with the region Content OP.
Scenarios	The time interval "January 2008" starts at 2008-01-01 and ends at and ends at 2008-01-31.
Known Uses	
Web References	
Other References	
Examples (OWL files)	http://www.ontologydesignpatterns.org/cp/examples/timeinterval/january2008.owl
Extracted From	
Reengineered From	
Has Components	
Specialization Of	
Related CPs	

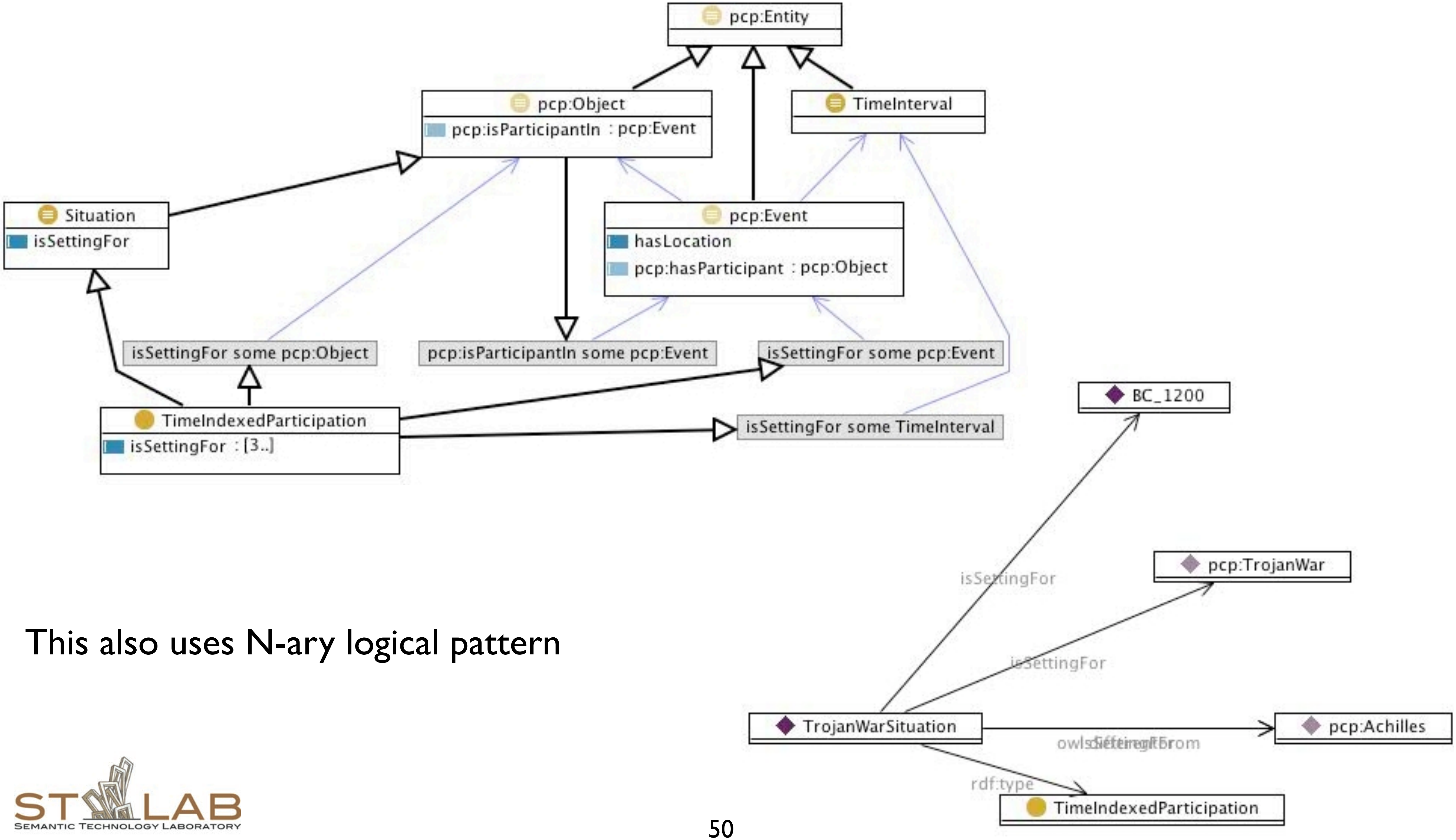
Example 3: Part



This also uses transitivity reasoning pattern

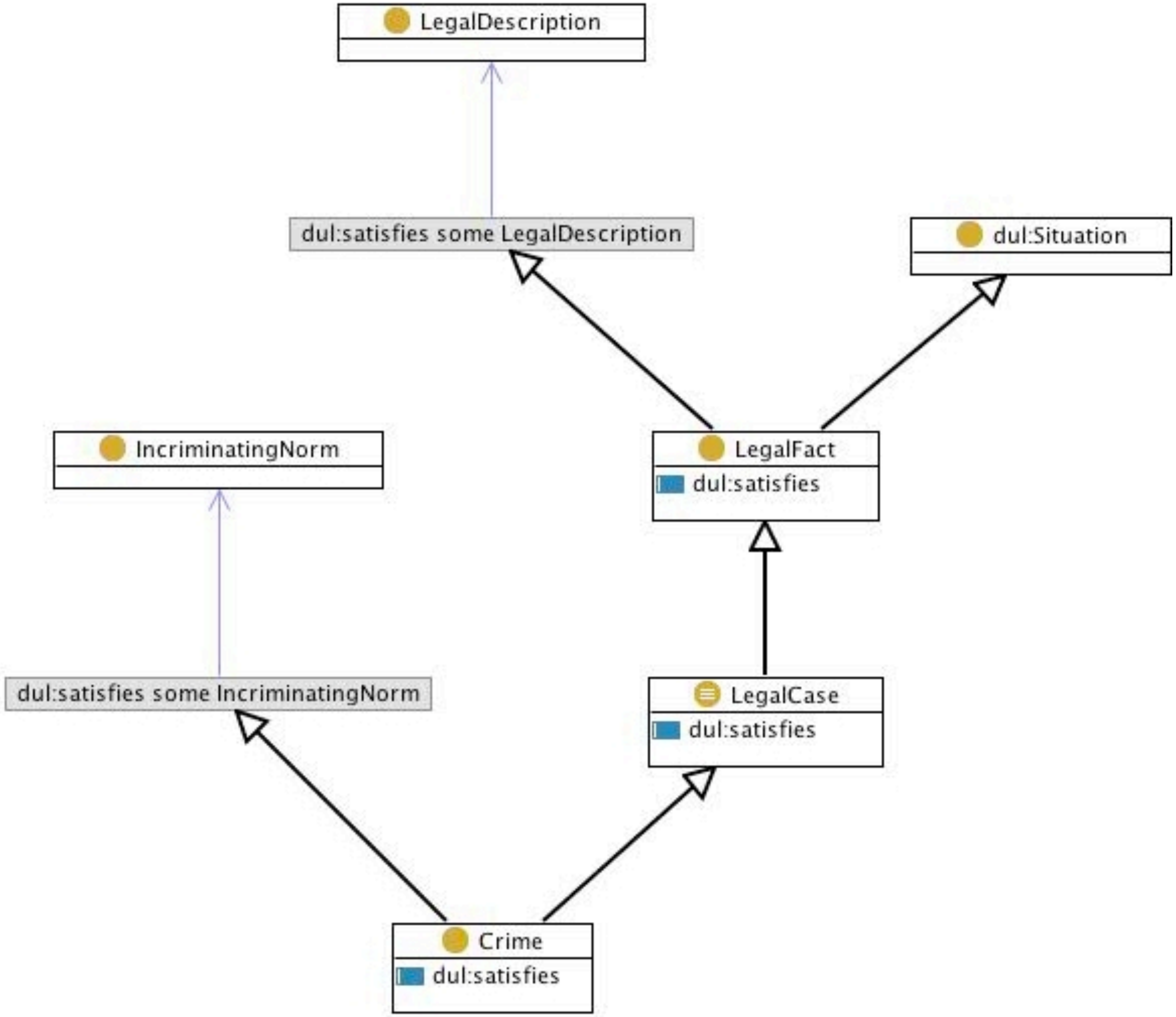
Cf. http://www.ontologydesignpatterns.org/cp/owl/partOf.owl

