Best practices in ontology design

Tutor: Aldo Gangemi
Assistant tutor: Valentina Presutti

Lecture @ SSSW 2009
Cercedilla, Spain

Tutor info

- Name: Aldo Gangemi
- Institute: ISTC-CNR, Rome, Italy
- Research Group: Semantic Technology Laboratory (STLab)
- Roles: Senior Researcher, Head of STLab
- Research interest: ontology design, KE+NLP, ...

Outline

- Ontology design
- Knowledge resources and OWL
- Pattern-based ontology design
- Pattern types

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

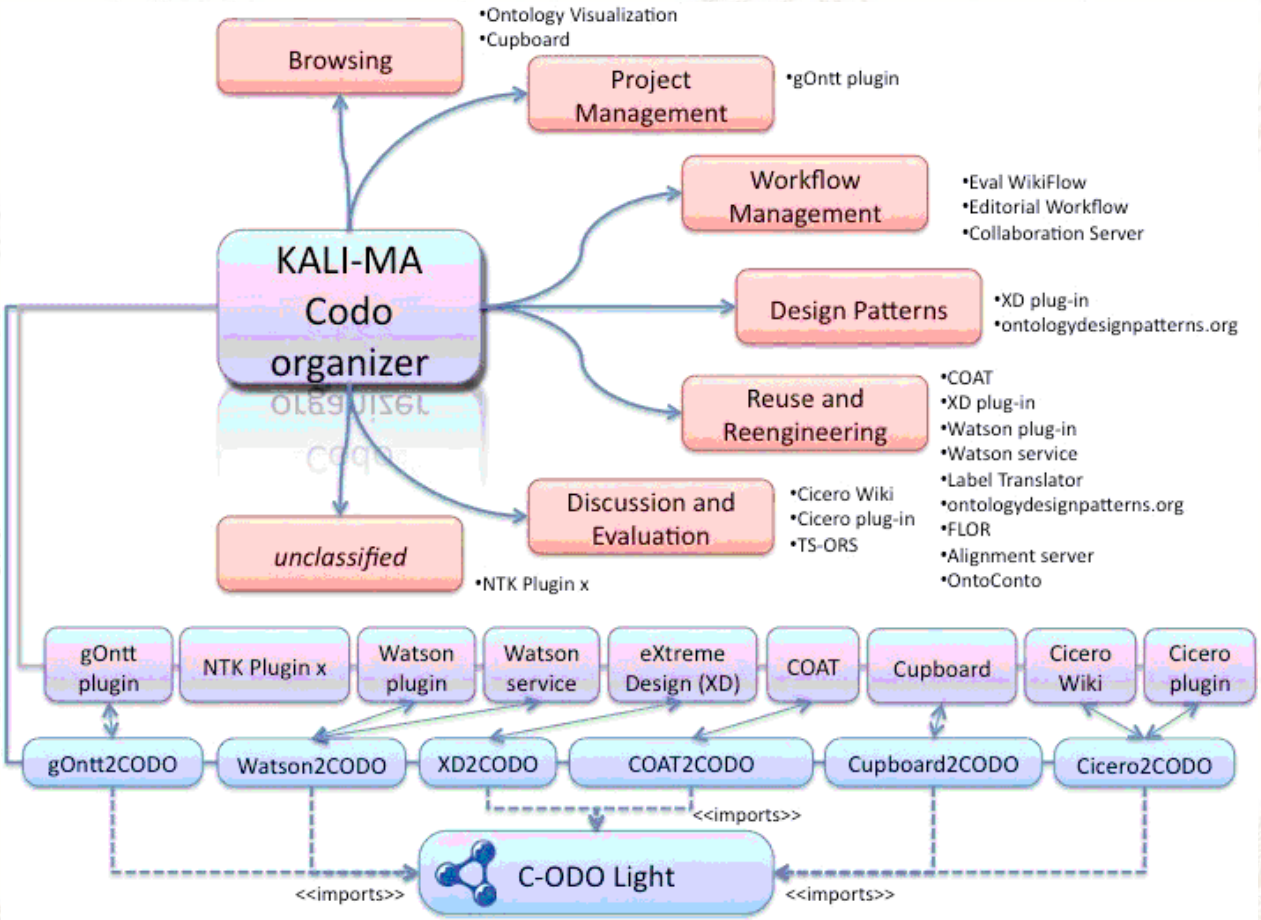
What is ontology design? 2/3

- 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 aspects in *Kali-mashup*



Two kinds of ontologies

- Coverage-oriented ontologies
  - They cover the terminology/metadata/textual corpora/ folksonomies ... that fit a specific domain [*big reengineering problem* - exploited for annotation, retrieval, etc.]
- Task-oriented ontologies
  - They are able to give a structure to a knowledge base that can be used to answer competency questions [*big design and reuse problem* - exploited for automated reasoning and querying]
- Currently
  - A mass of heterogeneous data and ontologies, either expressed or portable to RDF (DB lifting, rdf-ized sources, etc.)
  - with generally low quality in some quality dimension/aspect

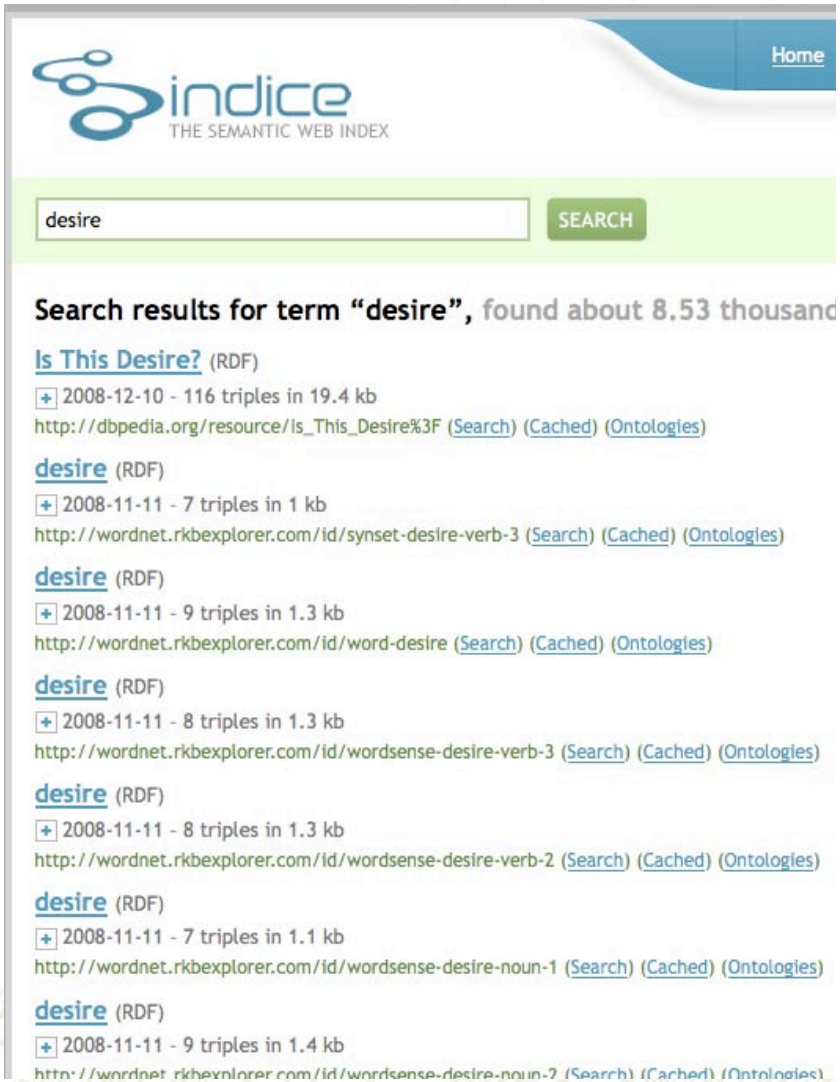
Ontology-related data in the web suq (“knowledge resources”)

