Ontology Design

Aldo Gangemi

Semantic Technology Lab

ISTC-CNR, Rome, Italy

aldo.gangemi@cnr.it

STLab
The Semantic Technology Lab
ISTC-CNR Rome

Thanks to: Valentina Presutti and the members of the STLab

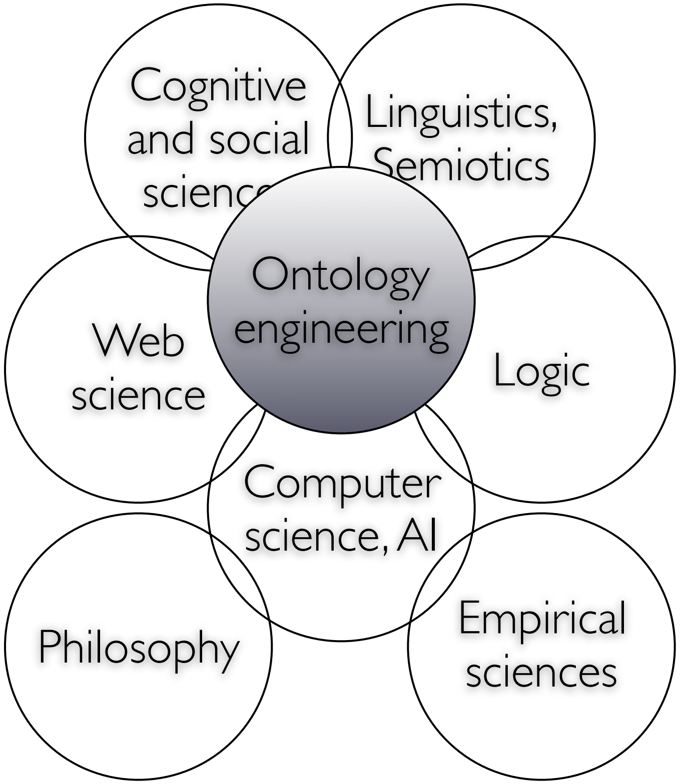
Outline

- The world of ontology design
- Ontologies and language
- Ontology design components
- Ontology design patterns
- Sample design issues and unit tests
- Summary

An ontology designer's world

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

The cultural context of ontologies



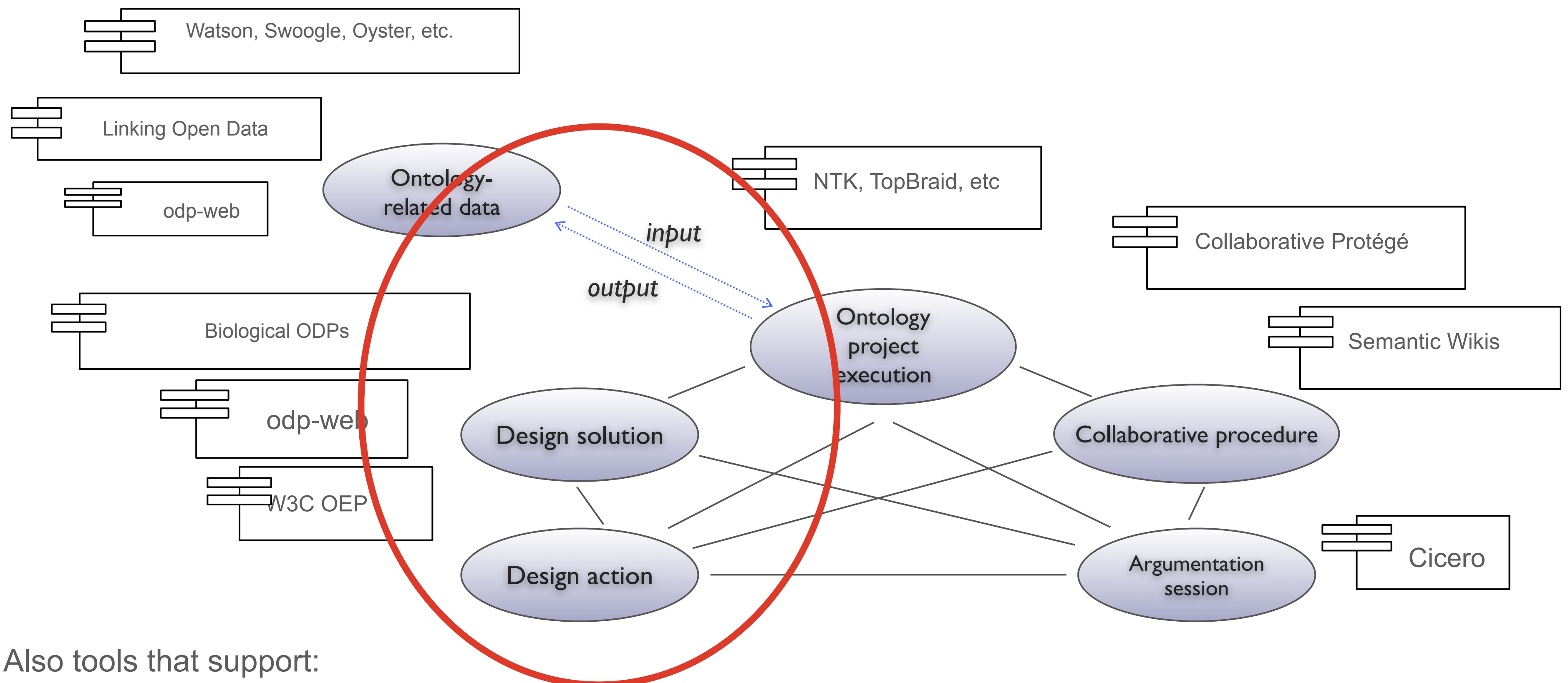
A well-designed ontology ...

- Obeys to “capital questions”:
  - What are we talking about?
  - Why do we want to talk about it?
  - Where to find reusable knowledge?
  - [also: 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

What is ontology design?

- Ontologies are artifacts
  - Have a structure (linguistic, “taxonomical”, logical)
  - Their function is to “encode” a description of the world (actual, possible, counterfactual, impossible, desired, etc.) for some purpose, e.g. the world of Semantic Web conferences
- Ontologies must match both domain and task
  - Allow the description of the entities (“domain”) whose attributes and relations are concerned by some purpose, e.g. *research topics as entities that are dealt with by a project, worked on by academic staff, and can be topic of documents, events, etc.*
  - Serve a purpose (“task”), e.g. *finding persons that work on a same topic*, *matching project topics to staff competencies, time left, available funds*, etc.
- Ontologies have a lifecycle
  - 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 patterns***

Design in C-ODO



Also tools that support:
pattern-based design
evaluation and selection
rengineering
reasoning and querying
evolution and mapping