Example 4: Time-indexed Participation

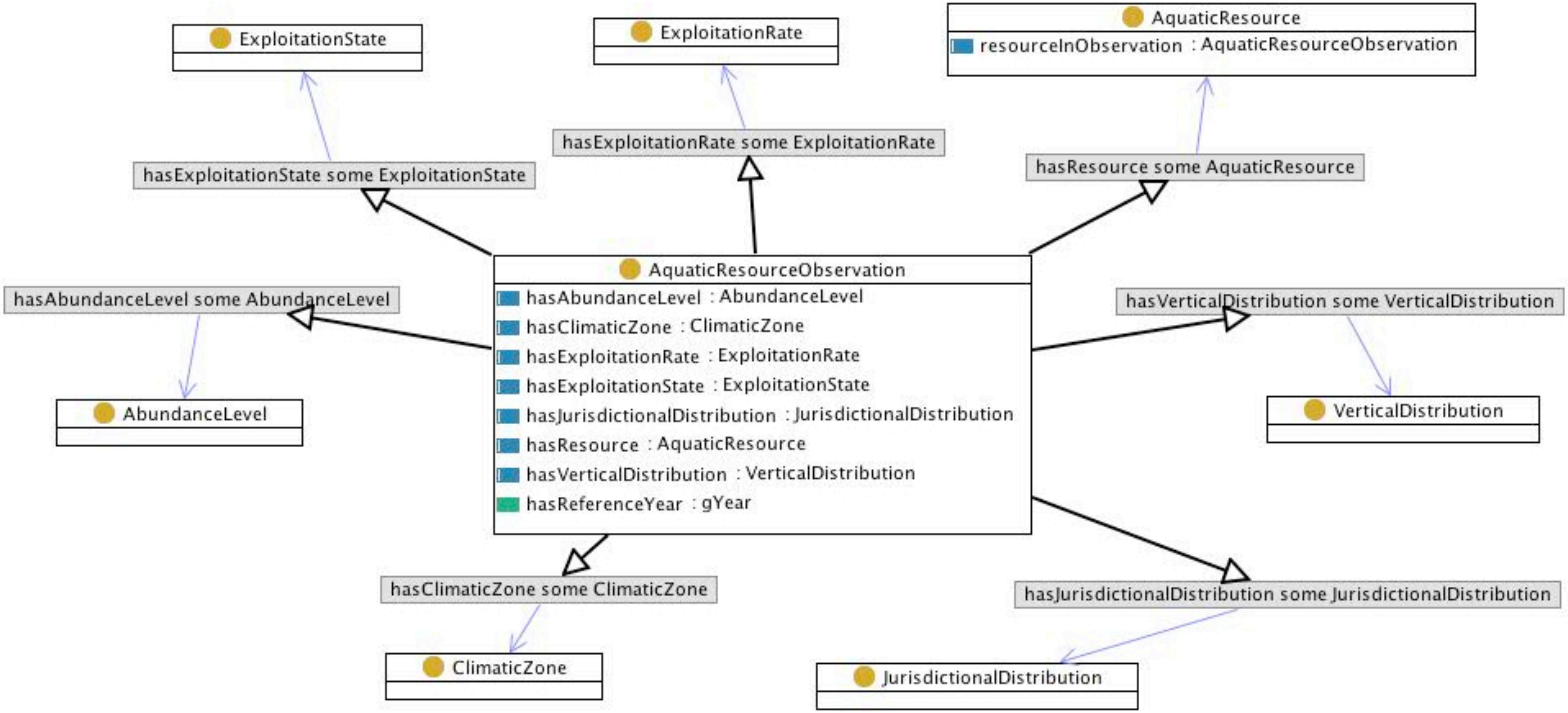


This also uses N-ary logical pattern

Example 5: Crime



Example 6: Aquatic Resource Observation


ExploitationState
ExploitationRate
AquaticResource
resourceInObservation : AquaticResourceObservation
hasExploitationRate some ExploitationRate
hasResource some AquaticResource
hasExploitationState some ExploitationState
AquaticResourceObservation
hasAbundanceLevel : AbundanceLevel
hasClimaticZone : ClimaticZone
hasExploitationRate : ExploitationRate
hasExploitationState : ExploitationState
hasJurisdictionalDistribution : JurisdictionalDistribution
hasResource : AquaticResource
hasVerticalDistribution : VerticalDistribution
hasReferenceYear : gYear
hasAbundanceLevel some AbundanceLevel
hasVerticalDistribution some VerticalDistribution
AbundanceLevel
VerticalDistribution
hasClimaticZone some ClimaticZone
hasJurisdictionalDistribution some JurisdictionalDistribution
ClimaticZone
JurisdictionalDistribution

(Re)use situations:
matching CPs covering against local problems

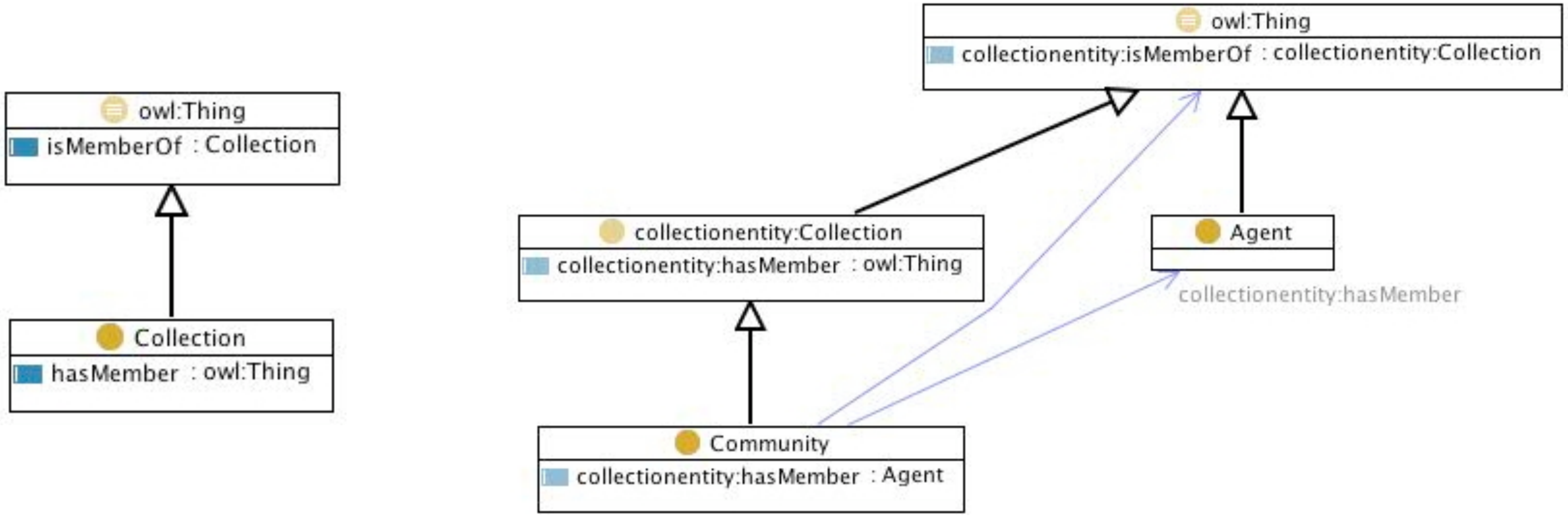
Representing local problems (LCQ)

- Local problems can be expressed in different ways:
  - use cases, scenarios, user requirements, local competency questions (cqs), etc.
- Following [11] all can be transformed to local “cqs”.
  - Red Hot Chili Peppers recorded the Stadium Arcadium album during 2005
  - *When did Red Hot Chili Peppers record the Stadium Arcadium album?*
  - *Which albums did Red Hot Chili Peppers record during 2005?*
  - ...
- Local “cqs” are not usually at the same level of generality as the cqs of CPs
  - e.g., they may contain reference to instance element e.g. Stadium Arcadium
  - we need to abstract them
  - *When did a certain band record a certain album?*
  - *Which albums did a certain band record during a certain time period?*
  - ...

What we mean by *matching cqs to CPs*

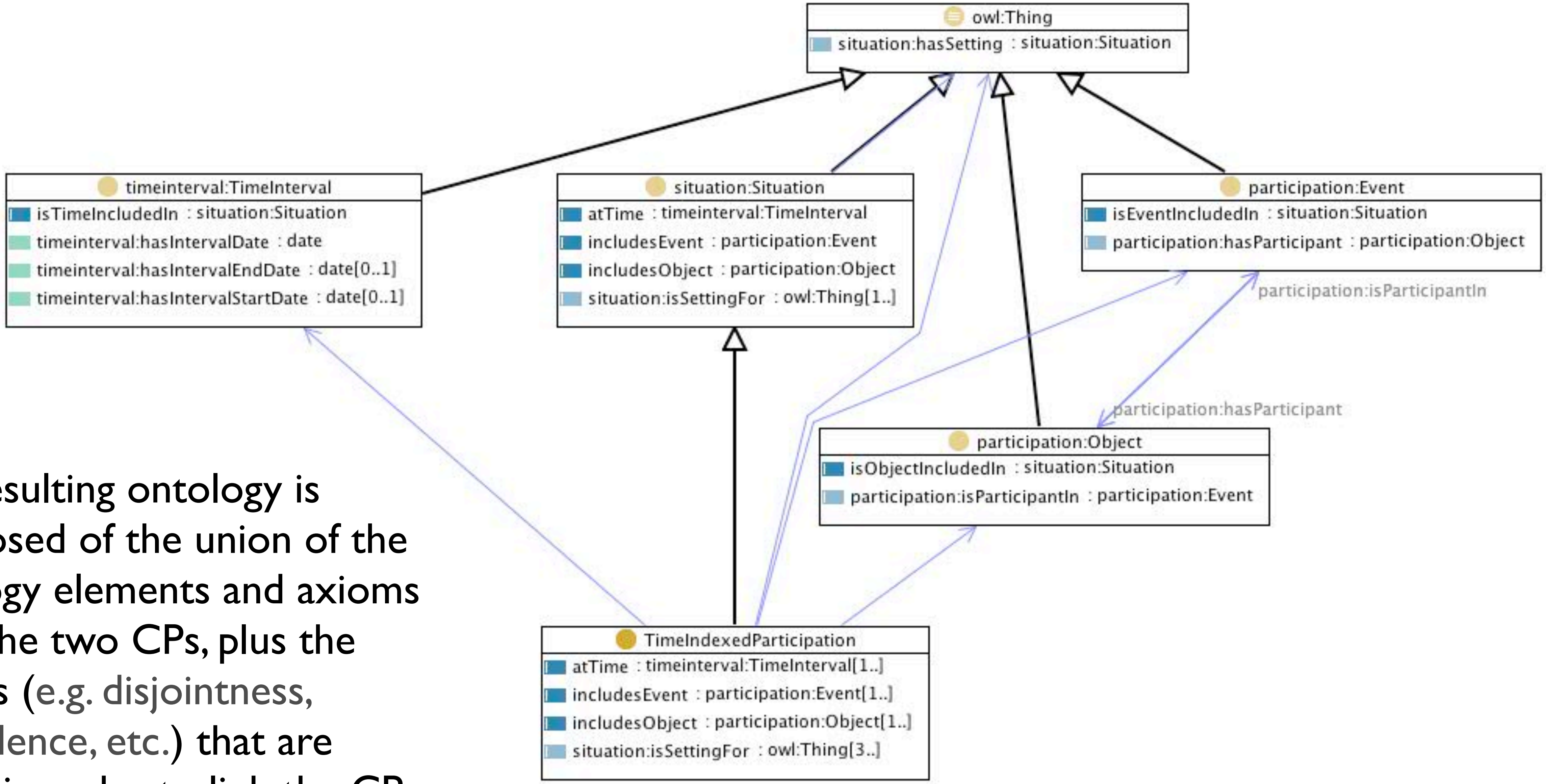
- What do we mean by matching a cq to CPs?
  - To compare the local cqs to the cqs covered by a CP in order to evaluate the CP suitability for solving the local problems
  - There is not yet automatic support for this task, hence it is performed as a human task
  - Ongoing work on automatic support for CP selection starting from local cqs
    - parsing of requirements and extraction of cqs
    - formalization of cqs
    - NLP support to match cqs terminology to CP lexicalizations
    - case-based reasoning [13]
    - ontology matching
    - ...