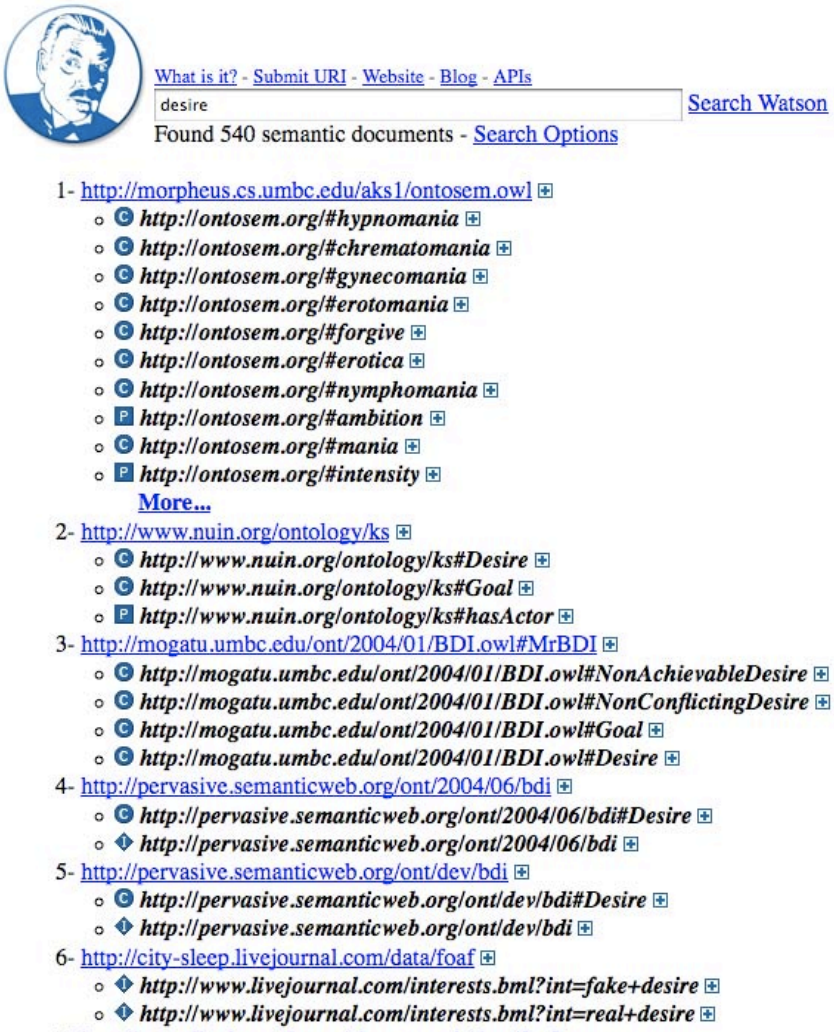
- Informal through formal
- Reengineering-oriented
  - Text corpora
  - Lexica (dictionaries, wordnets, terminologies, nomenclatures)
  - Knowledge organization systems (thesauri, classification schemes)
  - Folksonomies (tag sets, directories, topic trees, subject indexes)
  - Frames, semantic networks
  - Microformats, infoboxes, HTML templates
  - DB schemas and records (RDBS, XML, proprietary or not)
- Reuse-oriented
  - LOD datasets
  - RDFa, eRDF
  - DC, FOAF, SIOC, SKOS, etc. (RDFS vocabularies)
  - OWL ontologies

Standard languages help

- Transform all in RDF, or even OWL
  - Cf. *Triplify* initiative
  - Datasets extracted from heterogeneous sources, and triplified
  - Semantics depends on intended task of data and relations used for linking
- Then search/visualize RDF data, or make integrating applications

Knowledge search over the semantic web





Knowledge search over the LOD cloud

OPENLINK SOFTWARE Data Explorer

http://dbpedia.org/resource/Constitution

Query | Find

What | Where | When | Who | Images | Grid view | Tag Cloud | SVG Graph | Raw triples | Custom

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

Categories | Filters | Previous Queries | Bookmarks

There are 66 triples available.

Human readable

visible columns

#	Subject	Predicate	Object
1	Constitution	comment	En konstitusjon er et sett med r grunnleggende prinsipper for e konstitusjon er ofte kodifisert i stiftelsesdokument.
2	Constitution	comment	La costituzione di un'organizza struttura, attivit, carattere e r deriva dal latino constitutio, che legge di particolare importanza, dall'imperatore ed è tutt'ora usa indicare decisioni rilevanti prese costituzione apostolica che fissa il periodo di sede vacante e per conclave, del nuovo vescovo di comune del termine rimanda, se fondamentale di uno stato. Que ricco di significati, sia descrittiv
3	Constitution	comment	En forfatning eller grundlov læg

#	Subject	Predicate	Object
38	Constitution	subject	Category:Official_documents
39	Constitution	subject	Category:Constitutional_law
40	Constitution	subject	Category:Constitutions
41	Constitution	sameAs	guid.9202a8c04000641f800000000000cbee
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

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, owl:Class, owl:ObjectProperty, owl: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...

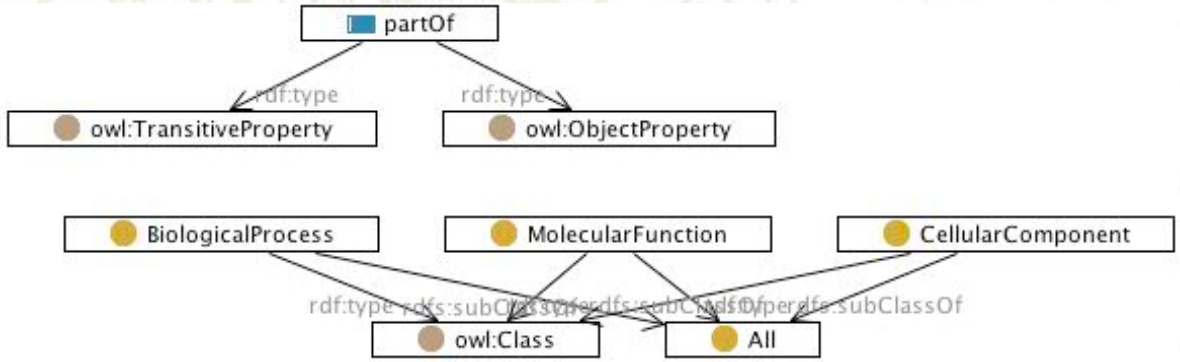
- cf. Semantic Web Interest Group post May 27th, 2008 by Zille Huma:

"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) “searching resorts is a type of functionality required for this kind of services”
  - owl:Class(searchResort) rdfs:subClassOf(Functionality)
- (2) “a functionality for searching resorts is implemented in our web service”
  - owl:Individual(searchResort) rdf:type(Functionality)
- (3) “who has been searching for what resorts in our web service?”
  - owl:ObjectProperty(searchResort) rdfs:domain(Customer) rdfs:range(Resort)
- (4) “how many users have been using our resort searching functionality?”
  - owl:DatatypeProperty(searchResort) rdfs:domain(Customer) 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 here 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