Collaborative Ontology Design Components

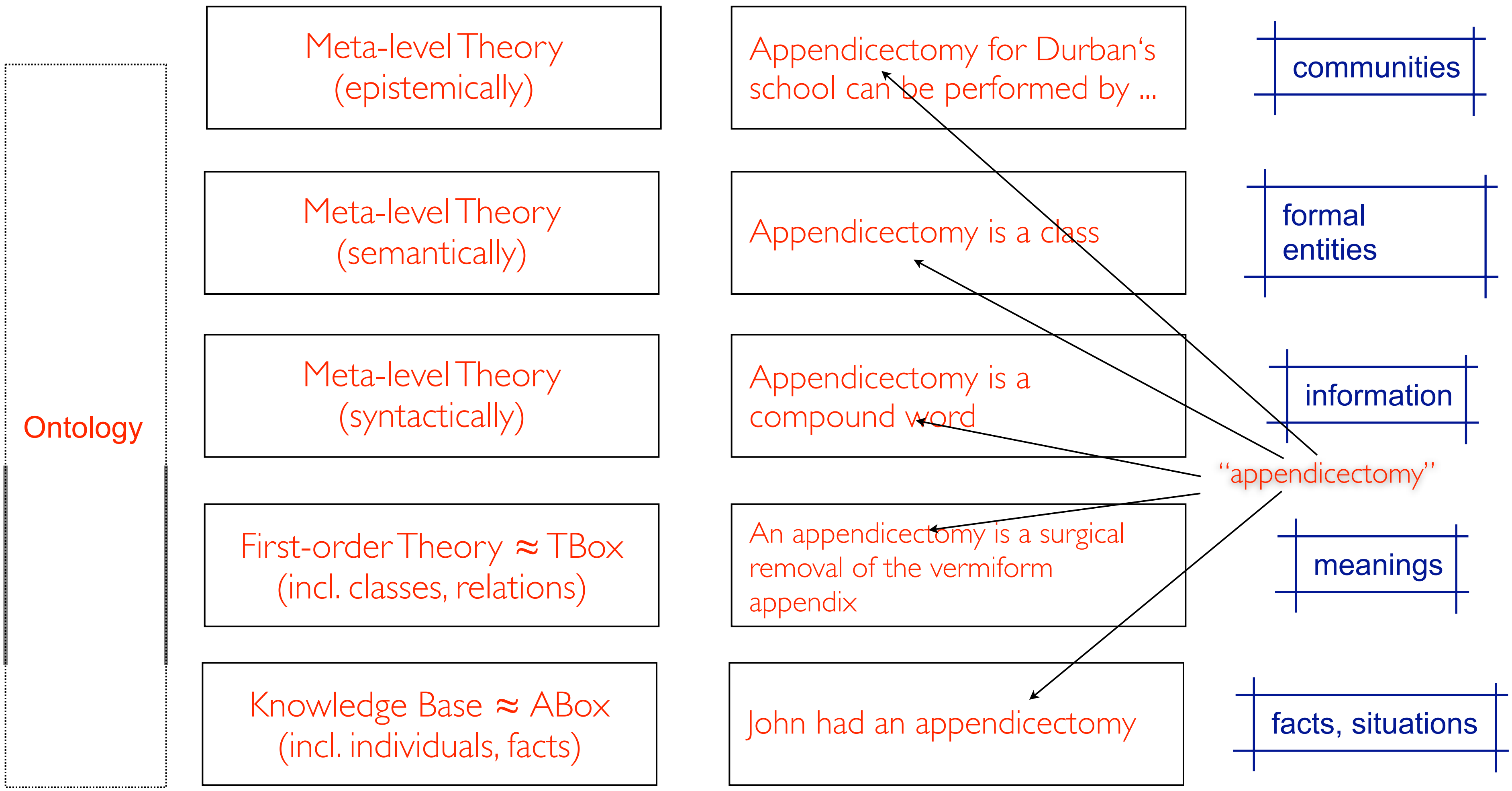
Ontologies and language

- Ontologies describe some domain (for some purpose)
- But also natural language can do it
- Ok, but natural languages are appropriate for humans, not for machines
- What’s the difference?
  - Humans share tacit knowledge (“presuppositions”) that provides the context for interpreting natural language utterances and texts
  - Some tacit knowledge is general
    - *US Army auditor who attacked Halliburton deal is fired*
    - ↳ auditor is a role played by persons within organizations
    - ↳ persons can “attack” others by denouncing something (e.g. a deal)
    - ↳ persons can be “fired” from a position (role)
  - Some is local
    - *US Army auditor who attacked Halliburton deal is fired*
    - ↳ denounced the decision to give billions of dollars in Iraq reconstruction contracts to a subsidiary of Vice-President Dick Cheney's old company Halliburton
    - ↳ “She told a congressional hearing that the decision was "the most blatant and improper abuse I have witnessed" in 20 years as a government contract supervisor”

Ontologies = controlled terminologies?

- Beware the mismatch between language and conceptualization!
- An ontology may not just be a controlled terminology
- We may have to capture the conceptual schema (or pattern) underlying the use of a certain terminology, in order to make it reusable for design, interoperability, meaning negotiation, etc.
- Should ontologies be considered reference conceptual schemas?
- Indeed, that was the original motivation for ontologies. Cf. Ontolingua library, 1992
  - http://www-ksl-svc.stanford.edu:5915
- Nowadays, it's pretty different
  - Thousands of ontologies, many different uses, the most successful are very simple (DublinCore, FOAF, WSGeo, ...), huge uptake on folksonomies
- Need for simple schemas, which are close to users' way of thinking

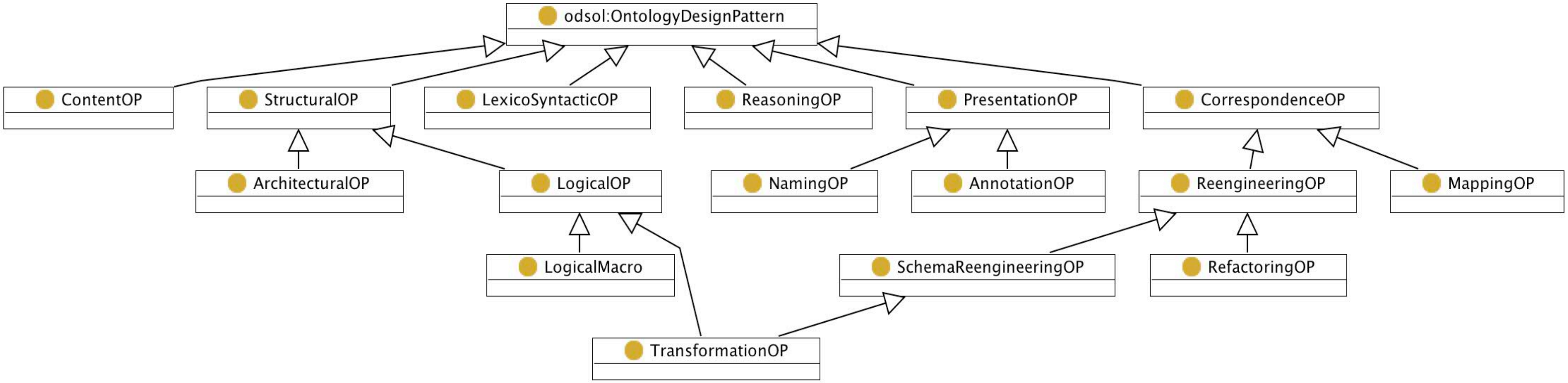
Logical layers, types of entities, and contexts



Pattern-based design

- Ontology design is presented here as 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*

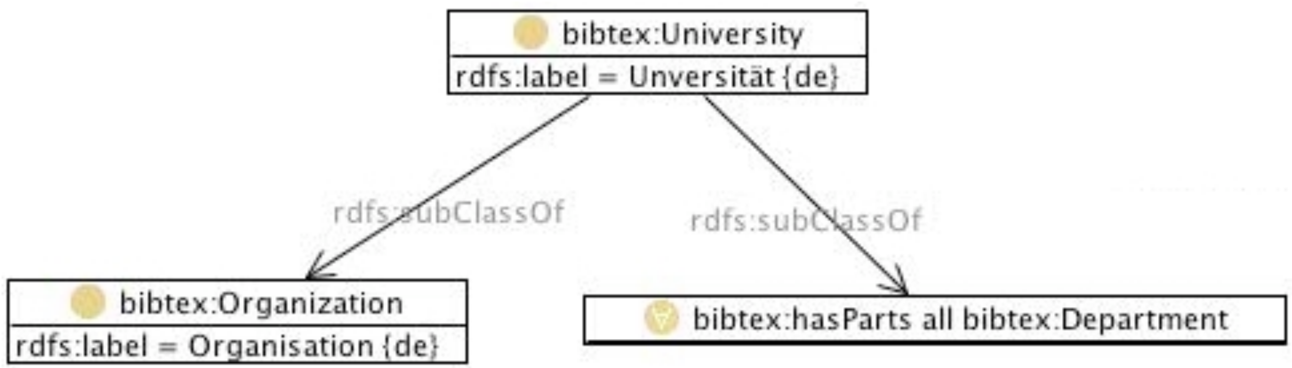
Kinds of ontology design patterns



Logical patterns (LPs). Definition

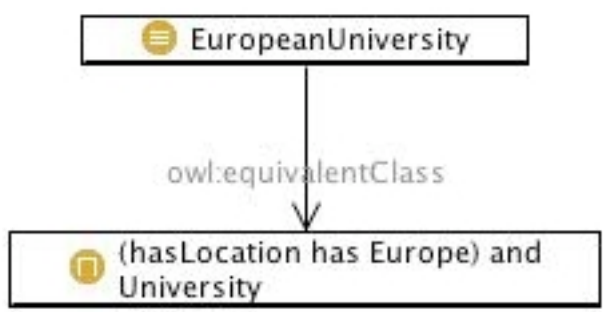
- Logical constructs or composition of them
- LPs are content-independent structures expressed only by means of a logical vocabulary (plus possible primitives, e.g. “owl:Thing”)
- They can be applied more than once in the same ontology in order to solve similar modeling problems
- Logical patterns presented here are specific to OWL (DL)