Sample Specialization


owl:Thing
isMemberOf : Collection
Collection
hasMember : owl:Thing
owl:Thing
collectionentity:isMemberOf : collectionentity:Collection
collectionentity:Collection
collectionentity:hasMember : owl:Thing
Agent
collectionentity:hasMember
Community
collectionentity:hasMember : Agent

- A content pattern CP_2 specializes CP_1 if at least one ontology element of CP_2 is subsumed by an ontology element of CP_1
  - i.e., either by *rdfs:subClassOf* or *rdfs:subPropertyOf*

Sample composition



owl:Thing
situation:hasSetting : situation:Situation

timeinterval:TimeInterval
isTimeIncludedIn : situation:Situation
timeinterval:hasIntervalDate : date
timeinterval:hasIntervalEndDate : date[0..1]
timeinterval:hasIntervalStartDate : date[0..1]

situation:Situation
atTime : timeinterval:TimeInterval
includesEvent : participation:Event
includesObject : participation:Object
situation:isSettingFor : owl:Thing[1..]

participation:Event
isEventIncludedIn : situation:Situation
participation:hasParticipant : participation:Object

participation:isParticipantIn

participation:hasParticipant

participation:Object
isObjectIncludedIn : situation:Situation
participation:isParticipantIn : participation:Event

TimeIndexedParticipation
atTime : timeinterval:TimeInterval[1..]
includesEvent : participation:Event[1..]
includesObject : participation:Object[1..]
situation:isSettingFor : owl:Thing[3..]

The resulting ontology is composed of the union of the ontology elements and axioms from the two CPs, plus the axioms (e.g. disjointness, equivalence, etc.) that are added in order to link the CPs

Where do CPs come from?

- Content ontology design patterns (CPs) come from the experience of ontology engineers in modeling foundational, core, or domain ontologies

- There are four ways of creating CPs, which can be summarized as follows:
    - Reengineering from patterns expressed in other data models
        - Data model patterns, Lexical Frames, Workflow patterns, Knowledge discovery patterns, etc.
    - Specialization/Generalization/Composition of other CPs
    - Extraction from reference ontologies (by cloning)
    - Creation by combining extraction, specialization, generalization, composition, and expansion

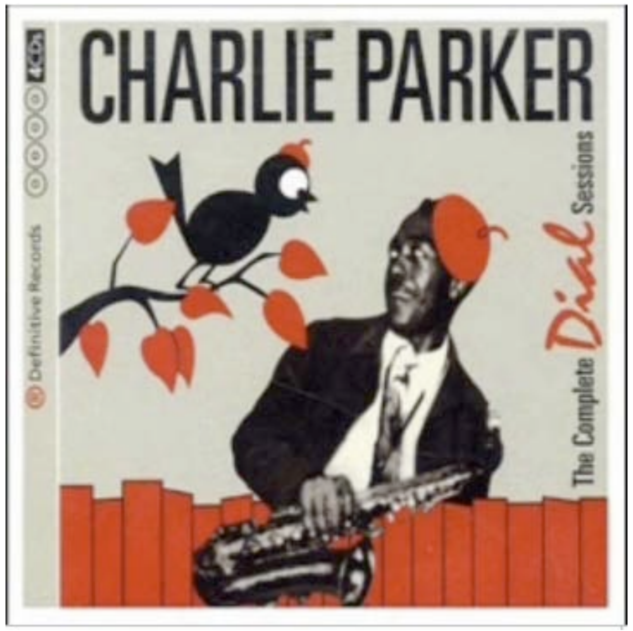
eXtreme ontology Design (XD)

- Inspired by eXtreme Programming basic rules
  - e.g., pair programming, test-oriented, continued integration, etc.
- Main principles
  - divide & conquer
    - understand the task and express it by means of competency questions
  - re-use "good" solutions i.e., ontology design patterns
  - evaluate the result against the task
- As an example, we apply an XD iteration with CPs

Sample XD iteration 1/3

- Sentence: Charlie Parker is the alto sax player on Lover Man, Dial, 1946
  - Charlie Parker (person)
  - the alto sax player (player role)
  - on Lover Man (tune)
  - Dial (publisher)
  - 1946 (recording year)
- CQs

  Alternative abstractions do exist!
  - what persons play a musical instrument?
  - on what tune?
  - for what publisher?
  - in what recording year?
- Queries
  - SELECT ?x ?y WHERE { ?x ?r ?y . ?x a :Person . ?y a :PlayerRole }
  - SELECT ?x ?z WHERE { ?x ?r ?y . ?x a :Person . ?x ?s ?z . ?z a :Tune }
  - SELECT ?z ?w WHERE { ?z ?t ?w . ?z a :Tune . ?w a :Publisher }
  - SELECT ?z ?k WHERE { ?z :recordingYear ?k . ?z a :Tune . ?k a xsd:gYear }



Sample XD iteration 2/3

- Retrieve/Match cqs to CPs, or possibly propose new ones
  - agentrole.owl, timeindexedpersonrole.owl, timeinterval.owl, ...
- Specialize/Compose/Expand CPs to local cq terminology
  - person-playerrole, playing-instrument-on-a-tune, playing-on-a-tune-in-recordingyear
- Populate ABox
  - Person(CharlieParker), PlayerRole(AltoSaxPlayer), Tune(LoverMan), Session(LoverManWithParkerOnDial), ...

Sample XD iteration 3/3

- Run unit test/Iterate until fixed
  - SELECT ?x ?y ?z ?w ?k
  - WHERE {
    - ?x ?r ?y .
    - ?x a :Person .
    - ?y a :PlayerRole .
    - ?x ?s ?z .
    - ?z a :Tune .
    - ?z ?t ?w .
    - ?w a :Publisher .
    - ?z :recordingYear ?k .
    - ?k a xsd:gYear }
  - ?x=CharlieParker ?y=AltoSaxPlayer ?z=LoverMan ?w=Dial ?k=1946

Experiments: first results

- During a four-day course for PhD students
  - Most have never constructed an ontology, or only a small example
  - Mostly taxonomies or lightweight ontologies
  - Most subjects familiar with some modeling language (like ER or UML), but only a few have tried OWL
- Background questionnaires, ontology design exercises (end of every day), subjective feedback questionnaire after exercise
  - first two days no patterns, second two days with patterns
- Some preliminary results based on subjective feedback questionnaires only
  - Main difficulties: mapping from the problem to the patterns, pattern composition
  - Most found the patterns useful and many perceived that they introduced some solution they did not think of themselves
  - Most perceived the second exercise as the easiest to solve, and the fourth as the most successfully modeled
  - The last day we have also got pattern proposals

Ongoing and future work

- ✓ Bootstrapping and improving functionalities in the ODP portal
- ✓ ODP APIs
- Building the NeOn Toolkit ODP plugin
- Continue with experimentation
- Use of CBR for pattern-based automatic ontology construction ... authority moves behind the curtains: do we need to redefine the theory of numbers in order to pay the grocery account?

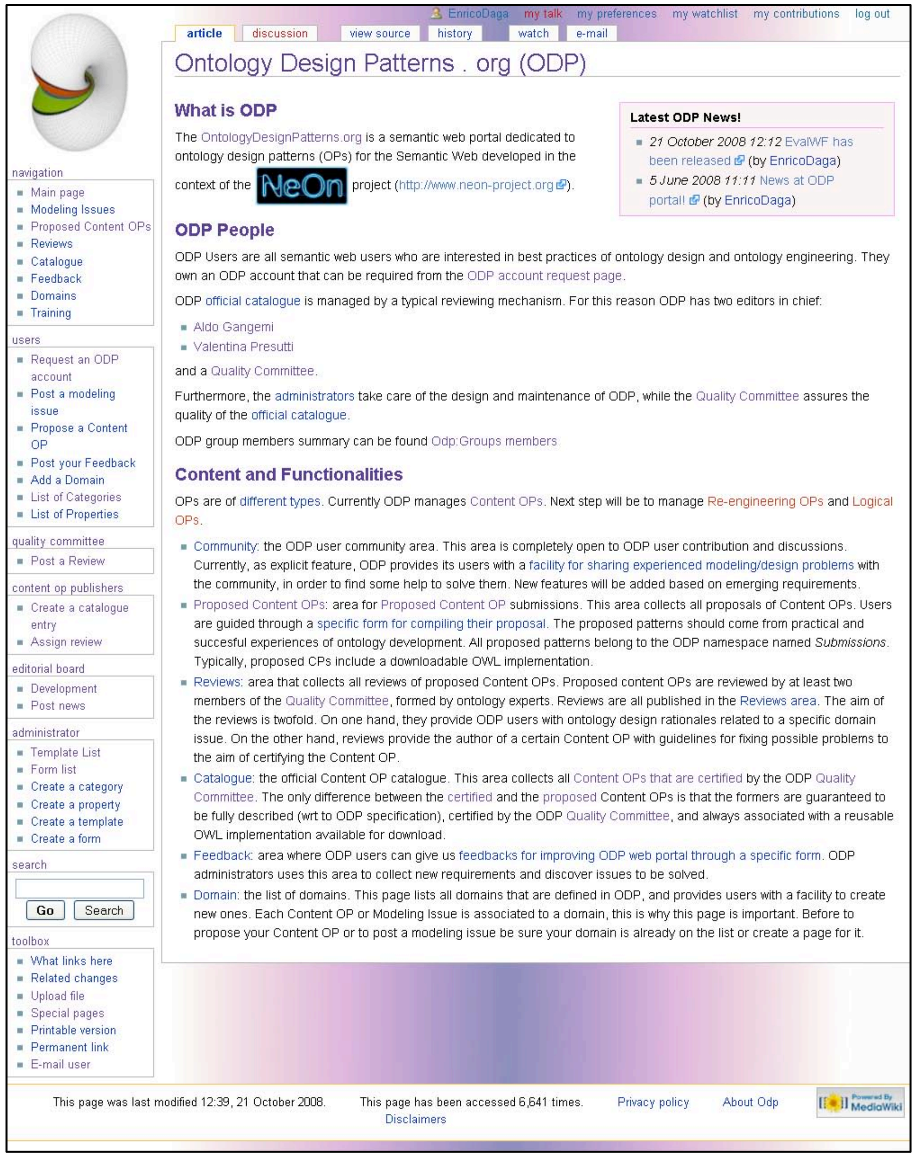
- Join the ODP community! *http://www.ontologydesignpatterns.org*

Outline

- Designing Computational Ontologies
- Ontology Design Patterns
- ontologydesignpatterns.org initiative

ontologydesignpatterns.org (ODP)

a semantic web portal



EnricoDaga my talk my preferences my watchlist my contributions log out

article discussion view source history watch e-mail

Ontology Design Patterns . org (ODP)

What is ODP

The OntologyDesignPatterns.org is a semantic web portal dedicated to ontology design patterns (OPs) for the Semantic Web developed in the context of the NeOn project (http://www.neon-project.org).