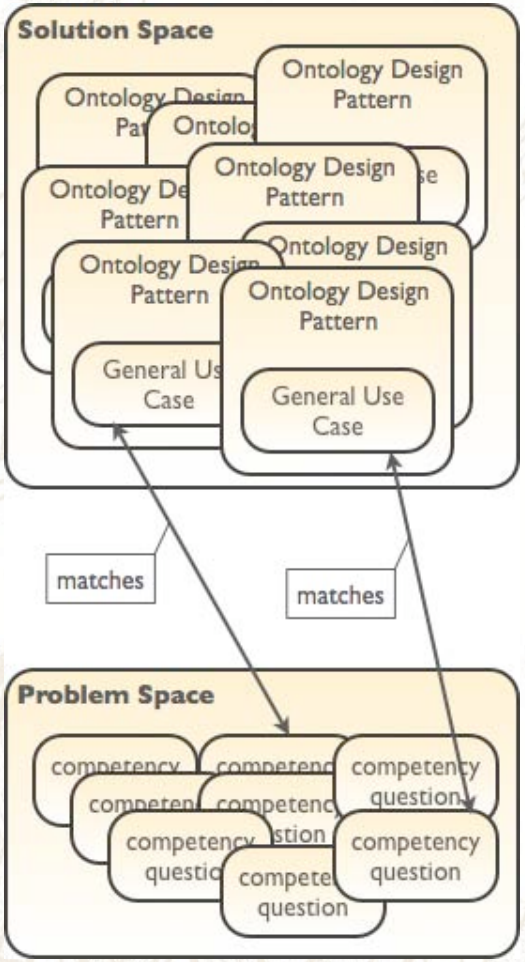
- Requirements (e.g. "*I want to attend my ideal talk*")
- Logical constructs (rdfs:subClassOf, owl:Restriction, ...)
- Existing ontologies (FOAF, BibTex, SWC, DOLCE, ...)
- Informal knowledge resources (CiteSeer, ACM topic catalog)
- Conventions and practices (e.g. naming, URI making, XML2OWL, SKOS,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

Ontology Design Pattern

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



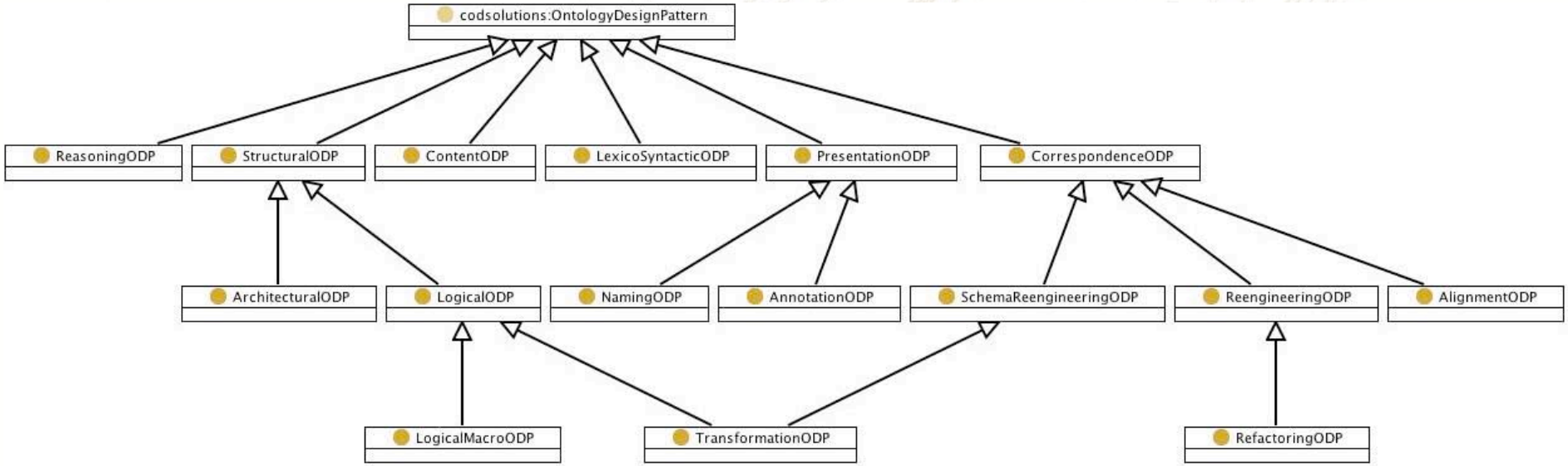
Pattern-based design aka *eXtreme Design* (XD)

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

eXtreme ontology Design (XD)

- Inspired by eXtreme Programming basic rules
  - e.g., pair programming, test-oriented, continued integration, etc.
  - but XP focuses on refactoring rather that on “design”
- 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
- An example will be presented and implemented in the afternoon

Types of Ontology Design Patterns (ODPs)

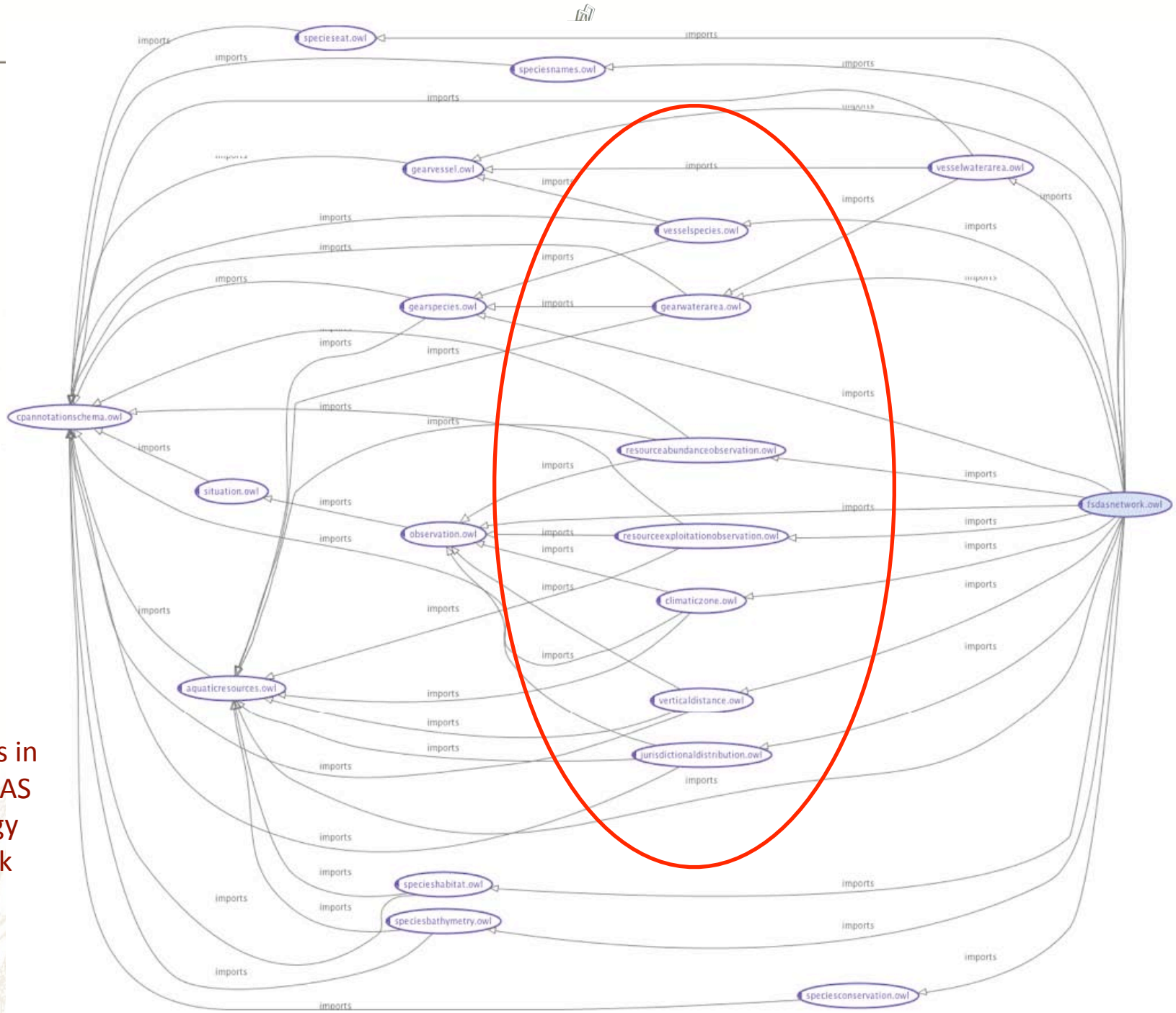


- We also distinguish between ontological resources that are not ODPs and Ontology Design Anti-Patterns (AntiODP)

