

Ontology Design Patterns:

A primer, with applications and perspectives

Aldo Gangemi
Laboratory for Applied Ontology
Institute of Cognitive Sciences and Technology
CNR, Rome, Italy
aldo.gangemi@istc.cnr.it

Code conventions

- XML namespaces for OWL abstract syntax (cf. <http://douce.semanticweb.org> for abbreviation resolution)
- UML class and activity diagrams
- Non-standard use of UML to visualize CODEPs:
 - generalisation → subsumption (“subClassOf”)
 - association → two-way conceptual relation (“property”)
 - association with no cardinality: 0..*
 - box → “class”
 - dashed box → “individual”
- OWL abstract syntax



Anything in common? (1)

- John and Mary are bunkmates
- John picked up Mary at the bar
- *Similar entities: yes*
- *Similar case: no*
- *Both are cases of co-participation: yes*



Anything in common? (2)

- Our plan is simple; it is comprehensive; and it is what our constituents want. They have raised just one issue: they say it costs too much.
 - http://www.house.gov/commerce_democrats/press/108st47.htm
- While the diagnosis is simple, the pathological changes in the dog's body make treatment complicated, expensive, and not always successful.
 - <http://www.canismajor.com/dog/bloat.html>
- *Similar entities: no*
- *Similar case: no*
- *Both are cases of co-participation: no*
- *Both are cases of super-description: yes*



Anything in common? (3)

- John *picked up* Mary at the bar; she's now at home
- John *picked up* Mary at the bar; he's now in love with her
- «As soon as the rattle died down, John *picked up* Mary and carried her for fifty feet to ensure her safety, and I followed a few feet behind.» <http://www.arizonahikingtrails.com/nakedwoodspages/040205.html>
- «From the afternoon I *picked up* Mary Jo Putney's *The Rake & the Reformer*, finished it by dinnertime and went back for more, I was hooked.» <http://www.likesbooks.com/karen.html>
- *Similar terms: yes*
- *Similar case: no*
- *All are cases of co-participation: yes*
- *All are cases of overdescription: no*



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 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>



Problems with reference ontologies

- Unfortunately, reference ontologies from the past and current times are often hard to exploit: usually large, heavily axiomatized, not easily reducible to a mild expressivity, often use philosophical or unusual names in their signature, are assumed to be imported as a whole (non-modular), are hardly built from real data or corpora, although assumed as mostly task-independent, their design reflects use cases that are usually quite far from real applications ...



Solutions?

- **Just give up with reference ontologies**
 - pros: stop bothering with reusability
 - cons: difficult agreement and quality-checking
- **Use reference taxonomies**
 - prof: efficient reasoning
 - cons: no structure/rationales to take much advantage
- **Split reference ontologies into generic modules**
 - pros: better processing
 - cons: what criteria for modularization? still not enough freedom from monolithic views of the world
- **Use patterns instead of reference ontologies**
 - pros: flexibility, small pieces, close to cognition and good interfaces
 - cons: how to represent and reason with them?



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*
- *Recent from W3C SWBPD: OWL modelling best practices, semantic SE patterns, techniques to vocabulary porting and migration to the SW*
- *Started in EU NeOn project: ontology design is the primary concern with ontology networking, contextualizing, and interfacing*



An example of a low-hanging fruit: FN lexico-semantic patterns

- http://framenet.icsi.berkeley.edu/index.php?option=com_wrapper&Itemid=118&frame=Residence

- **OntoFrameNet translation**

- Individual(F_Residence
 - type(Frame)
 - value(hasFE FE_Resident_535)
 - value(hasFE FE_Location_536)
 - value(hasFE FE_Co-resident_537))
- Class(Residence partial
 - FramedSituation
 - restriction(settingFor_1 allValuesFrom(Resident))
 - restriction(settingFor_1 someValuesFrom(Resident))
 - restriction(settingFor_2 allValuesFrom(Location))
 - restriction(settingFor_3 allValuesFrom(Co-resident))))
 - Individual(FE_Resident_535
 - type(FrameElement)
 - type(restriction(classifies allValuesFrom(Resident))
 - value(centrality core))
 - Class(LivingIn partial Residence)
 - Class(LivingWith partial Residence)

cf. <http://www.loa-cnr.it/ontologies/FrameNet/ofn.owl>

Examples:

- Marko LIVES in Ljubljana
- Boris still LIVES with his parents



Summary

- **Differences between ontology patterns and other patterns**
- A use case registry for *content* ontology design patterns (CODEP)
- Examples of, and operations on CODEPs
- How to frame CODEPs (a meta-model)



What's a pattern?

- Historical sense, e.g. OED's: middle-age Latin "patronus", meaning "patron", and, metonymically, "exemplar": *something proposed for imitation*
- General sense, e.g. Webster's, (f) entry: «*a discernible coherent system based on the intended interrelationship of component parts*»
- Theoretical architecture, Alexander (1979): *archetypal solutions to design problems in a certain context*
- Software engineering, (Gamma et al. 1995, Baker et al. 1999, Maplesden et al. 2002): *formatted guidelines in software reuse, recently also attempts at formalization*
- Data modelling, e.g. Hay 1996: *a convention of thought to be encoded in a modelling language like ER or UML, and then reused for local conceptual schemas*
- Knowledge engineering, Clark 1997: «*a theory template or schema , which denotes a structure of objects and relationships, but whose axioms are not directly part of the global KB*»
- Ontology engineering and the semantic web, (Reich 2000, Gangemi 2003,2005, W3C SWBPD 2003-5, Soshnikov 2003, Guizzardi et al. 2004, Svatek 2004, Vrandecic 2005): *various schemas and macros for UML, OWL, core ontologies, etc.*



Peter Clark's idea

- A pattern is a theory template. It denotes a *structure that is invariant under signature transformation*. 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.]*
- $\forall c \exists p ((\text{consumer}(c) \wedge \text{producer}(p) \wedge \text{connects}(c,p)) \rightarrow \text{supplied}(p))$
 - via *signature morphism* becomes e.g. in an application:
- *[If a light is connected to a battery, then it is powered.]*
- $\forall c \exists p ((\text{light}(c) \wedge \text{battery}(p) \wedge \text{connects}(c,p)) \rightarrow \text{powered}(p))$
- But if a pattern is just an *untyped structure*, there are no ways to distinguish a *logical* vs. a *conceptual* pattern, and we should concentrate only on e.g. OWL macros (cf. Vrandečić, 2005) or SWBPD OE patterns (see Alan's presentation).



Conceptual (content) vs. logical patterns

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

(subClassOf
 (intersectionOf
 Consumer
 (restriction (connects someValuesFrom Producer)))
 Supplied)

*Content pattern:
specific (non-logical)
vocabulary*

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

(subClassOf
 (intersectionOf
 Thing
 (restriction (AnyProperty someValuesFrom Thing)))
 Thing)

*Logical pattern:
no specific vocabulary*



Clark (2000), more explicit

- “A pattern is a first-order theory whose axioms are not part of the target knowledge base, but can be incorporated via a renaming of their non-logical symbols”
- “A theory acquires its status as a pattern by the way it is used, rather than by having some intrinsic property”
- A pattern is implemented “as an explicit, self-contained theory”, then is category-theoretically *morphed* “for each intended application in the target knowledge base”