Some LPs: *Subsumption Macros*



subsumption by class: *bibtex:University instances are also bibtex:Organization instances*

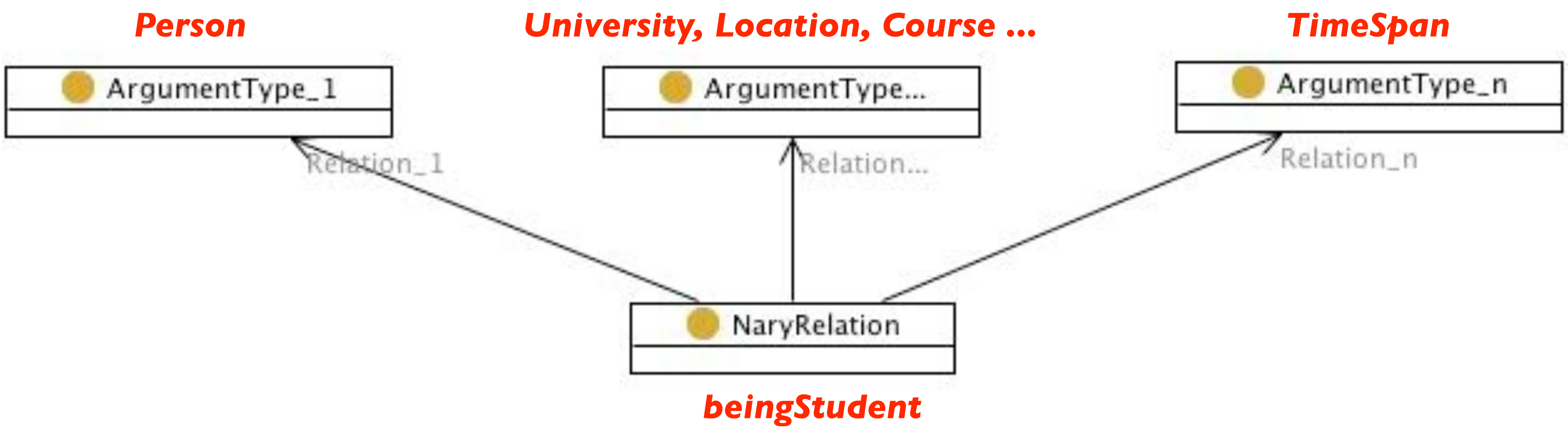
subsumption by restriction: *bibtex:University instances can only have bibtex:Department instances as Parts (!)*



equivalence by intersection: *European universities are universities that are located in Europe*

Some LPs: *N-ary relation*

- How to represent a relation with n arguments



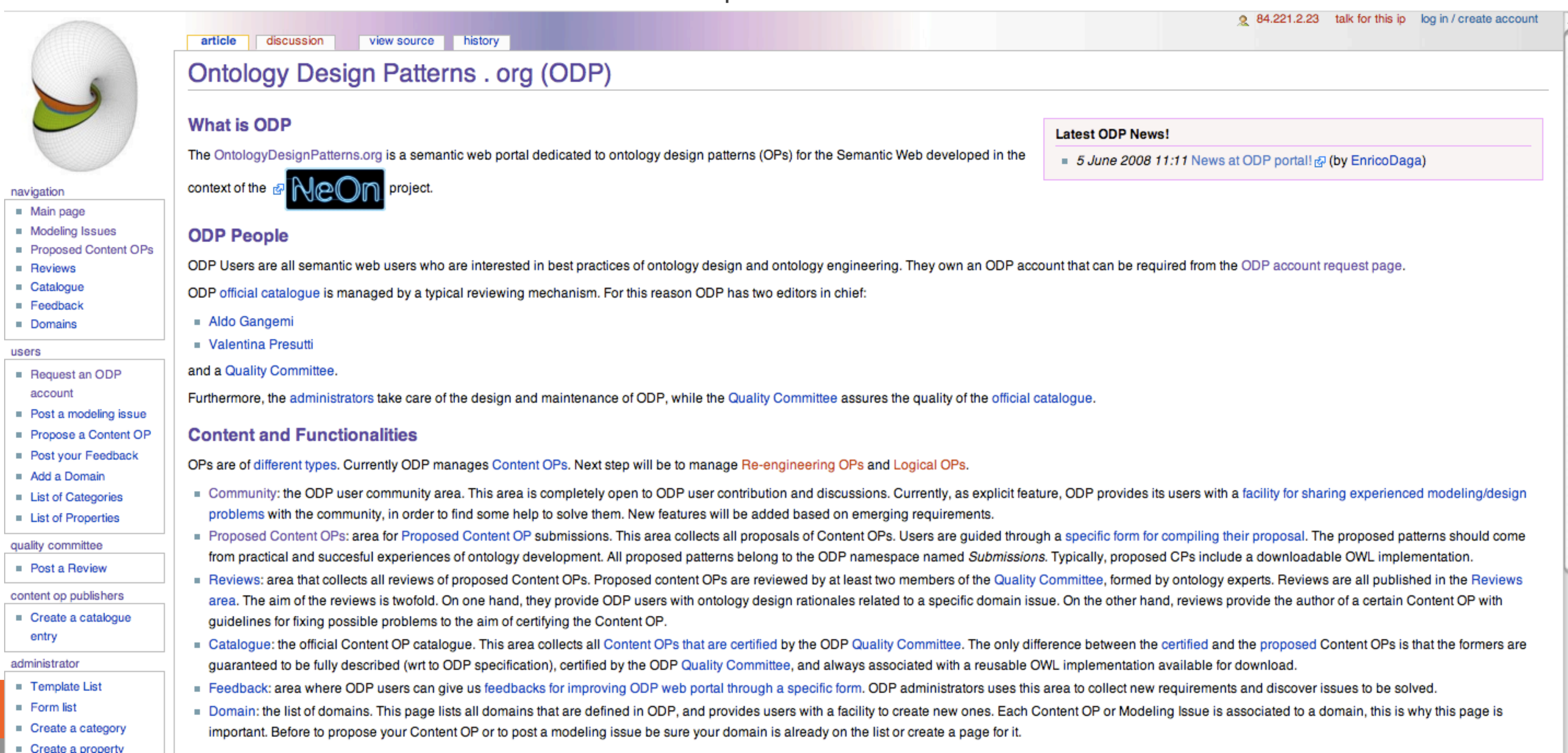
- Cf. W3C SWBPD, logical reification, DLR, UML association class

Content Patterns (CPs): Definition

- Instances of LPs or of compositions of LPs.
- Domain-dependent
  - Expressed with a domain specific (non-logical) vocabulary
- Solve domain modelling problems (expressible as tasks or “competency questions”)
- Affect the specific part of the ontology dealing with the related domain modelling problem
- Examples:
  - *PartOf, Participation, Plan, Medical Guideline, Sales Order, Research Topic, Legal Contract, Inflammation, Situation, TimeInterval*, etc.