Latest ODP News!

- *21 October 2008 12:12* EvalWF has been released (by EnricoDaga)
- *5 June 2008 11:11* News at ODP portal! (by EnricoDaga)

ODP People

ODP Users are all semantic web users who are interested in best practices of ontology design and ontology engineering. They own an ODP account that can be required from the ODP account request page.

ODP official catalogue is managed by a typical reviewing mechanism. For this reason ODP has two editors in chief:

- Aldo Gangemi
- Valentina Presutti

and a Quality Committee.

Furthermore, the administrators take care of the design and maintenance of ODP, while the Quality Committee assures the quality of the official catalogue.

ODP group members summary can be found Odp:Groups members

Content and Functionalities

OPs are of different types. Currently ODP manages Content OPs. Next step will be to manage Re-engineering OPs and Logical OPs.

- Community: the ODP user community area. This area is completely open to ODP user contribution and discussions. Currently, as explicit feature, ODP provides its users with a facility for sharing experienced modeling/design problems with the community, in order to find some help to solve them. New features will be added based on emerging requirements.
- Proposed Content OPs: area for Proposed Content OP submissions. This area collects all proposals of Content OPs. Users are guided through a specific form for compiling their proposal. The proposed patterns should come from practical and succesful experiences of ontology development. All proposed patterns belong to the ODP namespace named *Submissions*. Typically, proposed CPs include a downloadable OWL implementation.
- Reviews: area that collects all reviews of proposed Content OPs. Proposed content OPs are reviewed by at least two members of the Quality Committee, formed by ontology experts. Reviews are all published in the Reviews area. The aim of the reviews is twofold. On one hand, they provide ODP users with ontology design rationales related to a specific domain issue. On the other hand, reviews provide the author of a certain Content OP with guidelines for fixing possible problems to the aim of certifying the Content OP.
- Catalogue: the official Content OP catalogue. This area collects all Content OPs that are certified by the ODP Quality Committee. The only difference between the certified and the proposed Content OPs is that the formers are guaranteed to be fully described (wrt to ODP specification), certified by the ODP Quality Committee, and always associated with a reusable OWL implementation available for download.
- Feedback: area where ODP users can give us feedbacks for improving ODP web portal through a specific form. ODP administrators uses this area to collect new requirements and discover issues to be solved.
- Domain: the list of domains. This page lists all domains that are defined in ODP, and provides users with a facility to create new ones. Each Content OP or Modeling Issue is associated to a domain, this is why this page is important. Before to propose your Content OP or to post a modeling issue be sure your domain is already on the list or create a page for it.

navigation

- Main page
- Modeling Issues
- Proposed Content OPs
- Reviews
- Catalogue
- Feedback
- Domains
- Training

users

- Request an ODP account
- Post a modeling issue
- Propose a Content OP
- Post your Feedback
- Add a Domain
- List of Categories
- List of Properties

quality committee

- Post a Review

content op publishers

- Create a catalogue entry
- Assign review

editorial board

- Development
- Post news

administrator

- Template List
- Form list
- Create a category
- Create a property
- Create a template
- Create a form

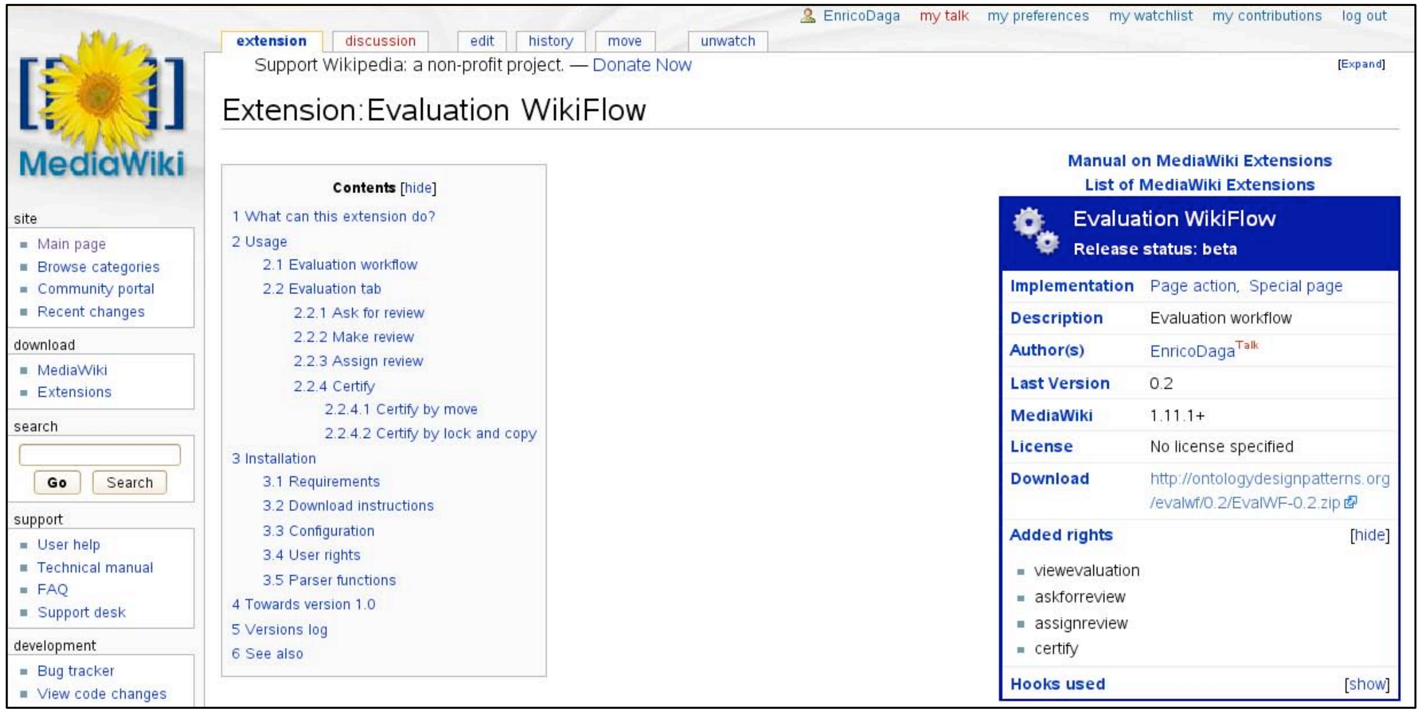
search

Go Search

toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version
- Permanent link
- E-mail user

This page was last modified 12:39, 21 October 2008. This page has been accessed 6,641 times. Privacy policy About Odp Disclaimers



EnricoDaga my talk my preferences my watchlist my contributions log out

extension discussion edit history move unwatch

Support Wikipedia: a non-profit project. — Donate Now [Expand]

Extension:Evaluation WikiFlow

Contents [hide]

1 What can this extension do?
2 Usage
2.1 Evaluation workflow
2.2 Evaluation tab
2.2.1 Ask for review
2.2.2 Make review
2.2.3 Assign review
2.2.4 Certify
2.2.4.1 Certify by move
2.2.4.2 Certify by lock and copy
3 Installation
3.1 Requirements
3.2 Download instructions
3.3 Configuration
3.4 User rights
3.5 Parser functions
4 Towards version 1.0
5 Versions log
6 See also

Manual on MediaWiki Extensions
List of MediaWiki Extensions

Evaluation WikiFlow **Release status: beta**	
Implementation	Page action, Special page
Description	Evaluation workflow
Author(s)	EnricoDaga[Talk]
Last Version	0.2
MediaWiki	1.11.1+
License	No license specified
Download	http://ontologydesignpatterns.org/evalwf/0.2/EvalWF-0.2.zip
Added rights	[hide]

- viewevaluation
- askforreview
- assignreview
- certify

Hooks used [show]

site

- Main page
- Browse categories
- Community portal
- Recent changes

download

- MediaWiki
- Extensions

search

Go Search

support

- User help
- Technical manual
- FAQ
- Support desk

development

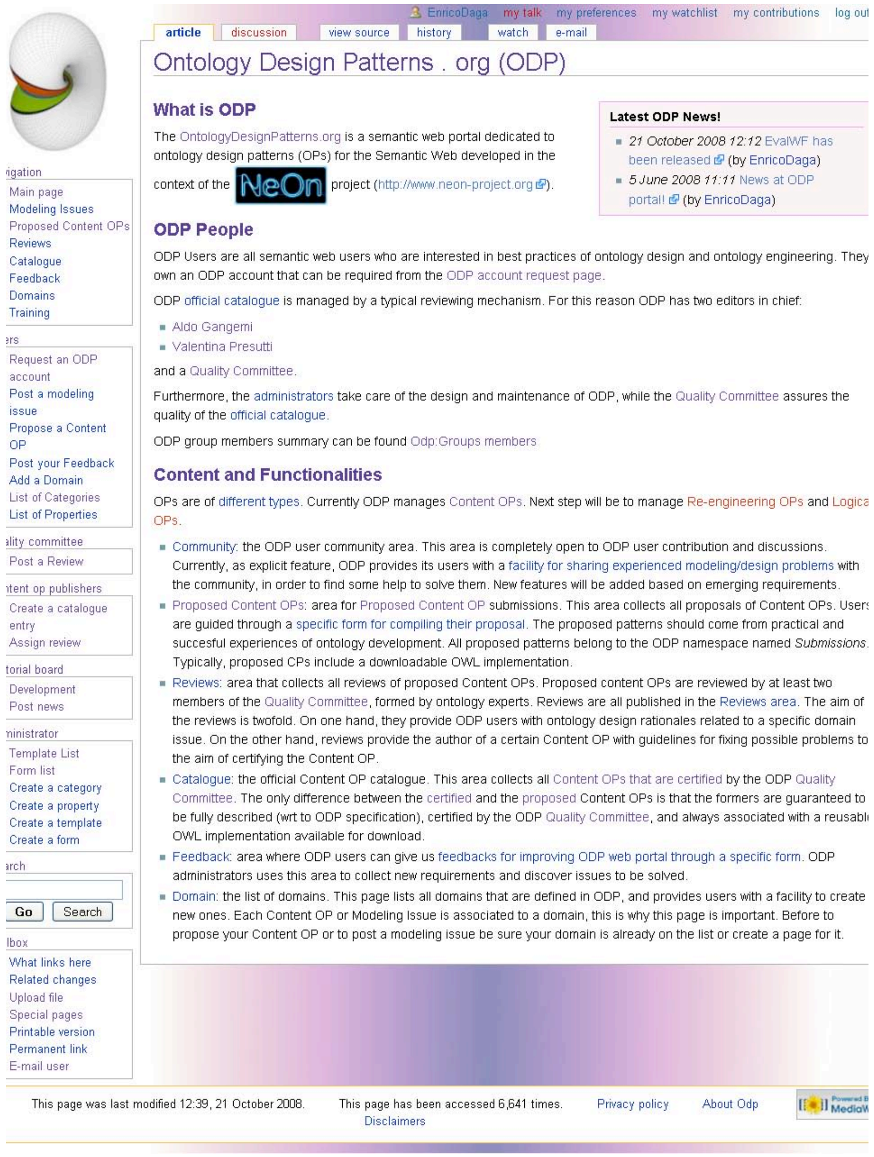
- Bug tracker
- View code changes

Evaluation WikiFlow

a Semantic MediaWiki extension

ontologydesignpatterns.org (ODP)