Signature morphisms

“A signature morphism (in the context of this category) is a consistent mapping of the pattern’s ... signature, to [another], specifying how the pattern should be transformed. Finally ... morphed copies of this pattern are imported, one for each morphism” (cf. Burstall&Goguen’s *derive* operation for algebraic theories). I.e., given two specifications (patterns): $\langle \text{Sig}_1, \Delta_1 \rangle$ $\langle \text{Sig}_2, \Delta_2 \rangle$
A signature morphism applies to them iff:

$$\forall a \in \Delta_1, (\Delta_2 \vdash M(a))$$

Every axiom a in Δ_1 , after being translated by M , follows from Δ_2



A distinction between logical and content patterns

- Signature morphism can apply either *downwardly* or *upwardly*, depending on the subsumption steps implied by the morphisms
 - $M(\text{consumer}) \text{ :-> light}$, $M(\text{producer}) \text{ :-> battery}$, $M(\text{supplied}) \text{ :-> powered}$
- Logical Ontology Design Patterns (LODeP) are invariant under either downward or upward signature morphism
- Content Ontology Design Patterns (COPeP) are only invariant under downward signature morphism



Pragmatic features of CODEPs

- A CODEP is a template for representing and possibly solving a domain modelling problem
- A CODEP "extracts" a fragment of a RO, which is its "background"
- A CODEP can be represented in any ontology representation language, but its intuitive and compact visualization seems an essential requirement
- A CODEP can be an element in a partial order, where the ordering relation requires that at least one of the classes or relations in the CODEP are specialized
- A CODEP should be intuitively exemplified, and should catch relevant, "core" notions of a domain. Independently of the generality at which a CODEP is singled out, it must contain the central notions that "make rational thinking move" for an expert in a given domain for a task
- A CODEP can/should be used to describe a "best practice" of modelling
- A CODEP can be often built from informal schemata used by domain experts. Typically, experts spontaneously develop schemata to improve their business, and to store relevant know-how. These schemata can be reengineered with appropriate methods
- A CODEP is similar to a DB schema, but a CODEP is defined wrt a reference ontology and should have a general character, independently from local design details



Looking for Ontology Design Patterns

- *Search for invariances*
 - Across existing ontologies, templates, methods, practices, rules, vocabularies, linguistic structures, social networks, cognitive theories, etc.
- (Formally) *encode the invariance*
 - Some existing repositories (FrameNet, Hay's data modelling patterns, etc.)
- *Annotate* it as a method to help solving a usage problem (a *Generic* or *Local* Use Case)
- *Store* the framed (annotated) invariance in the form of a CODEP [alternative: *manage* a library of ontologies, each one specifying a pattern]
- *Specialize&Compose*