The ODP portal

- 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



84.221.2.23 talk for this ip log in / create account

article discussion view source history

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.

navigation

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

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

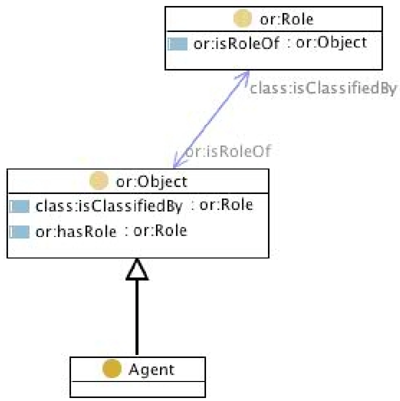
content op publishers

- Create a catalogue entry

administrator

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

Example 1: Agent Role



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.

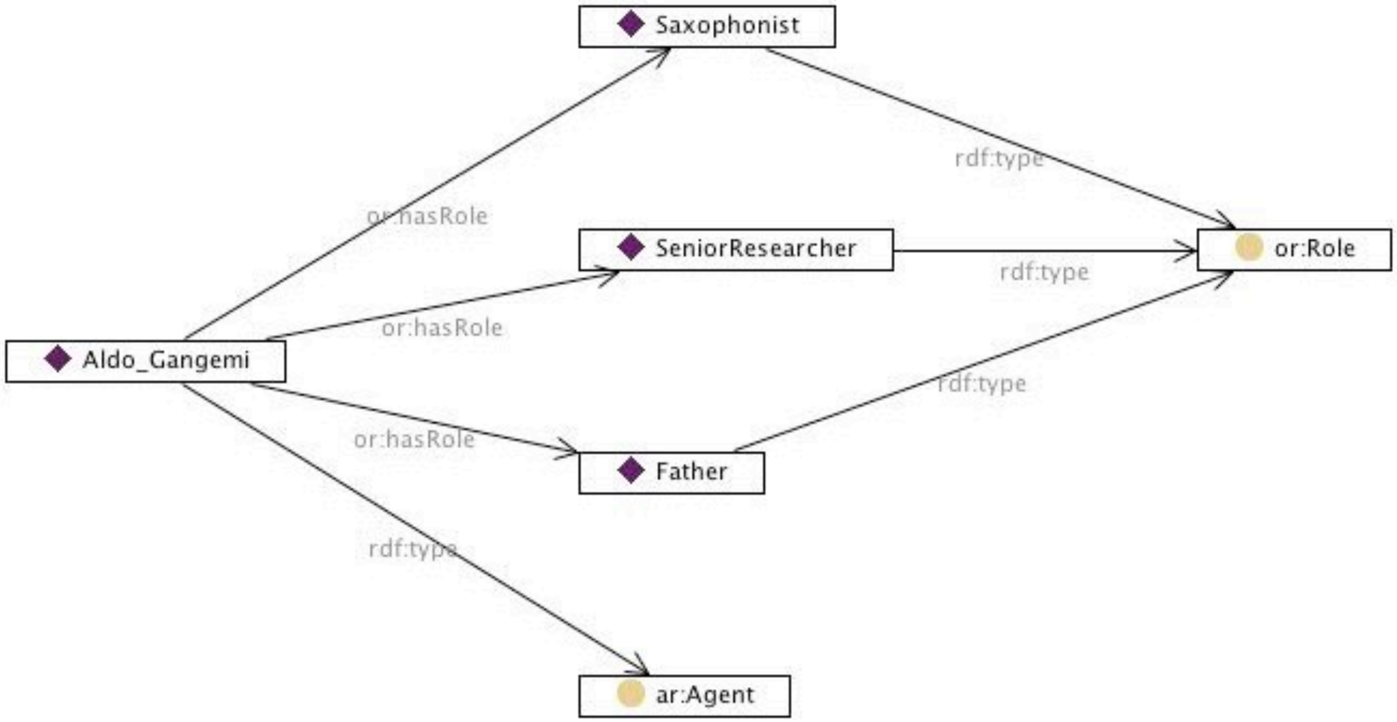
The elements of this Content OP are added with the elements of its components and/or the elements of the Content OPs it is a specialization of.

AgentRole

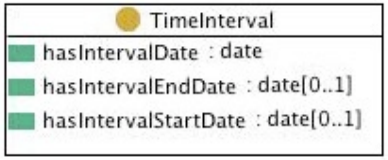
Submitted by	ValentinaPresutti
Name	agent role
Also Known As	
Intent	To represent agents and the roles they play.
Domains	Management, Organization, Scheduling
Competency Questions	which agent does play this role?, what is the role that played by that agent?
Reusable OWL Building Block	http://www.ontologydesignpatterns.org/cp/owl/agentrole.owl
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

- Scenario: Aldo Gangemi is a senior researcher. He is also father and saxophonist.



Example 2: Time Interval



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

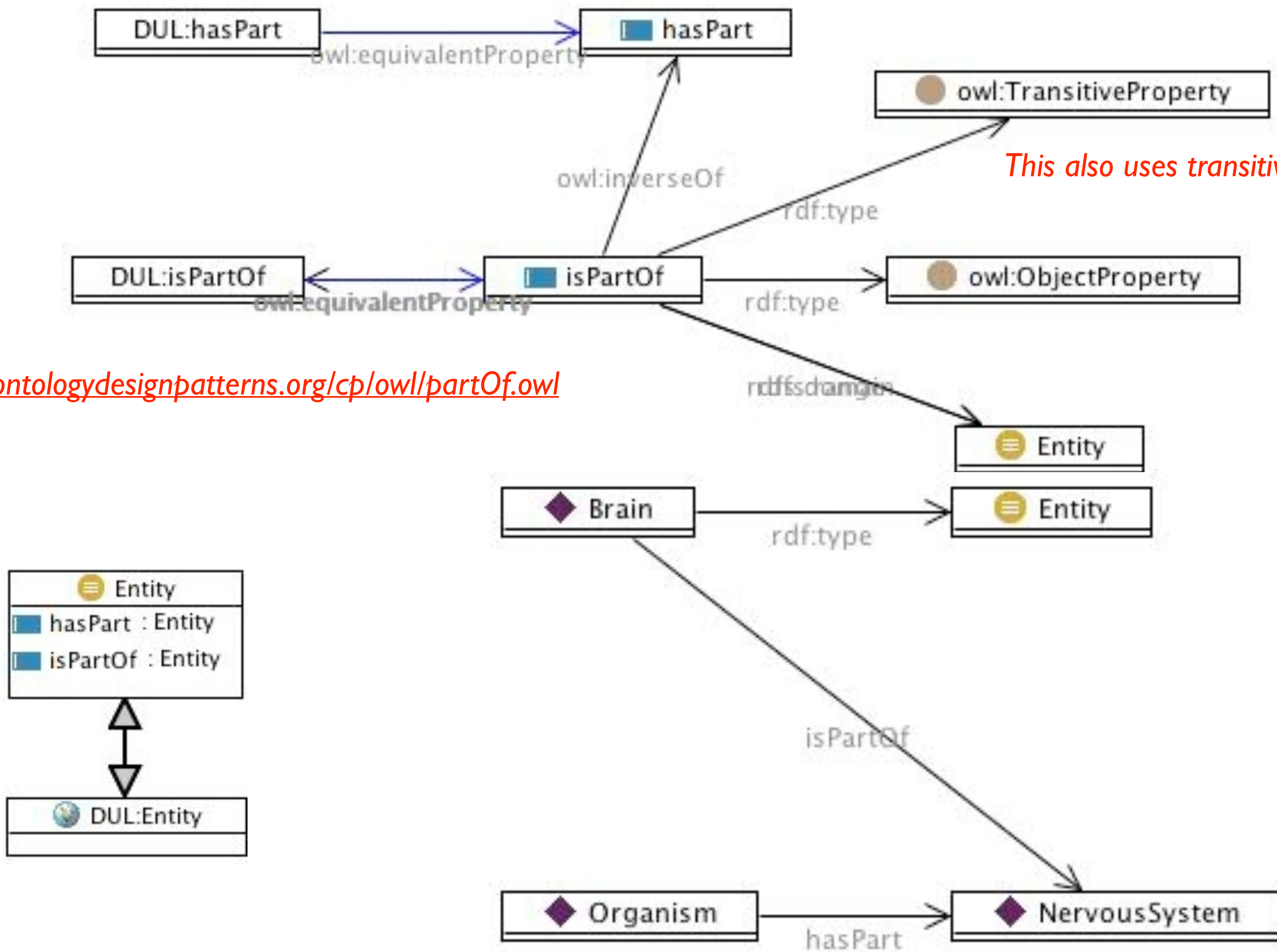
has interval end date (owl:DatatypeProperty)