- A semantic web portal about OPs (Logical, Content, Presentation ...)
- currently supports CPs
- best practices for ontology design and ontology engineering
- evaluation, training and repository of reusable OWL ontologies



Ontology Design Patterns . org (ODP)

What is ODP

The OntologyDesignPatterns.org is a semantic web portal dedicated to ontology design patterns (OPs) for the Semantic Web developed in the context of the NeOn project (http://www.neon-project.org).

Latest ODP News!

- *21 October 2008 12:12* EvalWF has been released (by EnricoDaga)
- *5 June 2008 11:11* News at ODP portal! (by EnricoDaga)

ODP People

ODP Users are all semantic web users who are interested in best practices of ontology design and ontology engineering. They own an ODP account that can be required from the ODP account request page.

ODP official catalogue is managed by a typical reviewing mechanism. For this reason ODP has two editors in chief:

- Aldo Gangemi
- Valentina Presutti

and a Quality Committee.

Furthermore, the administrators take care of the design and maintenance of ODP, while the Quality Committee assures the quality of the official catalogue.

ODP group members summary can be found Odp:Groups members

Content and Functionalities

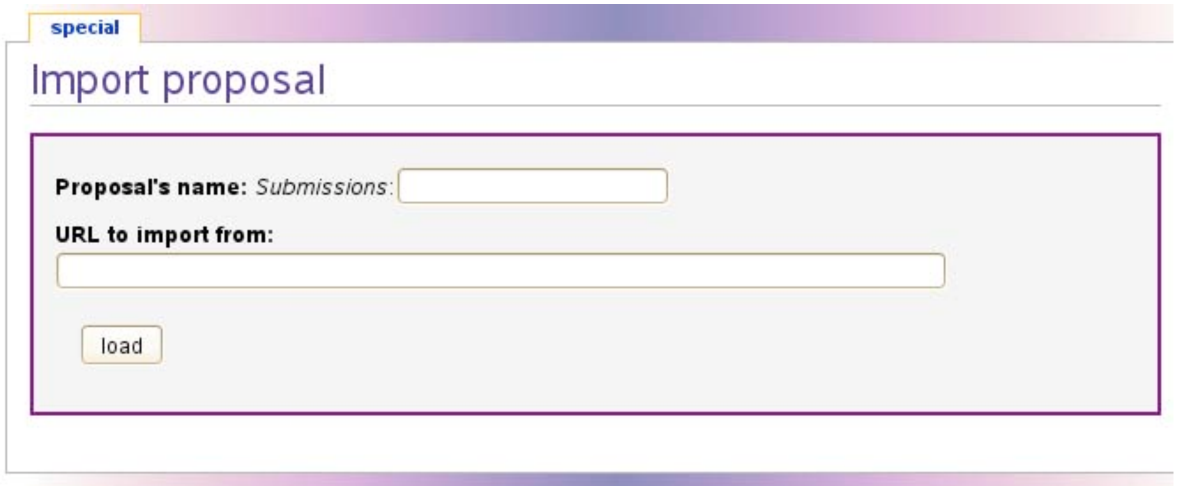
OPs are of different types. Currently ODP manages Content OPs. Next step will be to manage Re-engineering OPs and Logica OPs.

- Community: the ODP user community area. This area is completely open to ODP user contribution and discussions. Currently, as explicit feature, ODP provides its users with a facility for sharing experienced modeling/design problems with the community, in order to find some help to solve them. New features will be added based on emerging requirements.
- Proposed Content OPs: area for Proposed Content OP submissions. This area collects all proposals of Content OPs. Users are guided through a specific form for compiling their proposal. The proposed patterns should come from practical and succesful experiences of ontology development. All proposed patterns belong to the ODP namespace named *Submissions*. Typically, proposed CPs include a downloadable OWL implementation.
- Reviews: area that collects all reviews of proposed Content OPs. Proposed content OPs are reviewed by at least two members of the Quality Committee, formed by ontology experts. Reviews are all published in the Reviews area. The aim of the reviews is twofold. On one hand, they provide ODP users with ontology design rationales related to a specific domain issue. On the other hand, reviews provide the author of a certain Content OP with guidelines for fixing possible problems to the aim of certifying the Content OP.
- Catalogue: the official Content OP catalogue. This area collects all Content OPs that are certified by the ODP Quality Committee. The only difference between the certified and the proposed Content OPs is that the formers are guaranteed to be fully described (wrt to ODP specification), certified by the ODP Quality Committee, and always associated with a reusabl OWL implementation available for download.
- Feedback: area where ODP users can give us feedbacks for improving ODP web portal through a specific form. ODP administrators uses this area to collect new requirements and discover issues to be solved.
- Domain: the list of domains. This page lists all domains that are defined in ODP, and provides users with a facility to create new ones. Each Content OP or Modeling Issue is associated to a domain, this is why this page is important. Before to propose your Content OP or to post a modeling issue be sure your domain is already on the list or create a page for it.

This page was last modified 12:39, 21 October 2008. This page has been accessed 6,641 times.

ODP areas

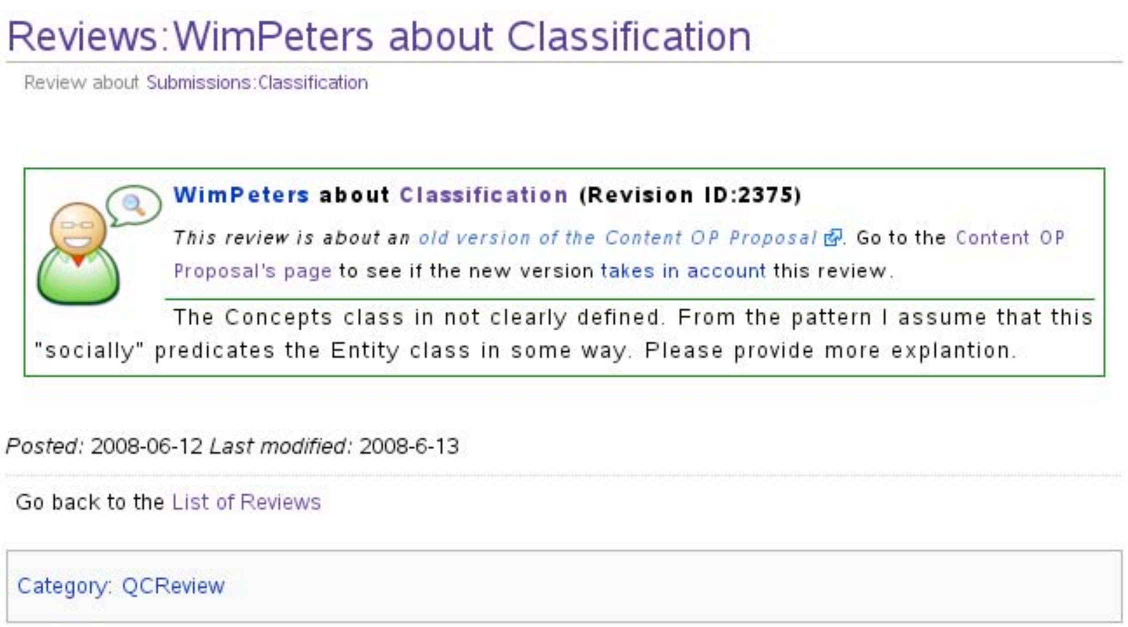
- Community: share experience, collects modeling issues and domains
- Proposals: collects ProposedCP.
  - Submissions by form and import facility
- Reviews: guidelines and rationales.
  - Open reviews and Quality Committee reviews
- Catalogue: collects CertifiedCP (complete, reusable, well-done)
- Training: tutorials, exercises
- Feedbacks: from users' feedback to development tasks (Editorial Board)



Import proposal

Proposal's name: *Submissions:*

URL to import from:



Reviews:WimPeters about Classification

Review about Submissions:Classification

WimPeters about Classification (Revision ID:2375)

This review is about an old version of the Content OP Proposal. Go to the Content OP Proposal's page to see if the new version takes in account this review.