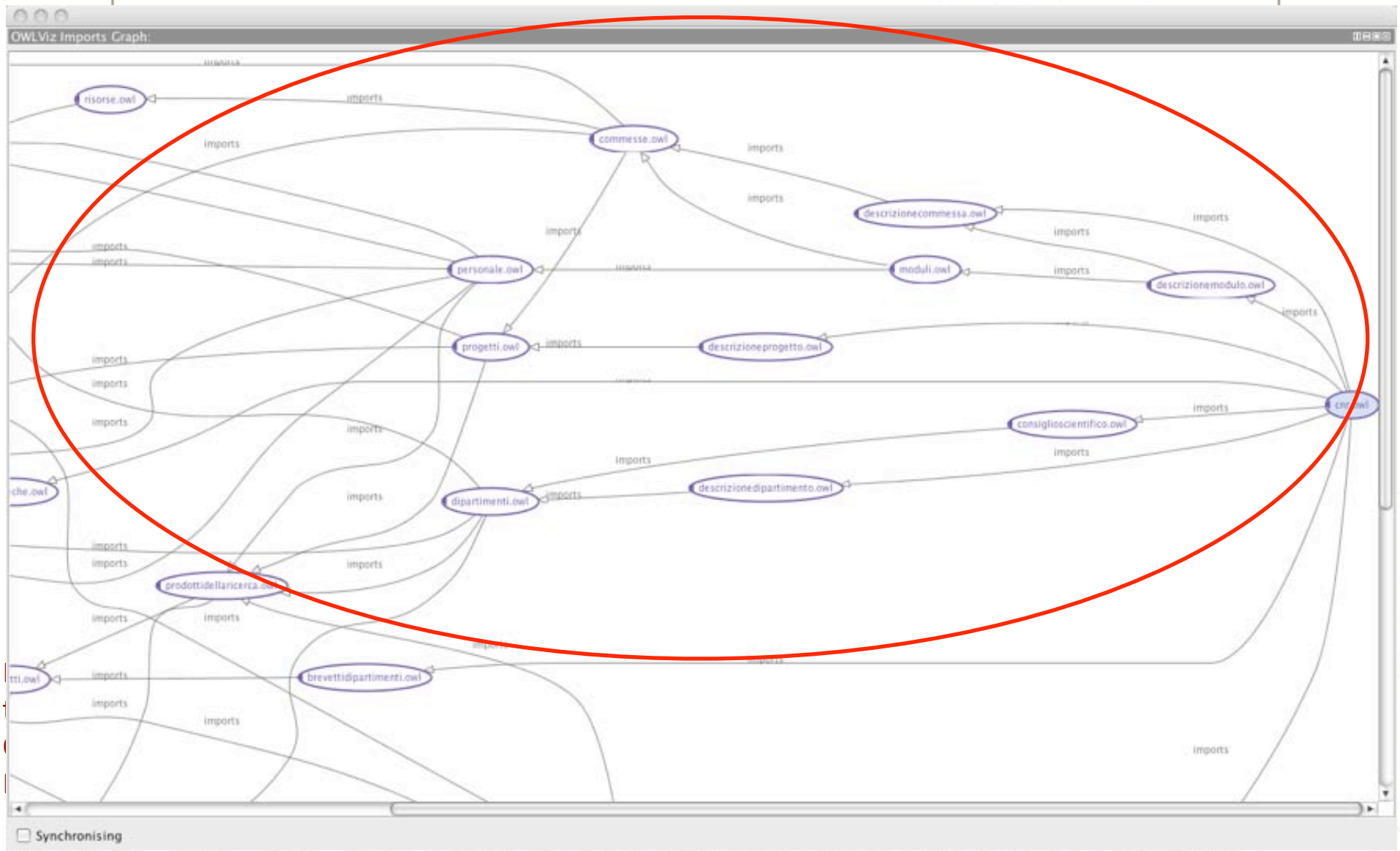
Architectural ODPs

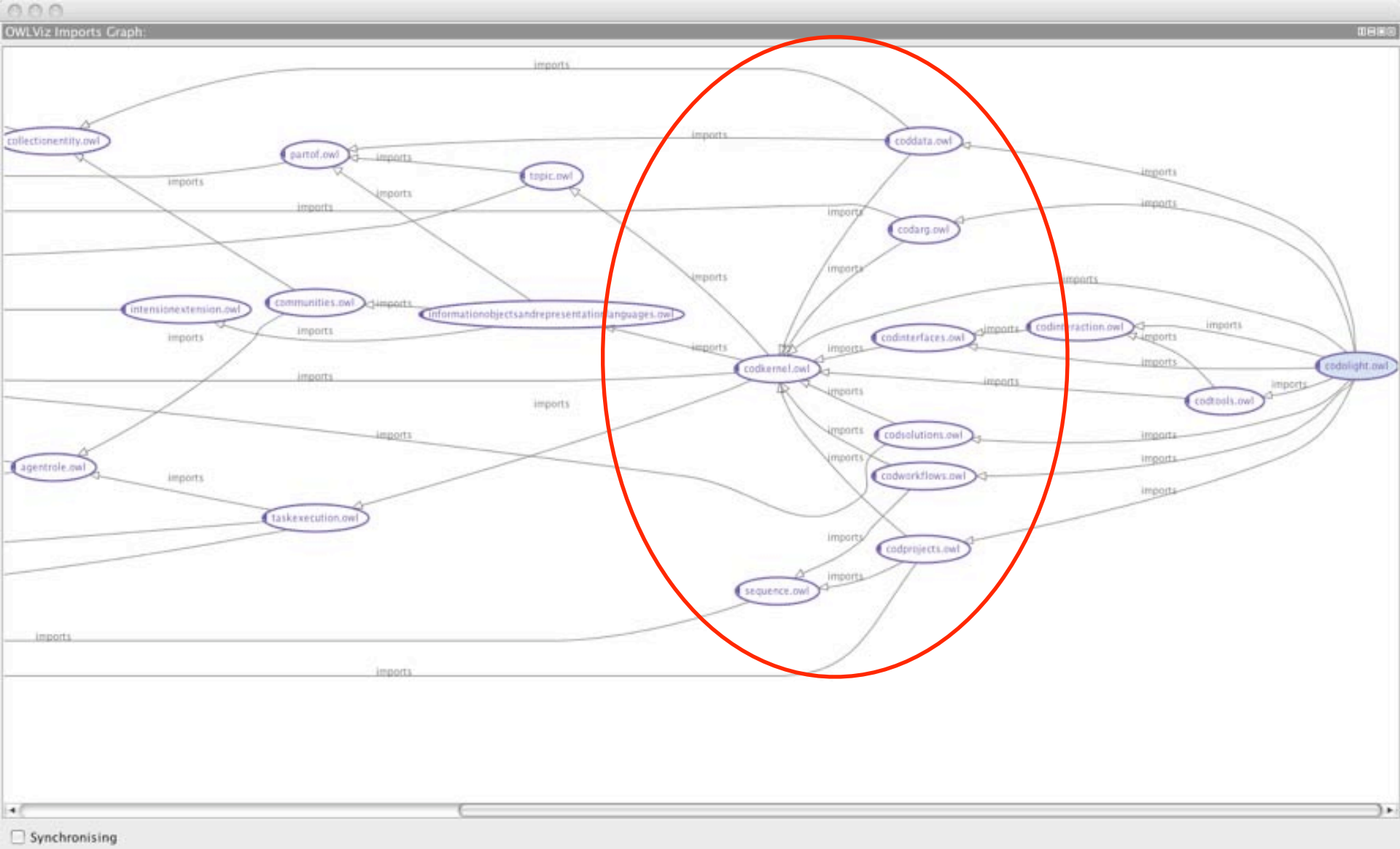
Definition

Architectural ODPs affect the overall shape of the ontology: their aim is to constrain 'how the ontology should look like'



Patterns in the FSDAS Ontology Network


OWLViz Imports Graph:
risorse.owl
commesse.owl
descrizionecommessa.owl
personale.owl
moduli.owl
descrizionemodulo.owl
progetti.owl
descrizioneprogetto.owl
consiglioscientifico.owl
dipartimenti.owl
descrizionedipartimento.owl
prodottidellaricerca.owl
brevettidipartimenti.owl
imports
Synchronising


OWLViz Imports Graph:
collectionentity.owl
partof.owl
topic.owl
coddata.owl
codarg.owl
intensionextension.owl
communities.owl
informationobjectsandrepresentationlanguages.owl
codinterfaces.owl
codinteraction.owl
codkernel.owl
codlight.owl
codtools.owl
codsolutions.owl
codworkflows.owl
agentrole.owl
taskexecution.owl
codprojects.owl
sequence.owl
imports
Synchronising

What criteria for architecture?