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	

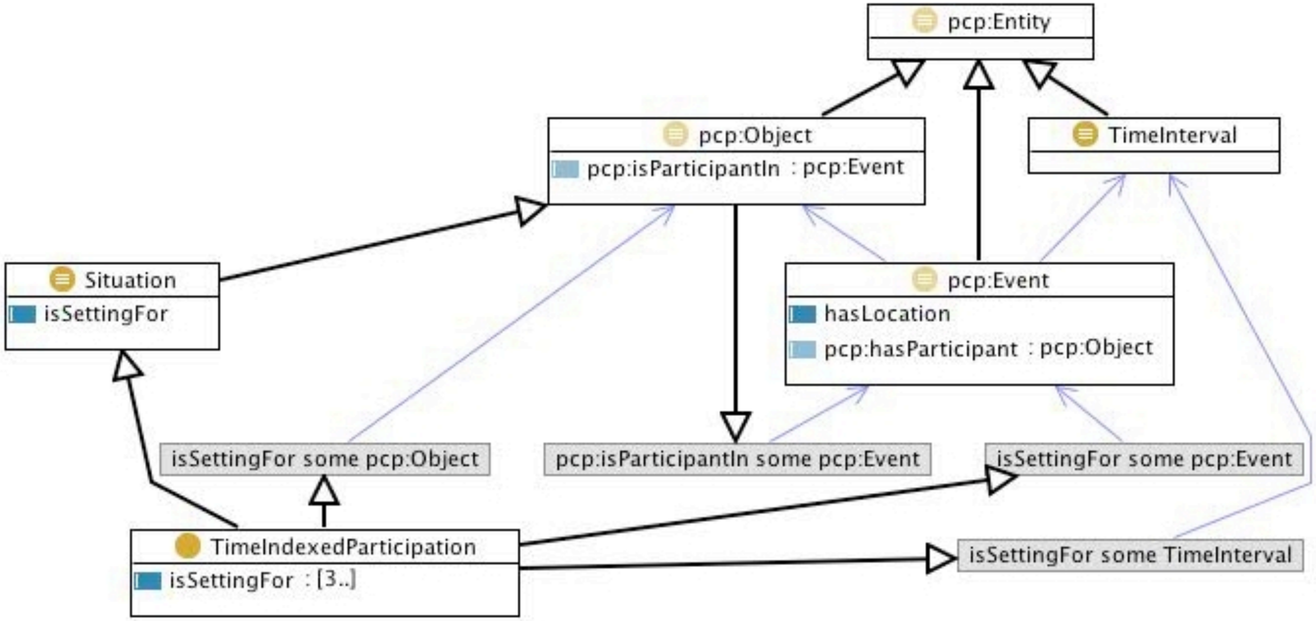
Example 3: PartOf



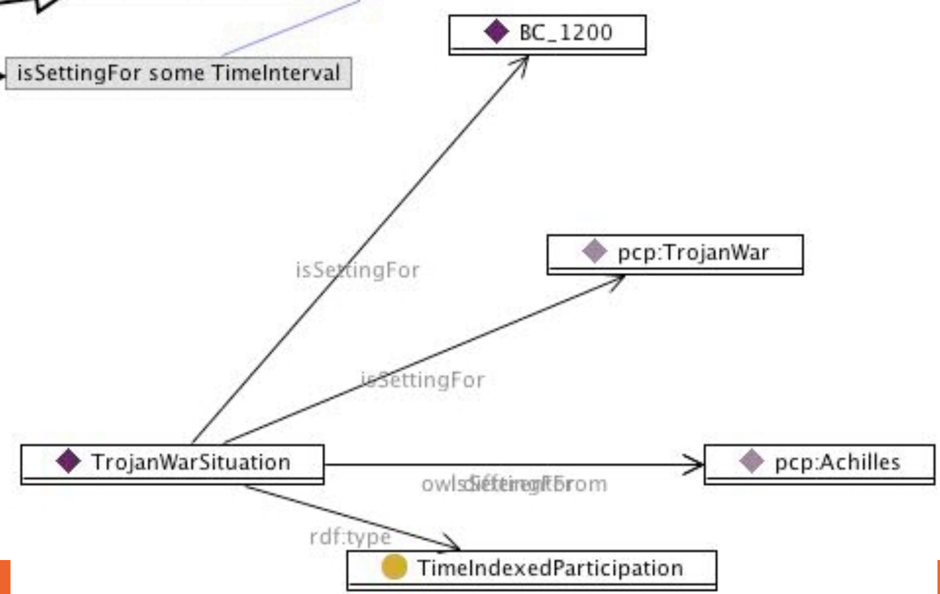
This also uses transitivity reasoning pattern