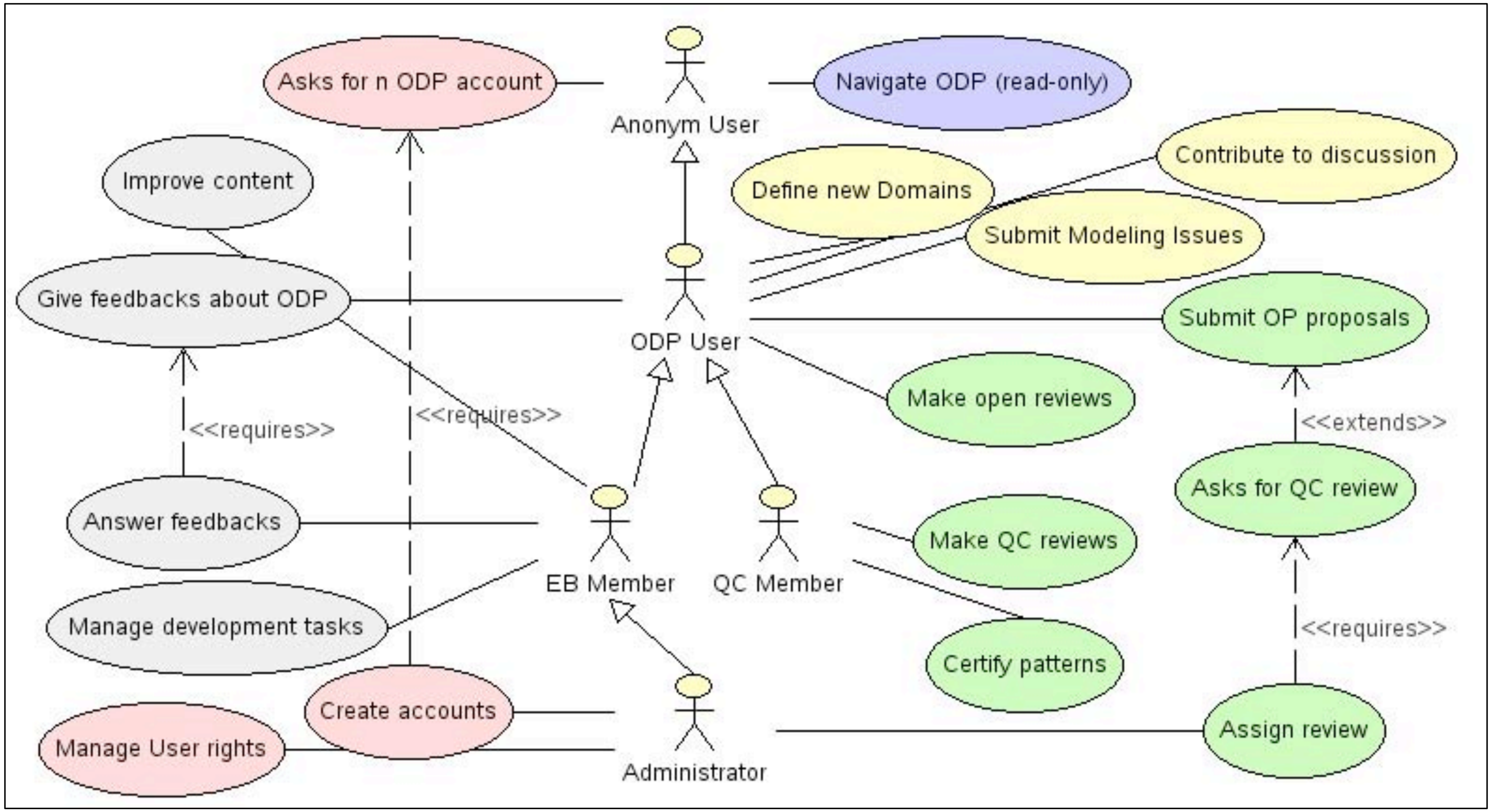
The Concepts class in not clearly defined. From the pattern I assume that this "socially" predicates the Entity class in some way. Please provide more explantion.

Posted: 2008-06-12 *Last modified:* 2008-6-13

Go back to the List of Reviews

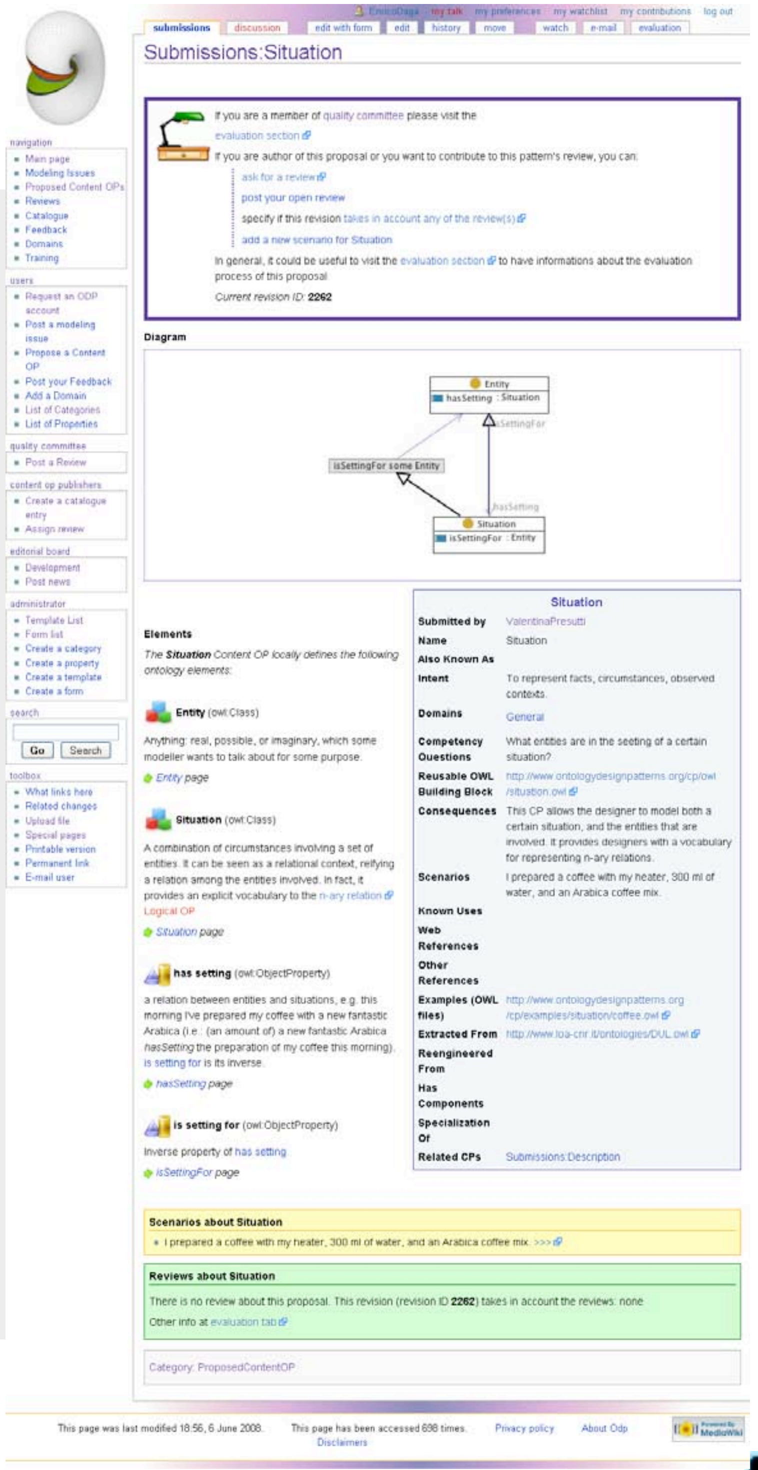
Category: QCReview

ODP types of user



ODP content

- Semantic representation
  - Semantic MediaWiki (SMW) and Semantic Forms (SF) + exts
- Each pattern is described by:
  - diagram
  - annotations (user, name, intent, domain(s), competency questions, known uses, consequences, OWL file, related CPs and Ontologies
  - elements (list, description)
  - scenarios
  - reviews



Submissions:Situation

If you are a member of quality committee please visit the evaluation section

If you are author of this proposal or you want to contribute to this pattern's review, you can:

ask for a review

post your open review

specify if this revision takes in account any of the review(s)

add a new scenario for Situation

In general, it could be useful to visit the evaluation section to have informations about the evaluation process of this proposal

Current revision ID: **2262**

Diagram

Elements

The ***Situation*** *Content OP locally defines the following ontology elements:*

Entity (owl:Class)

Anything: real, possible, or imaginary, which some modeller wants to talk about for some purpose.

Entity page

Situation (owl:Class)

A combination of circumstances involving a set of entities. It can be seen as a relational context, reifying a relation among the entities involved. In fact, it provides an explicit vocabulary to the n-ary relation Logical OP

Situation page

has setting (owl:ObjectProperty)

a relation between entities and situations, e.g. this morning I've prepared my coffee with a new fantastic Arabica (i.e.: (an amount of) a new fantastic Arabica *hasSetting* the preparation of my coffee this morning). is setting for is its inverse.

hasSetting page

is setting for (owl:ObjectProperty)

Inverse property of has setting

isSettingFor page

Situation

Submitted by	ValentinaPresutti
Name	Situation
Also Known As	
Intent	To represent facts, circumstances, observed contexts.
Domains	General
Competency Questions	What entities are in the seeting of a certain situation?
Reusable OWL Building Block	http://www.ontologydesignpatterns.org/cp/owl/situation.owl
Consequences	This CP allows the designer to model both a certain situation, and the entities that are involved. It provides designers with a vocabulary for representing n-ary relations.
Scenarios	I prepared a coffee with my heater, 300 ml of water, and an Arabica coffee mix.
Known Uses	
Web References	
Other References	
Examples (OWL files)	http://www.ontologydesignpatterns.org/cp/examples/situation/coffee.owl
Extracted From	http://www.loa-cnr.it/ontologies/DUL.owl
Reengineered From	
Has Components	
Specialization Of	
Related CPs	Submissions:Description

Scenarios about Situation

- I prepared a coffee with my heater, 300 ml of water, and an Arabica coffee mix. >>>

Reviews about Situation

There is no review about this proposal. This revision (revision ID **2262**) takes in account the reviews: none