- Manageability, work distribution
  - how large the sections? how to partition?
  - weak evidence of good results when doing partitioning ex post
- Requirements ("competency questions")
  - cf. fisheries example, modelling issues for fsdas
- Broad topics
  - when no requirements available, or with large , lightweight ontologies

Logical ODPs

- cf. Sean's tutorial

Content ODPs

Content ODPs (CPs) 1/3

- CPs encode conceptual, rather than logical design patterns.
  - Logical ODPs 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

33

Content ODPs (CPs) 2/3

- CPs are instantiations of Logical ODPs (or of compositions of Logical ODPs), 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 ODPs (CPs) 3/3

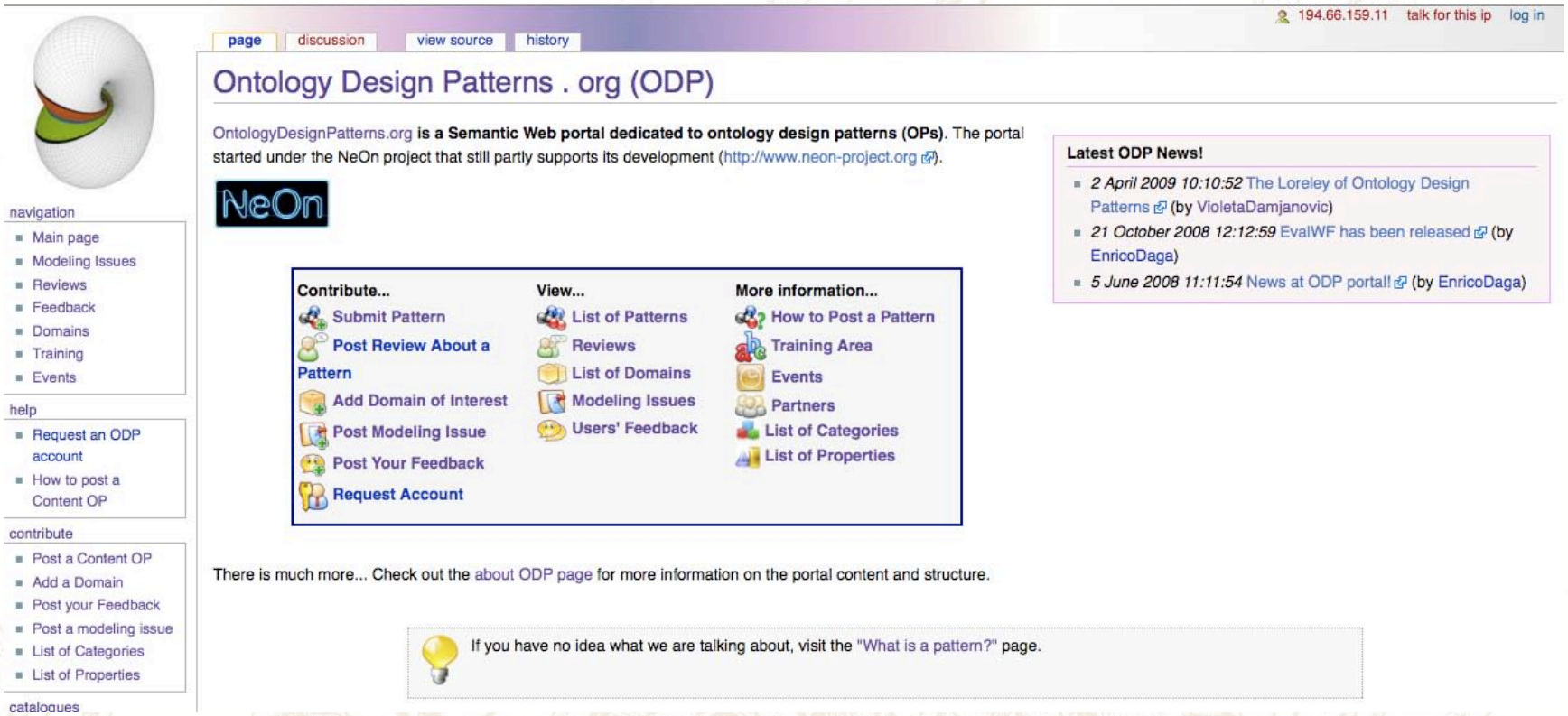
- Modeling problems solved by CPs have two components: domain and requirements.
  - A same domain can have many requirements (e.g. different scenarios in a clinical information context)
  - A same requirement can be found in different domains (e.g. different domains with a same “expert finding” scenario)
- A typical way of capturing requirements is by means of competency questions

35

Catalogues of CPs 1/2

- Content ODPs are collected and described in catalogues and comply to a common presentation template
- The ontologydesignpatterns.org initiative maintain a repository of CPs and a semantic wiki for their description, discussion, evaluation, certification, etc.

Catalogues of CPs 2/2



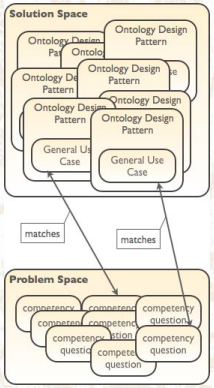
Presentation template of CPs on ontologydesignpatterns.org

Pragmatic characteristics of CPs

- Domain-dependent
  - Expressed with a domain-specific (non-logical) vocabulary
- Requirement-covering
  - Solve domain modeling 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, Sequence, Plan, Legal Norm, Legal Fact, Sales Order, Research Topic, Legal Contract, Inflammation, Medical Guideline, Gene Ontology Top, Situation, Time Interval, Tagging, Aquatic Resources ...

Generic ontology requirements

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



Imagine we have to model the following

- Arnold Schwarzenegger is Shylock in the performance of the play "Merchant of Venice”, that is given at the theatre “Globe” during July 2009

Analyze the sentence, detect the modeling issues, and match to the CPs

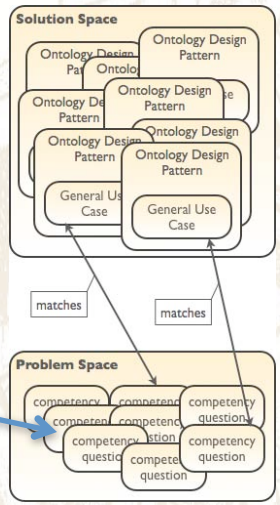
- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character

Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character

Analyze the sentence, detect the modeling issues, and match to the CPs

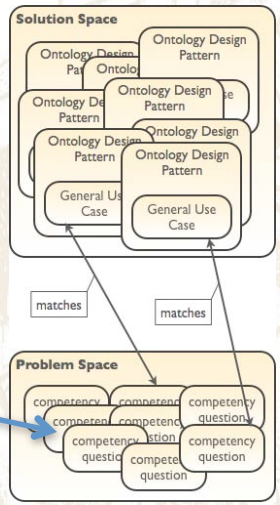
- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character



Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character

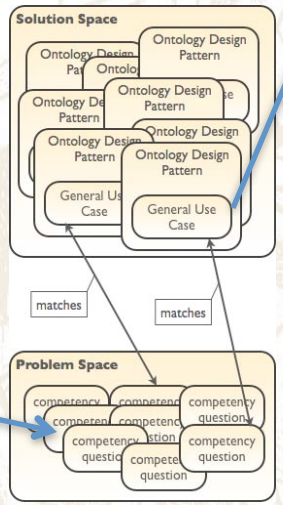
- To represents objects and the roles they play.