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

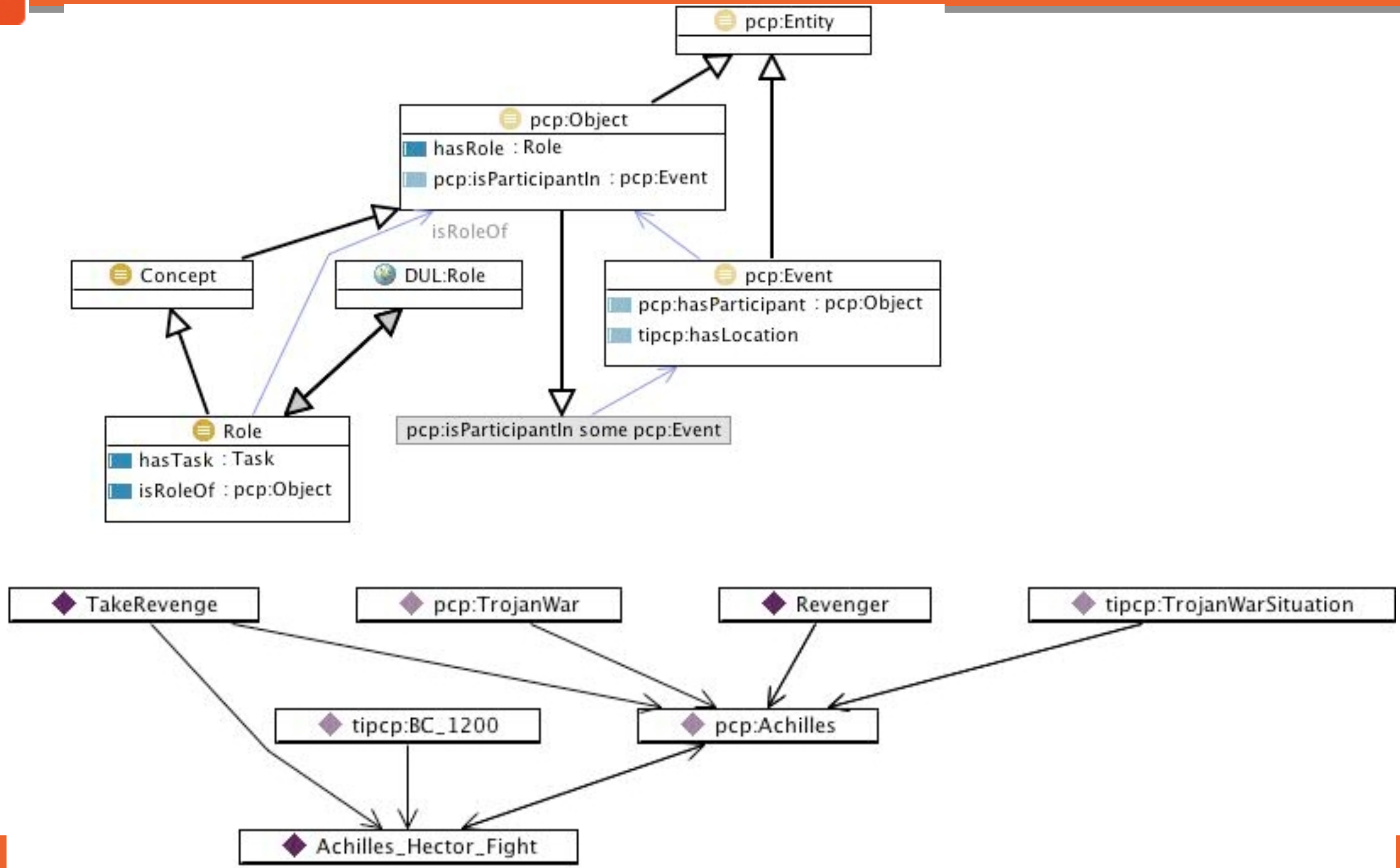
Example 4: Time-indexed Participation



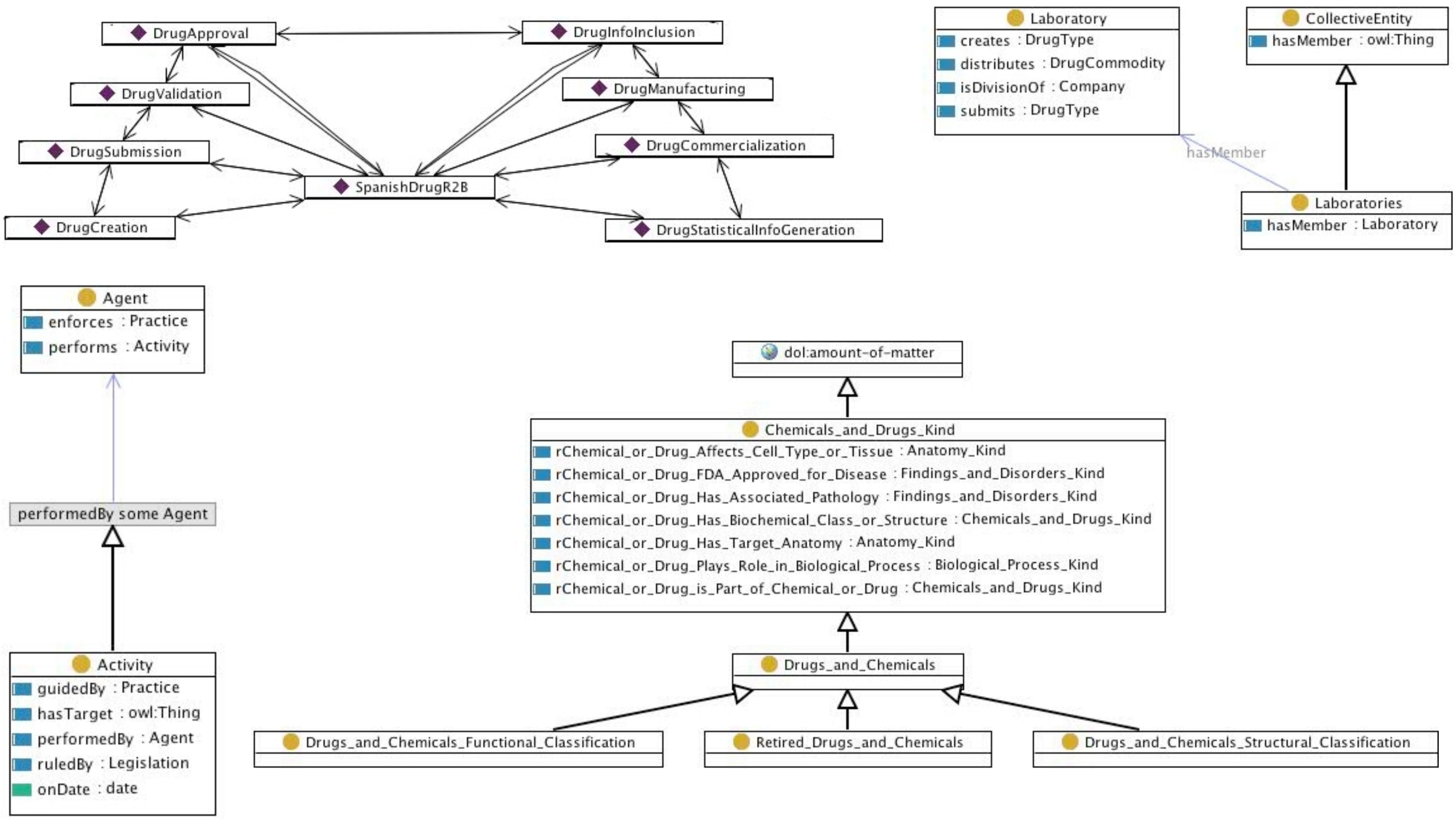
This also uses N-ary logical pattern



Example 5: Role-based Participation



Other applied CPs



Specializing patterns

- Same structure down the taxonomy hierarchy
- A CP p_2 specializes another p_1 when at least one of the classes or properties from p_2 is a sub-class or a sub-property of some class or property from p_1, while the remainder of the CP is identical.
- Participation (of an object in an event)
  - Taking part in a public enterprise activities
    - Funding a Semantic Web project
- Co-participation
  - Having a social relationship
    - Being bunkmates
- Renaming elements of an imported patterns is a bad practice
  - Specializing is the way of using CPs