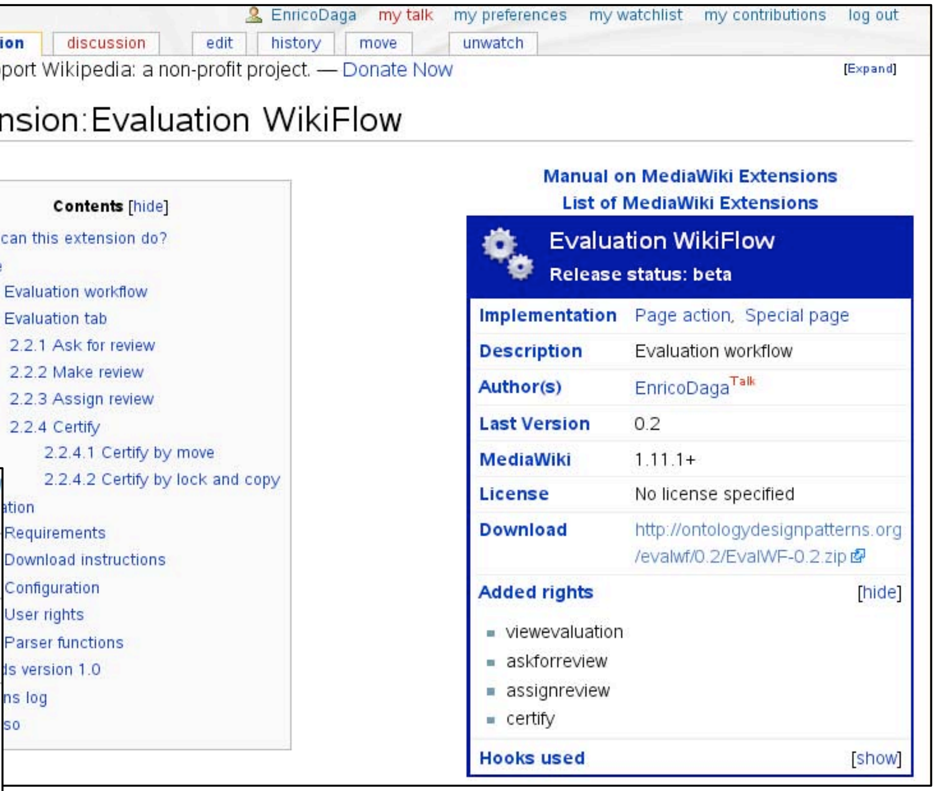
Other info at evaluation tab

Category: ProposedContentOP

This page was last modified 18:56, 6 June 2008. This page has been accessed 698 times. Privacy policy About Odp Disclaimers

Evaluation WikiFlow

- Extends MW, SMW and SF extension
- Evaluation tab



Support Wikipedia: a non-profit project. — Donate Now

Extension:Evaluation WikiFlow

Contents [hide]

1 What can this extension do?
2 Usage
2.1 Evaluation workflow
2.2 Evaluation tab
2.2.1 Ask for review
2.2.2 Make review
2.2.3 Assign review
2.2.4 Certify
2.2.4.1 Certify by move
2.2.4.2 Certify by lock and copy
ation
Requirements
Download instructions
Configuration
User rights
Parser functions
ls version 1.0
ns log
so

Manual on MediaWiki Extensions
List of MediaWiki Extensions

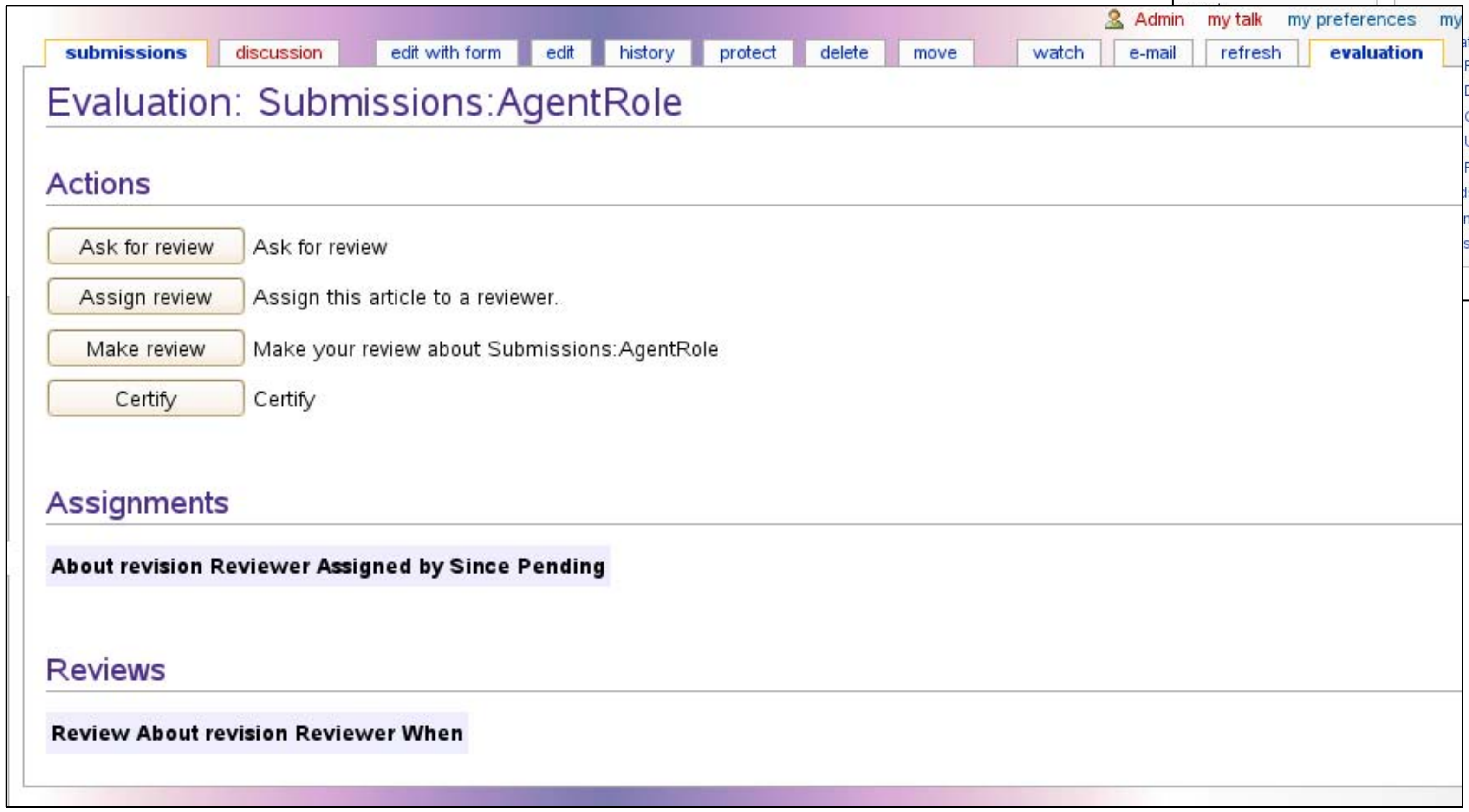
Evaluation WikiFlow
Release status: beta

Implementation	Page action, Special page
Description	Evaluation workflow
Author(s)	EnricoDaga[Talk]
Last Version	0.2
MediaWiki	1.11.1+
License	No license specified
Download	http://ontologydesignpatterns.org/evalwf/0.2/EvalWF-0.2.zip

Added rights [hide]

- viewevaluation
- askforreview
- assignreview
- certify

Hooks used [show]



Evaluation: Submissions:AgentRole

Actions

Ask for review	Ask for review
Assign review	Assign this article to a reviewer.
Make review	Make your review about Submissions:AgentRole
Certify	Certify

Assignments

About revision	Reviewer	Assigned by	Since	Pending

Reviews

Review	About revision	Reviewer	When

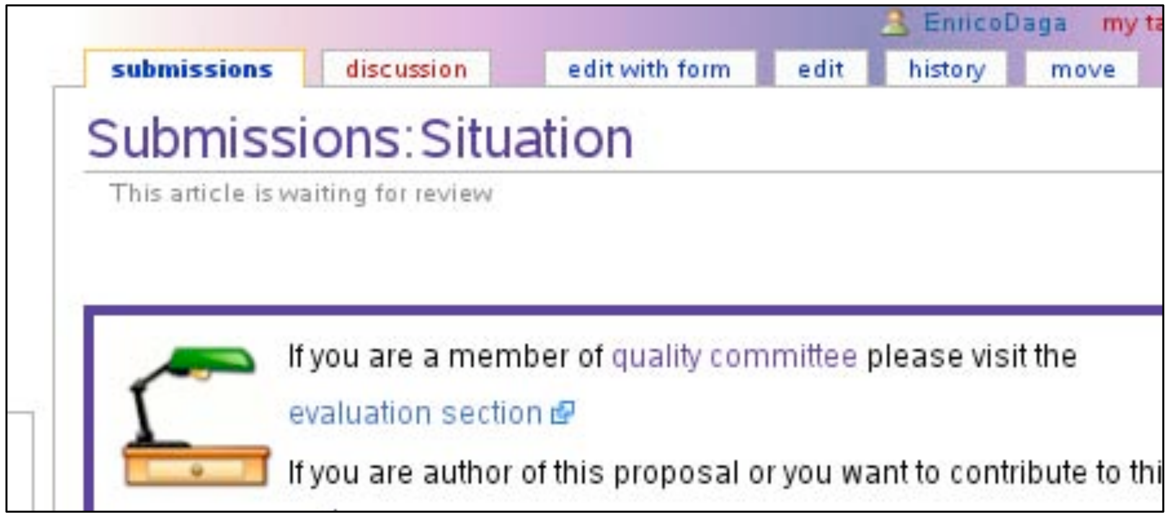
- Features:
  - **configuration**
  - **functionality**

Evaluation WikiFlow: configuration

- Activation tab
  - categories to evaluate.
  - e.g. currently ODP activates it for the ProposedCP category.
- Review schema(s) customization
  - different review schemas can be defined
- Category/review association
  - categories to evaluate with review schemas
    - E.g. ProposedLP and ProposedCP have different review schemas
- User rights configuration
  - view, ask for, assign, make, certify
  - E.g. QualityCommity members make reviews, while every ODPUser can request reviews

Evaluation WikiFlow: functionality

- ***ask for review***: `+WaitingForReview`
- ***assign review***: `+AssignedReview`
- ***make review***: `-WaitingForReview, +AssignedReview`.
- ***certify***: `+Certified`, freezed; new lifecycle.
- ***semantic report of evaluation history***
  - *aim: to analyze rationales behind evaluation of design patterns*