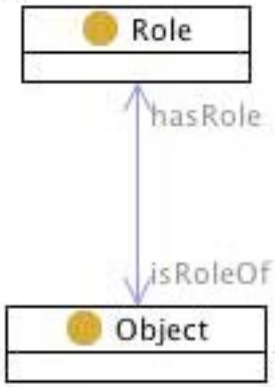


Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character

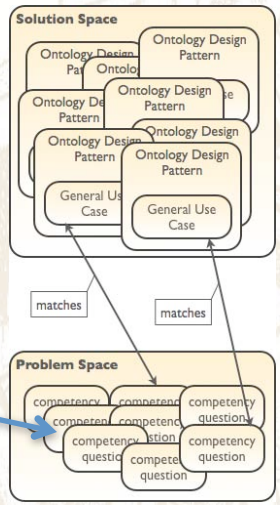
- To represents objects and the roles they play.





Analyze the sentence, detect the modeling issues, and match to the CPs

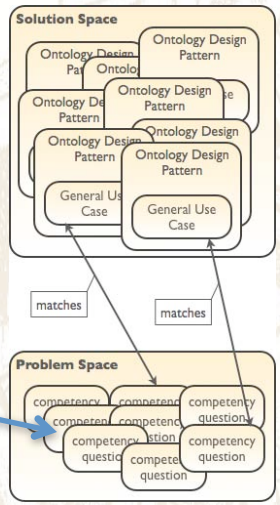
- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- The play of some drama



Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- The play of some drama

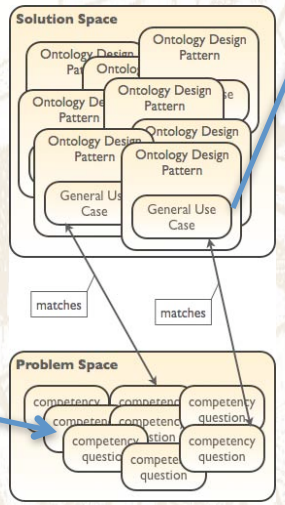
- To distinguish information objects from their concrete realizations.

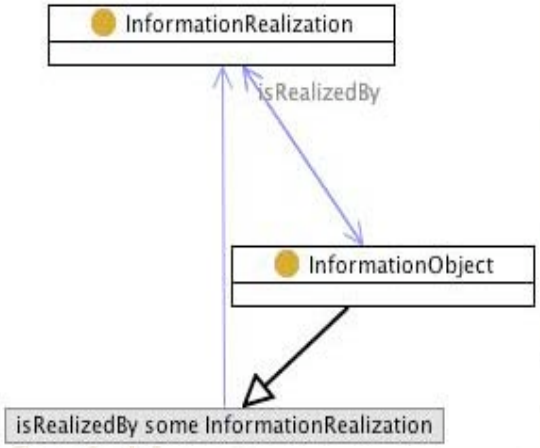


Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- The play of some drama

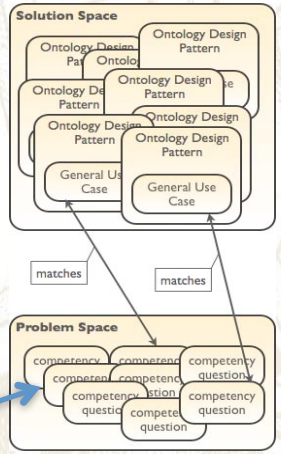
- To distinguish information objects from their concrete realizations.





Analyze the sentence, detect the modeling issues, and match to the CPs

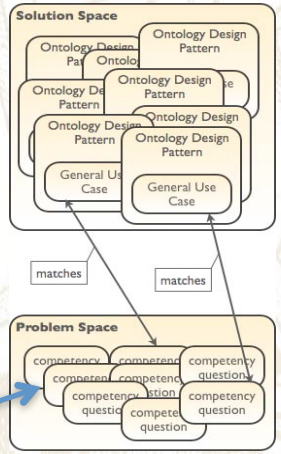
- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A time period



Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A time period

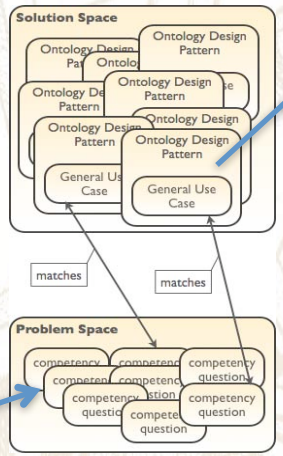
- To represent time intervals, their start/end dates, and any dates falling into the period

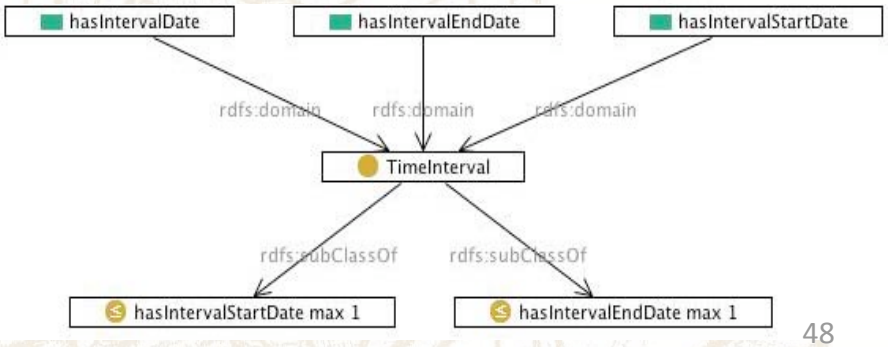


Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A time period

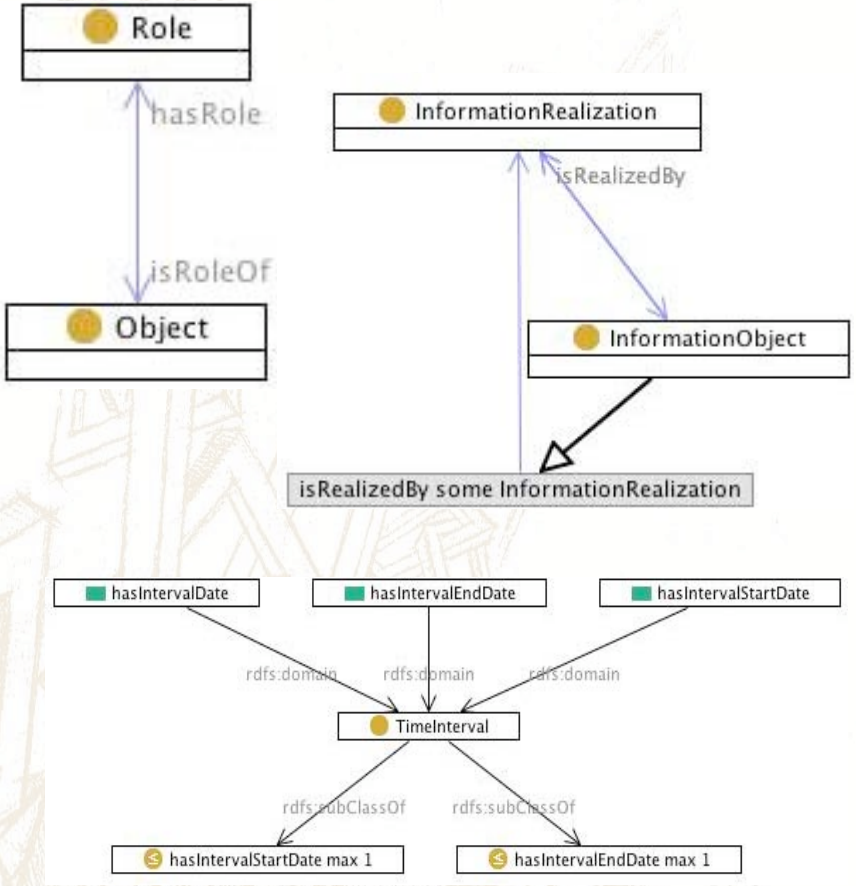
- To represent time intervals, their start/end dates, and any dates falling into the period