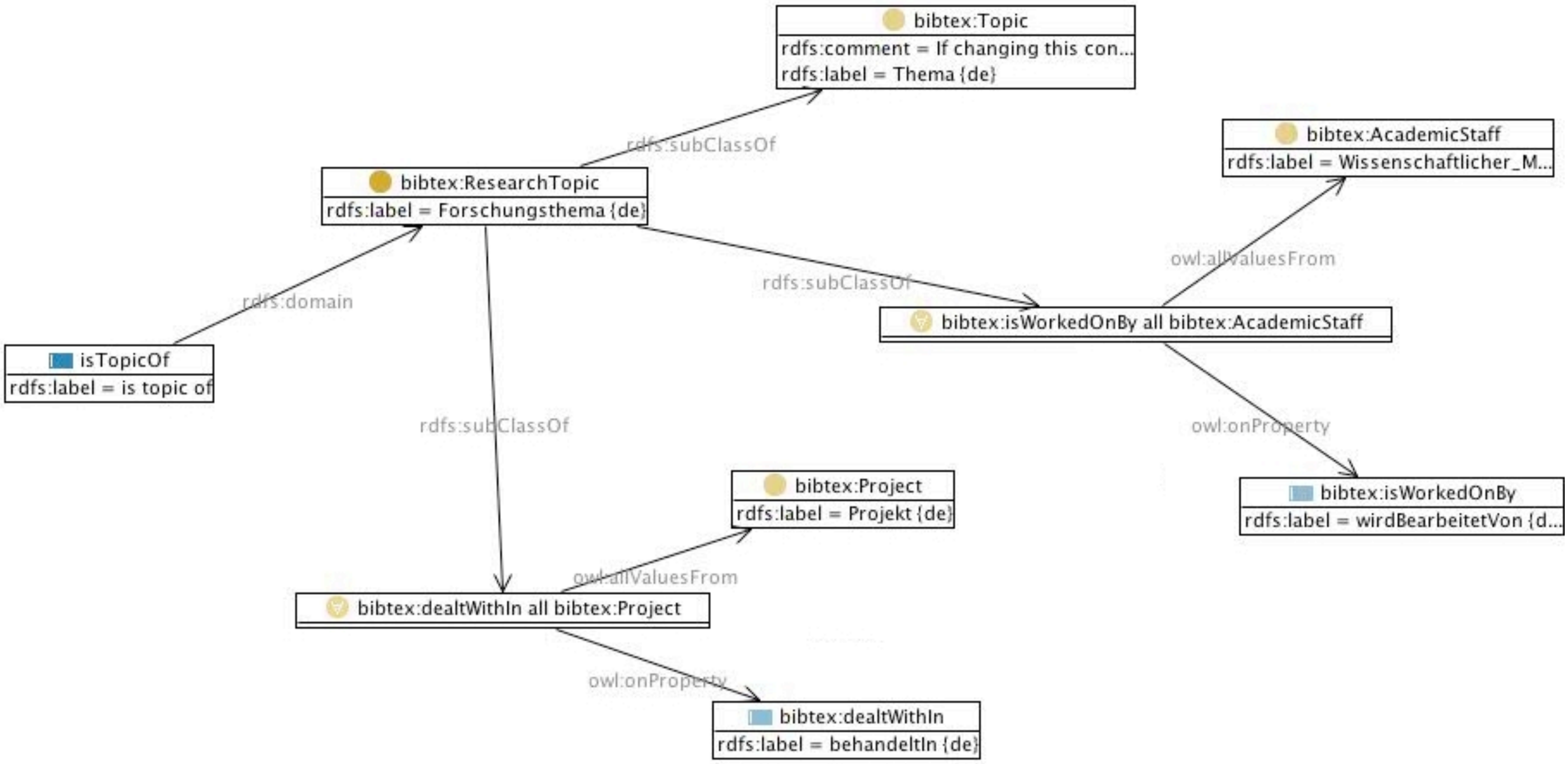
Composing patterns

- Linking sensible classes on the background of a common (or integrated) reference ontology
- A CP p_2 extends p_1 when p_2 contains p_1, while adding some other class, property, or axiom
- A CP p_3 integrates p_1 and p_2 when p_3 contains both p_1 and p_2
- A CP p_3 merges p_1 and p_2 when p_3 contains both p_1 and p_2, and there exist explicit links between at least two classes or properties from both p_1 and p_2
- *BiochemicalTreatment* → (*Role*⇔*Task* ° *Description*⇔*Situation* ° *Substance*⇔*Agent* ° *Time-indexedParticipation*)

A quick test: the SWC ontology

- Patterns used
  - Logical patterns
    - N-ary: as in *Product*
  - Content patterns
    - *Topic* pattern: obeys some tasks, generic coverage
  - Architectural patterns: *Alignment without import* to schemas used in applications: FOAF, SWRC, iCAL, WordNet1.6
  - Naming patterns

The “topic” content pattern as extracted from the SWRC ontology



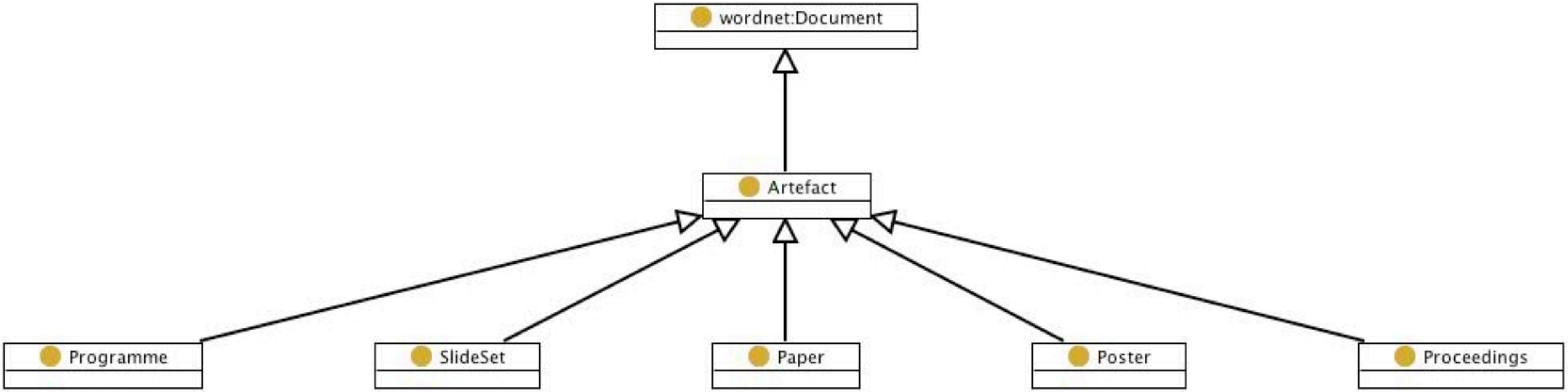
Design evaluation

- Coverage: *topics, staff, projects, dealt with by, worked on by, being a topic of*
- Task: *reasoning on semantic web entities*
- Does the *topic* pattern satisfy coverage and task requirements?

Best practice check

– *Check that names are intuitive*
  - Antipattern: using a generic name for a subclass of class that have a specific name:
    – *Artefact subClassOf wn:Document*

Counterintuitive naming



Task-based unit test 1

– *Finding what documents have a same topic*
  - Impossible: hasTopic not an inverse of isTopicOf (!),
  - Workaround: use SPARQL query
  - Also: Document class detached from the pattern
  - Minor problem for task, but implies design “sparseness”
  - Also: topics related to papers are instances of DBpedia:Topic, not from the list of individuals from swrc:ResearchTopic
  - Fix: equivalence axiom between swrc:ResearchTopic and DBpedia:topic

Task-based unit test 2

- *Checking that only events can be sub-events* ("atEvent") *of other events* (universal restriction)
  - Impossible: Event is not disjoint from e.g. Document
  - Consequence: e.g. a document that is said "atEvent" of an event, will be an event as well

Task-based unit test 3

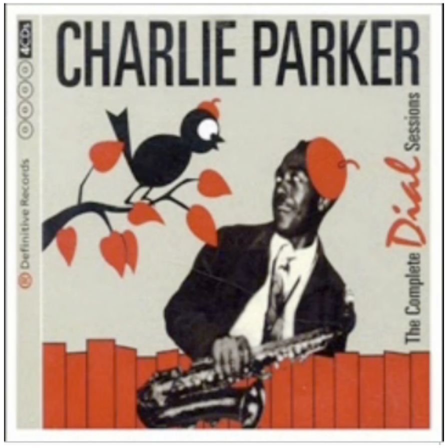
– *Finding all parts of the proceedings*

- Impossible: swc:hasPart and swc:isPartOf are not Transitive (and not Inverses)
- Consequence: e.g. a paper that is part of a section of the proceedings will not be part of the proceedings; a laboratory that is part of a department of a university will not be part of the university; that department will not be asserted to have the laboratory as part
- Also: no relation between transitive part for events (swc:subEvent), and the generic hasPart
- Fix: apply *partOf* patterns (e.g. SWBPD, DOLCE-Ultralite patterns), with *Transitive Reduction* pattern: transitive version of a property should be the more generic

Sample eXtreme Design iteration

- Sentence: *Charlie Parker is the alto sax player on Lover Man, Dial, 1946*
  - Charlie Parker (person)
  - the alto sax player (player role)
  - Lover Man (tune)
  - Dial (publisher)
  - 1946 (recording year)
- Competency Questions
  - what persons do play a musical instrument?
  - on what tune?
  - for what publisher?
  - in what recording year?

Alternative abstractions do exist!



- 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 }

cont.d

- 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), ...
- 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

Appendix:

Other types of ontology design patterns

Reasoning Patterns (RPs): Definition

- Application of LPs oriented to obtain certain inferencing results, based on the behavior implemented in a reasoning engine
- They are inference schemas, depending on the inference rules defined for a language
- Examples: Classification, Subsumption, Inheritance, Materialization, Query result construction

Classification and Subsumption RPs

- Automatic classification
  - Yes-Man(x) =df Man(x) $\wedge$ $\exists$y(hasFiancee(x,y))
  - Man(John)
  - hasFiancee(John,Mary)
  - $\therefore$ Yes-Man(John)
- Automatic subsumption
  - Yes-Man(x) =df Man(x) $\wedge$ $\exists$y(hasFiancee(x,y))
  - ItalianMan(x) $\Rightarrow$ Man(x)
  - hasFrenchFiancee(x,y) $\Rightarrow$ hasFiancee(x,y)
  - $\therefore$ ((ItalianMan(x) $\wedge$ $\exists$y(hasFrenchFiancee(x,y)) $\Rightarrow$ Yes-Man(x))

Inheritance and Materialization RPs

- Inheritance
  - $Man(x) \Rightarrow Human(x)$
  - $Yes\text{-}Man(x) \Rightarrow Man(x)$
  - $\therefore (Yes\text{-}Man(x) \Rightarrow Human(x))$
- Materialization
  - $hasFiancee(x,y) \Leftrightarrow hasFiance(y,x)$
  - hasFiancee(John,Mary)
  - $\therefore$ hasFiance(Mary,John)

Architectural Patterns (APs): Definition

- Equivalent to LPs (or compositions of them) that are used exclusively in the design of an ontology
- An AP is a content-independent structure
- It is supposed to characterize the overall structure of an ontology
- An AP dictates how the ontology should look like

Examples of APs

- Taxonomy
  - A hierarchical structure of classes only related by subsumption relations.
- Lightweight ontology. Taxonomy + other features, e.g.:
  - A class can be related to other classes through the disjointWith relation.
  - Object and datatype properties can be defined and used to relate classes.
  - A specific domain and range can be associated with defined object and datatype properties.
- Modular architecture
  - Structuring an ontology as a configuration of components, each having its own identity based on some design criteria
  - When an ontology is committed to a huge domain of knowledge, a good practice is to decompose the domain into smaller subdomains which address simpler tasks
  - Each subdomain can be then encoded in an ontology module, in order to provide the whole ontology with a modular architecture.