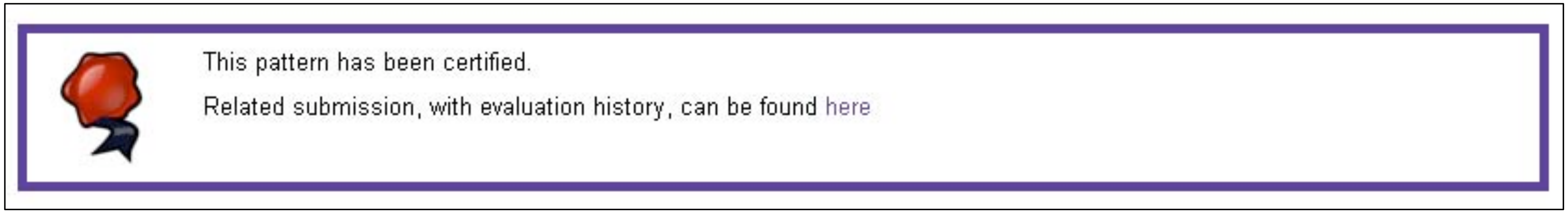


Submissions:Situation

This article is waiting for review

If you are a member of quality committee please visit the evaluation section

If you are author of this proposal or you want to contribute to thi

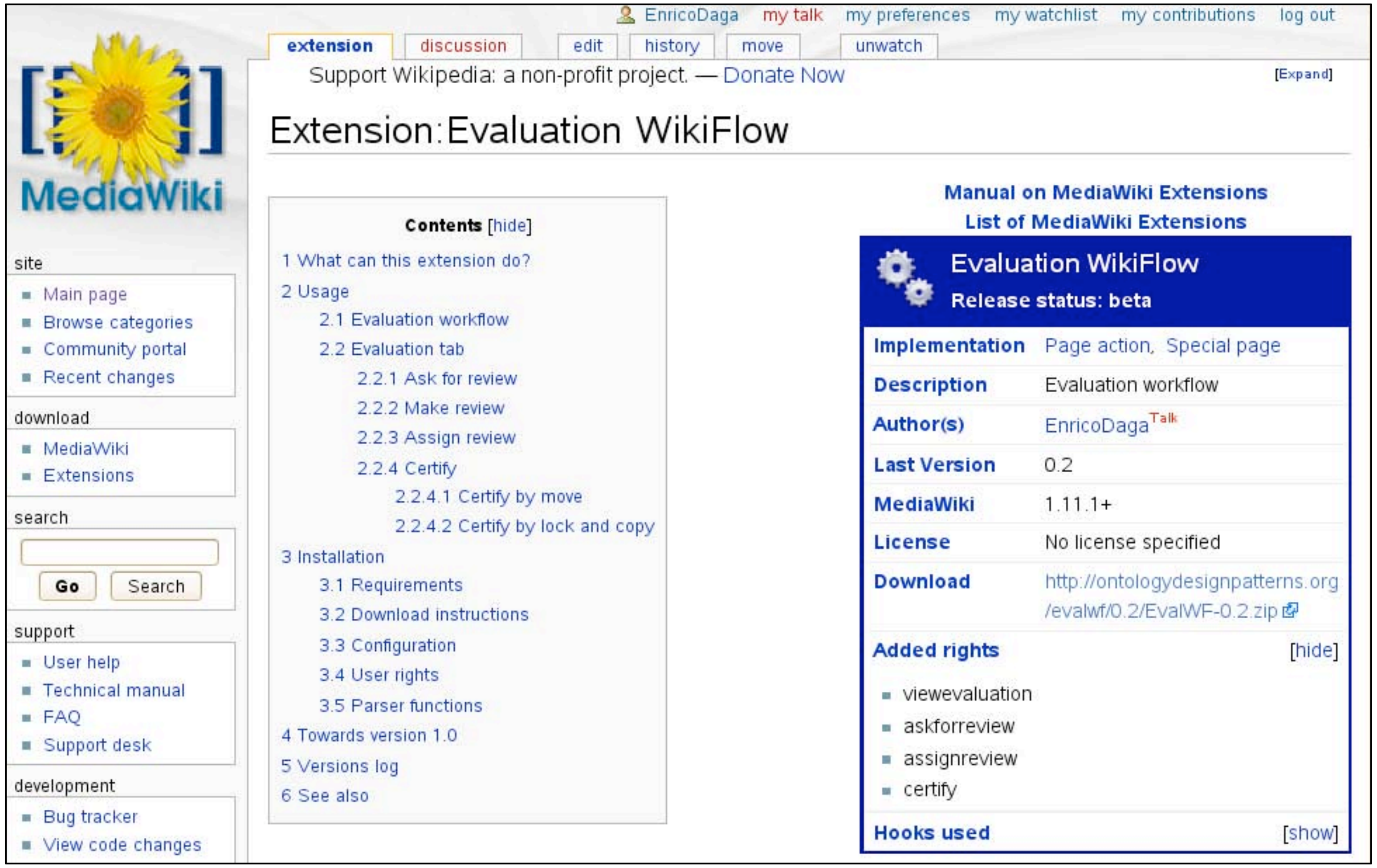


This pattern has been certified.

Related submission, with evaluation history, can be found here

Evaluation WikiFlow: software

- *alpha version* as **open source software**
- can be downloaded from the MediaWiki wiki site
  - http://www.mediawiki.org/wiki/Extension:Evaluation_WikiFlow



Support Wikipedia: a non-profit project. — Donate Now [Expand]

Extension:Evaluation WikiFlow

Contents [hide]

1 What can this extension do?
2 Usage
2.1 Evaluation workflow
2.2 Evaluation tab
2.2.1 Ask for review
2.2.2 Make review
2.2.3 Assign review
2.2.4 Certify
2.2.4.1 Certify by move
2.2.4.2 Certify by lock and copy
3 Installation
3.1 Requirements
3.2 Download instructions
3.3 Configuration
3.4 User rights
3.5 Parser functions
4 Towards version 1.0
5 Versions log
6 See also

Manual on MediaWiki Extensions
List of MediaWiki Extensions

Evaluation WikiFlow
Release status: beta

Implementation	Page action, Special page
Description	Evaluation workflow
Author(s)	EnricoDaga[Talk]
Last Version	0.2
MediaWiki	1.11.1+
License	No license specified
Download	http://ontologydesignpatterns.org/evalwf/0.2/EvalWF-0.2.zip

Added rights [hide]

- viewevaluation
- askforreview
- assignreview
- certify

Hooks used [show]

Conclusion and future work

- Ontologydesignpatterns.org and Evaluation WikiFlow
  - A community-based web portal (training, discussion and repository)
  - A domain-independent extension for SMW and SF
- ODP ongoing and planned work includes
  - new types of ontology design patterns
    - e.g Logical, Reengineering

      1st f2f editorial board meeting on Feb 23rd)
  - a search service based on Watson

    http://watson.kmi.open.ac.uk
  - the ODP repository APIs
  - OWL/RDF export service
  - an open rating system for open reviews (based on NeOn ORS)
  - statistical monitoring of CP downloads to be used as a dimension of user-based evaluation of CPs and ODP usage

References (1)

- 1. Valentina Presutti, Aldo Gangemi, Stefano David, Guadalupe Aguado de Cea, Mari-Carmen Suarez Figueroa, Elena Montiel-Ponsoda, and Marıa Poveda. Library of design patterns for collaborative development of networked ontologies. Deliverable D2.5.1, NeOn project, 2008.
- 2. Mari Carmen Suarez-Figueroa, Saartje Brockmans, Aldo Gangemi, Asuncion Gomez-Perez, Jos Lehmann, Holger Lewen, Valentina Presutti, and Marta Sabou. Neon modelling components. Deliverable D5.1.1, NeOn project, 2007.
- 3. Carola Catenacci, Jos Lehmann, Malvina Nissim, Valentina Presutti, and Geri Steve. Design rationales for collaborative development of networked ontologies state of the art and the collaborative ontology design ontology. Deliverable D2.1.1, NeOn project, 2007.
- 4. Aldo Gangemi, Jos Lehmann, Valentina Presutti, Malvina Nissim, and Carola Catenacci. C-ODO: an OWL meta-model for collaborative ontology design. Workshop on Social and Collaborative Construction of Structured Knowledge (CKC 2007) at WWW 2007, Banff, Canada, (2007).
- 5. Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, Jos Lehmann. Modelling Ontology Evaluation and Validation, Y. Sure (ed.), Proceedings of the Third European Semantic Web Conference, Springer, 2006.
- 6. Denny Vrandecic, York Sure, Raul Palma, and Francisco Santana. Ontology repository and content evaluation. Deliverable D1.2.10v2, KnowledgeWeb project, 2007.
- 7. Denny Vrandecic. Explicit Knowledge Engineering Patterns with Macros. In Proceedings of the Ontology Patterns for the Semantic Web Workshop at the ISWC 2005, Galway, Ireland, November2005.
- 8. Aldo Gangemi. Ontology Design Patterns for Semantic Web Content.In Proceedings of the 4th International Semantic Web Conference (ISWC2005), volume3729ofLNCS, Springer Verlag Berlin-Heidelberg, November2005.

References (2)

- 9. Catalogue of ODPs focused on the biological knowledge domain, http://odps.sourceforge.net/
- 10. W3C Ontology Engineering and Patterns Task Force (OEP), http://www.w3.org/2001/sw/BestPractices/OEP/
- 11. M. Gruninger and M. Fox. The role of competency questions in enterprise engineering. In Proceedings of the IFIP WG5.7 Workshop on Benchmarking Theory and Practice, Trondheim, Norway, 1994.
- 12. Denny Vrandecic and Aldo Gangemi. Unit tests for ontologies. In Proceedings of the 1st International Workshop on Ontology content and evaluation in Enterprise, Montpellier, France, Springer, OCT2006.
- 13. Eva Blomqvist. Fully automatic construction of enterprise ontologies using design patterns: Initial method and first experiences. In Robert Meersman, Zahir Tari, Mohand-Said Hacid, John Mylopoulos, Barbara Pernici, Ozalp Babaoglu, Hans-Arno Jacobsen, Joseph P. Loyall, Michael Kifer, and Stefano Spaccapietra, editors, OTM Conferences (2), volume 3761 of Lecture Notes in Computer Science, pages 1314–1329. Springer, 2005.
- 14. Valentina Presutti and Aldo Gangemi. Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. In Proceedings of the 27th International Conference on Conceptual Modeling (ER 2008)
- 15. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini. Identification Constraints and Functional Dependencies in Description Logics, IJCAI 2001.
- 16. François Scharffe, Jérome Euzenat, Ying Ding and Dieter Fensel. Correspondence Patterns for Ontology Mediation. In Proceedings of the Ontology Matching Workshop at ISWC, 2007.