Analyze the sentence, detect the modeling issues, and match to the CPs

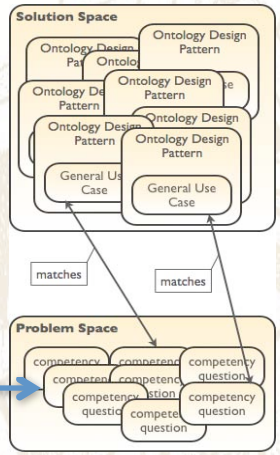
- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character in a play of a drama, given at a theater during a time period
- How can we relate them together?



Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character in a play of a drama, given at a theater during a time period

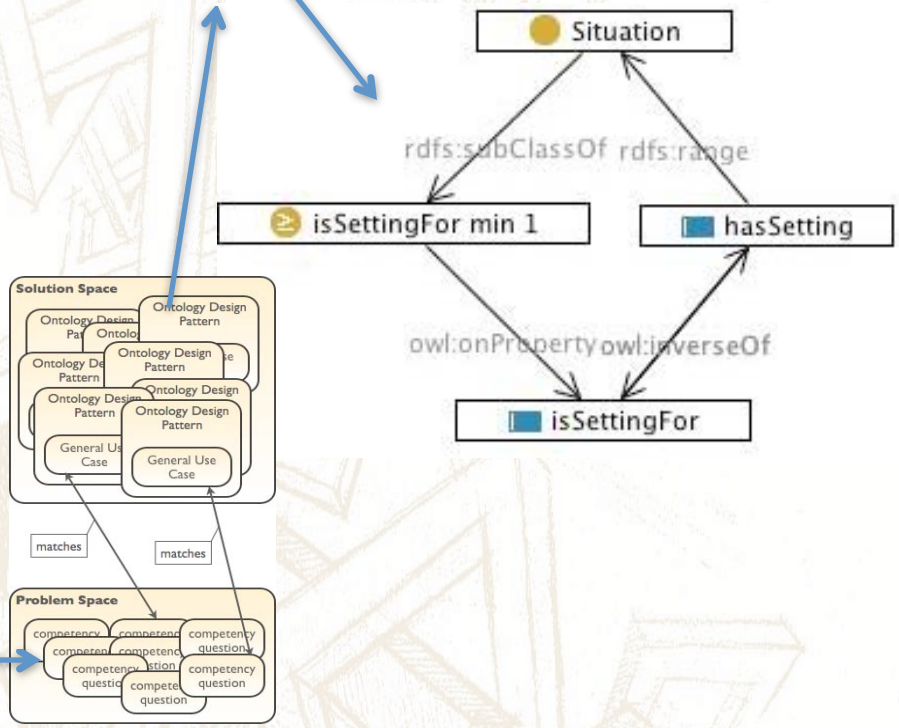
- A situation, a set of circumstances in a defined setting



Analyze the sentence, detect the modeling issues, and match to the CPs

- Arnold Schwarzenegger is Shylock in the play of "Merchant of Venice", that is given at the theater "Roma" during September and October 2009
- A person plays a character in a play of a drama, given at a theater during a time period

- A situation, a set of circumstances in a defined setting

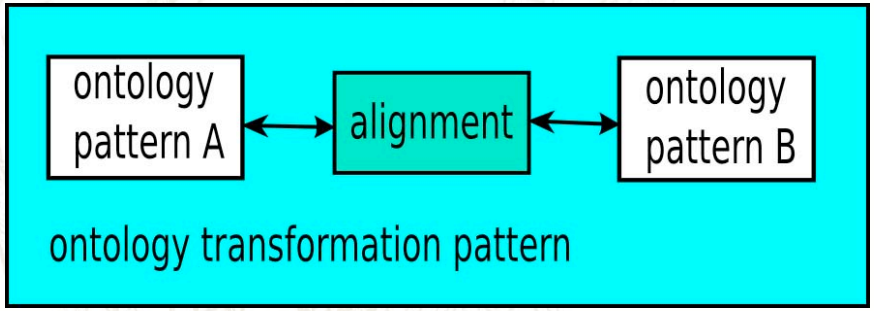


Reengineering ODPs

- Reengineering ODPs contain transformation rules applied in order to create a new ontology (*target* model) starting from elements of a *source* model
- The target model is an ontology, while the source model can be either an ontology, or any other knowledge resource
  - e.g., a thesaurus concept, a data model pattern, a UML model, a linguistic structure, etc.