Re-engineering Patterns (RPs): Definition

- Transformation rules to be applied in order to map elements of a source model (i.e. knowledge resource) to elements of a target model.
- The target model is an ontology, while the source model can be either an ontology, a thesaurus, a DB schema, a UML model, etc.

Ontology-related data: knowledge resource types

- Modeling Languages
  - E/R, UML, XSD, Petri Nets, ebXML, BPEL4WS
- Conceptual models
  - Database schemas, UML diagrams, XSD schemas, etc.
- Informal Data Structures
  - Spreadsheets, tables, etc.
- Lexical resources
  - WordNet, FrameNet, Oxford Dictionary, etc.
- Concept Schemes
  - Thesauri, classifications, nomenclatures, etc.
- Open tag systems
  - Flickr, Wikipedia, MySpace, ...
- Linked Open Data
  - DBpedia, Microformats, RDFa, etc.
- Text extractors
  - Text2Onto, TermExtractor, SST, Frame Detector, ...

Searching and using ontologies, on-the-fly data reengineering

- Watson (RDF search engine)
- Sindice (RDF search engine)
- Yago (a metamodel for dbpedia and wordnet)
- Umbel (a topic ontology for Linked Open Data)
- LMM (a semiotic metamodel for Linked Open Data and lexical resources)
- Freebase (a metamodel and user interface to enriched Linked Open Data)
- OpenLink Data Explorer (a user interface over Linked Open Data)
- GRDDL, RDFa (RDFizers over Web pages and Microformats)

Example of RP: from thesauri to ontologies in SKOS

- KOS ⇒ skos:ConceptSchema
- Descriptor ⇒ skos:Concept
- Broader Term ⇒ skos:broader
- Related Term ⇒ skos:related

Summary

- Interdisciplinary character of ontology design
- Ontology design and ontology evaluation
- Problem space vs. Solution space
  - The issue of matching problems to solutions
- Ontology design patterns
  - Ontology building blocks
  - Allow design by re-engineering, specialization and composition
  - Support ontology evaluation

Contribute to the collaborative design effort!

- http://www.ontologydesignpatterns.org
- http://www.neon-project.org
- http://www.w3.org/2001/sw/BestPractices/

Some references

Alexander, C.: The Timeless way of building. Oxford University Press, New York (1979).

Catenacci, C., A. Gangemi, J. Lehmann, M. Nissim, V. Presutti, G. Steve, N. Guarino, C. Masolo, H. Lewen, K. Dellschaft, and M. Sabou. NeOn Deliverable D2.1.1 Design rationales for collaborative development of networked ontologies - State of the art and the Collaborative Ontology Design Ontology. February 2007. Available at: http://www.neon-project.org.

Clark, P., Thompson, J., Porter, B.: Knowledge Patterns. KR2000 (2000).

Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, Reading, MA (1995).

Gangemi, A., Catenacci, C., Battaglia, M. Inflammation Ontology Design Pattern: an Exercise in Building a Core Biomedical Ontology with Descriptions and Situations", in Pisanelli D. (ed.), Biomedical Ontologies, IOS Press, Amsterdam, 2004.

Gangemi, A. Ontology Design Patterns for Semantic Web Content. Musen et al. (eds.): Proceedings of the Fourth International Semantic Web Conference, Galway, Ireland, 2005. Springer.

Gangemi, A, C. Catenacci, M. Ciaramita, J. Lehmann. Modelling Ontology Evaluation and Validation. Proceedings of ESWC 2006.

Gangemi, A., V. Presutti. Ontology Design for Interaction in a Reasonable Enterprise. Staab et al. (eds.): Handbook of Ontologies for Business Interaction, 2007. IGI Global.

Gangemi, A., V. Presutti. Ontology Design Patterns. Staab et al. (eds.): Handbook of Ontologies (2nd Edition), to appear. Springer.

Gruninger, M., and Fox, M.S.: The Role of Competency Questions in Enterprise Engineering. Proceedings of the IFIP WG5.7 Workshop on Benchmarking - Theory and Practice, Trondheim, Norway (1994).

Guizzardi, G., Wagner, G., Guarino, N., van Sinderen, M.: An Ontologically Well-Founded Profile for UML Conceptual Models. A. Persson, J. Stirna (eds.) Advanced Information Systems Engineering, Proceedings of16th CAiSE Conference, Riga, Springer (2004).

Haase, P, S. Rudolph, Y. Wang, S. Brockmans, R. Palma, and J. Euzenat, M. d'Aquin. NeOn Deliverable D1.1.1 Networked Ontology Model. November 2006. Available at: http://www.neon-project.org.

Masolo, C., A. Gangemi, N. Guarino, A. Oltramari and L. Schneider: WonderWeb Deliverable D18: The WonderWeb Library of Foundational Ontologies (2004).

Masolo, C., L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi and N. Guarino: Social Roles and their Descriptions. Procedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning, Whistler (2004).

Noy, N.: Representing Classes As Property Values on the Semantic Web. W3C Note, http://www.w3.org/2001/sw/BestPractices/OEP/ClassesAsValues-20050405/ (2005).

Noy, N, A. Rector. Defining N-ary Relations on the Semantic Web. W3C Working Group Note. 2006.

Pan, JF, L. Lancieri, D. Maynard, F. Gandon, R. Cuel, and A. Leger. Knowledge Web Deliverable D1.4.2.v2. Success Stories and Best Practices. January 2007. Available at: http://www.csd.abdn.ac.uk/~jpan/pub/TR/D142v2-final.pdf.

Pinto, S, S. Staab, C. Tempich. DILIGENT: Towards a Fine-Grained Methodology towards Distributed, Loosely-Controlled and Evolving Engineering of Ontologies. ECAI 2004.

Presutti V., Gangemi, A. Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies. Proceedings of ER2008.

Rector, A.L., Rogers, J.:Patterns, Properties and Minimizing Commitment: Reconstruction of the GALEN Upper Ontology in OWL, in (Gangemi and Borgo 2004) (2004).

Sabou, M, V. Lopez, E. Motta. Ontology Selection on the Real Semantic Web: How to Cover the Queens Birthday Dinner? In Proceedings of the European Knowledge Acquisition Workshop (EKAW), Podebrady, Czech Republic (2006).

Shum, SB, E. Motta, and J. Domingue. Augmenting Design Deliberation with Compendium: The Case of Collaborative Ontology Design. Position paper at the Workshop on Facilitating Hypertext Collaborative Modelling in conjunction with ACM Hypertext Conference, Maryland, June 11-12, 2002.

Svatek V.: Design Patterns for Semantic Web Ontologies: Motivation and Discussion. In: 7th Conference on Business Information Systems, Poznan (2004).

Welty, C.: Semantic Web Best Practices and Deployment Working Group, Task Force on Ontology Engineering Patterns. Description of work, archives, W3C Notes and recommendations available from http://www.w3.org/2001/sw/BestPractices/OEP/ (2004-5).