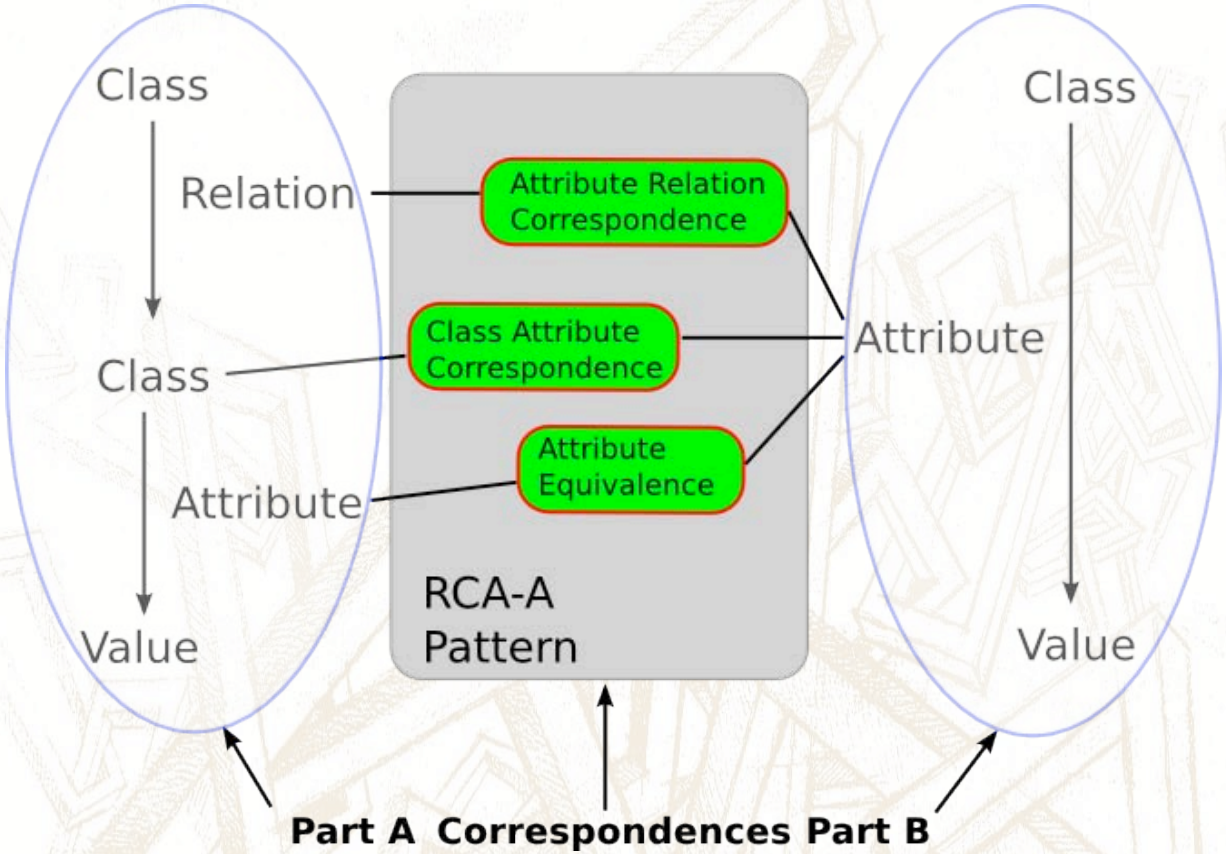
Relation to other ODPs

- Correspondence patterns can make use of other ODPs:
  - Name patterns
  - Logical patterns
  - Content ODPs
- Alignment patterns might result from linking two equivalent ODPs
- Correspondence patterns can be used together with content patterns for parts A and B



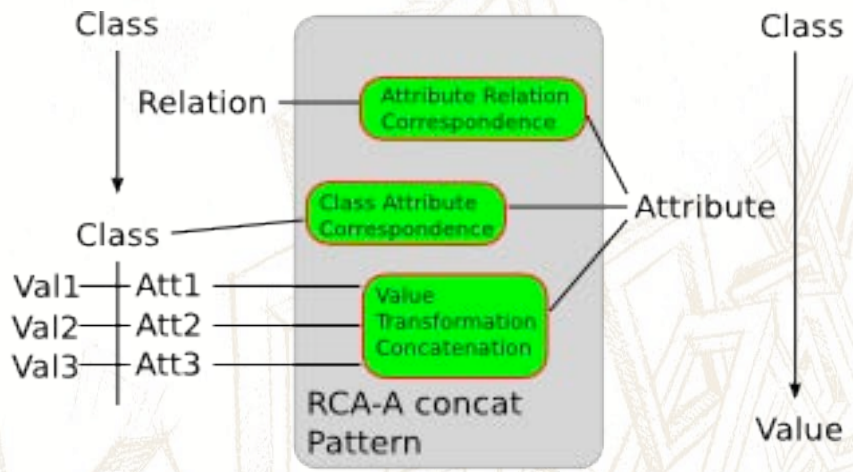
Ex: An ontology transformation pattern (refactoring pattern) making use of two Content patterns and an alignment between them to transform from one structure to the other. (See application at the end of the lecture)

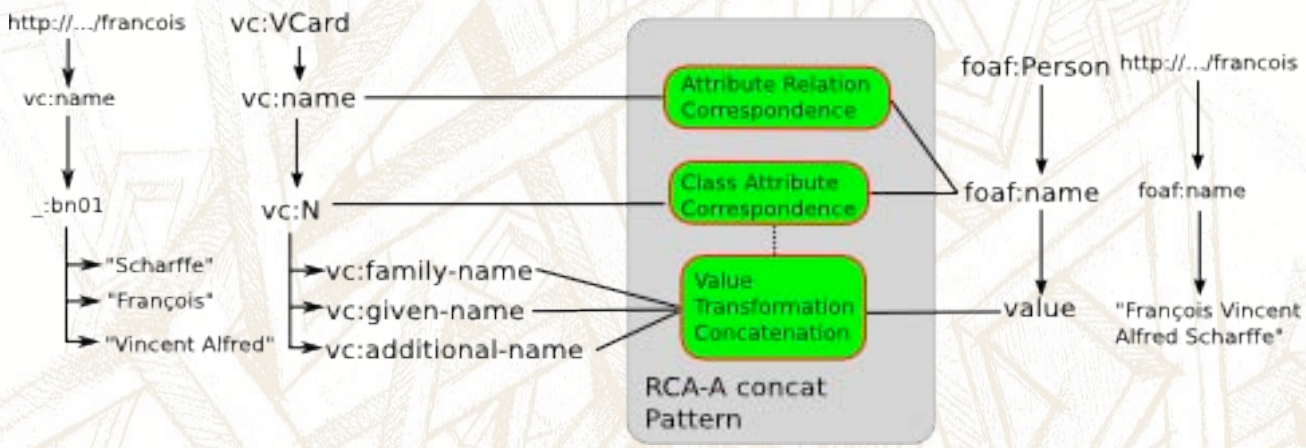
Particularities of correspondence ODPs



This alignment corresponds to a **Granularity Mismatch**

Example: FOAF-VCard





Other alignment patterns

- Some alignment patterns are interpretation-preserving (IP), i.e. the semantics arising from the composition of the two resources stays in OWL-DL, while other patterns do not bear that characteristic (NIP), and therefore induce a non-standard semantics:
  - ***Individual is equivalent to class*** (NIP):
    - (m1) *asfad:Aquatic_organisms* owl:equivalentClass FI_commodities:FI_commodity_classification
    - (m2) *asfad:Stocks* owl:equivalentClass stock:Stock
    - (m3) *asfad:Water_bodies* owl:equivalentClass water:FAO_fishing_area
  - ***Individual is subclass of class*** (NIP):
    - (m4) *asfad:Ocean_space* rdfs:subClassOf water:FAO_fishing_area
  - ***Individual has type class*** (IP):
    - (m5) *asfad:Research_vessels* rdf:type vessels:vessel_classification

Reasoning ODPs

Reasoning ODPs are applications of Logical ODPs oriented to obtain certain reasoning results, based on the behavior implemented in reasoning engines

Examples of Reasoning ODPs

- Precise
  - Classification
  - Subsumption
  - Inheritance
  - Materialization
  - Rule firing
  - Constructive query
  - ...
- Approximate
  - Approximate classification
  - Similarity induction
  - Taxonomy induction
  - Relevance detection
  - Latent semantic indexing
  - Automatic alignment
  - ...

or some workflow of them, e.g.

1) *Materialize property chains*
2) *Run classifier+subsumer*
3) *Run constructive query*
4) *Assert results*
5) *Run similarity engine*
6) *Assert results*

Presentation ODP

Presentation ODPs deal with usability and readability of ontologies from a user perspective

Examples of Naming ODPs

- Namespace declared for ontologies.
- It is recommended to use the base URI of the organization that publishes the ontology
  - e.g. http://www.fao.org for the FAO
- followed by a reference directory for the ontologies
  - e.g. http://www.fao.org/ontologies/
- It is also important to choose an approach for encoding versioning, either on the name, or on the reference directory

Benefits of naming patterns for humans (I)

- Example: Elementary triple
  - IBM product ThinkPad
- What does this say?
  - „IBM has a product called ThinkPad“?
  - or, actually, „IBM is a product of ThinkPad“?
- What is the issue?
  - Trivial violation of naming pattern: plain noun (‘product’) used as object property name

Benefits of naming patterns for humans (II)

- There can be multiple adequate naming patterns
- IBM produces ThinkPad
  - The ‘semantic direction’ of the relation is made clear
  - ‘produces’ can be understood as ‘manufactures’ as processive relationship
- IBM has_product ThinkPad
  - The use of ‘has’ can be sign of a stative relationship, often to a dependent entity
  - “IBM has ThinkPad as its product” (even if it is no longer manufactured)

Other examples of naming patterns

- 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 Annotation ODPs

- RDF Schema labels and comments (crucial for manual selection and evaluation)
- Each class and property should be annotated with meaningful labels
  - i.e., by means of the annotation property rdfs:label, with also translations in different languages
- Each ontology and ontology element should be annotated with the rationale they are based on
  - i.e., by means of the annotation property rdfs:comment

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.

17. ODP Portal, http://www.ontologydesignpatterns.org