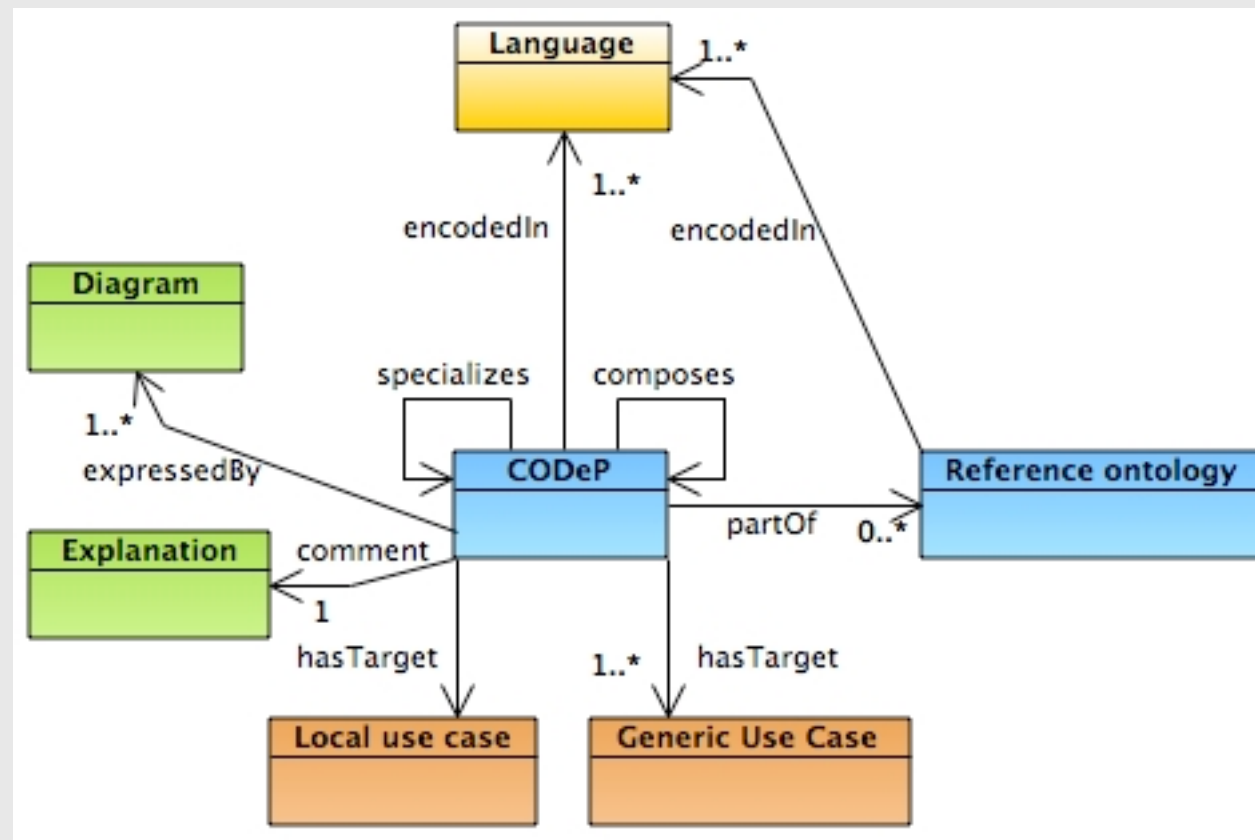


A logical pattern (Classes as Values)

Slot	Value
General issue	It is often convenient to put a class (e.g., <i>Animal</i>) as a property value (e.g., topic or book subject) when building an ontology. While OWL Full and RDF Schema do not put any restriction on using classes as property values, in OWL DL and OWL Lite most properties cannot have classes as their values.
Use case example	Suppose we have a set of books about animals, and a catalog of these books. We want to annotate each catalog entry with its subject, which is a particular species or class of animal that the book is about. Further, we want to be able to infer that a book about African lions is also a book about lions. For example, when retrieving all books about lions from a repository, we want books that are annotated as books about African lions to be included in the results.
Notation	In all the figures below, ovals represent classes and rectangles represent individuals. The orange color signifies classes or individuals that are specific to a particular approach. Green arrows with green labels are OWL annotation properties. We use abstract syntax to represent the examples.
Approaches	Approach 1: Use classes directly as property values In the first approach, we can simply use classes from the subject hierarchy as values for properties (in our example, as values for the <i>dc:subject</i> property). We can define a class <i>Book</i> to represent all books.
Considerations	<ul style="list-style-type: none">• The resulting ontology is compatible with RDF Schema and OWL Full, but it is outside OWL DL and OWL Lite.• This approach is probably the most succinct and intuitive among all the approaches proposed here.• Applications using this representation can directly access the information needed to infer that <i>Lion</i> (the subject of the <i>LionsLifeInThePrideBook</i> individual) is a subclass of <i>Animal</i> and that <i>AfricanLion</i> (the subject of the <i>TheAfricanLionBook</i> individual) is a subclass of <i>Lion</i>.
OWL code (abstract syntax)	<pre>Class(<i>BookAboutAnimals</i> partial Thing unionOf(restriction(<i>dc:subject</i> someValuesFrom(<i>Animal</i>)) restriction(<i>dc:subject</i> someValuesFrom(restriction(<i>rdfs:subClassOf</i> hasValue(<i>Animal</i>))))))</pre>



The CDeP datamodel



Summary

- Differences between ontology patterns and other patterns
- **A use case registry for *content* ontology design patterns (CODEP)**
- Examples of, and operations on CODEPs
- How to frame CODEPs (a meta-model)



Generic use cases

Competency questions, PSM, Q/A

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

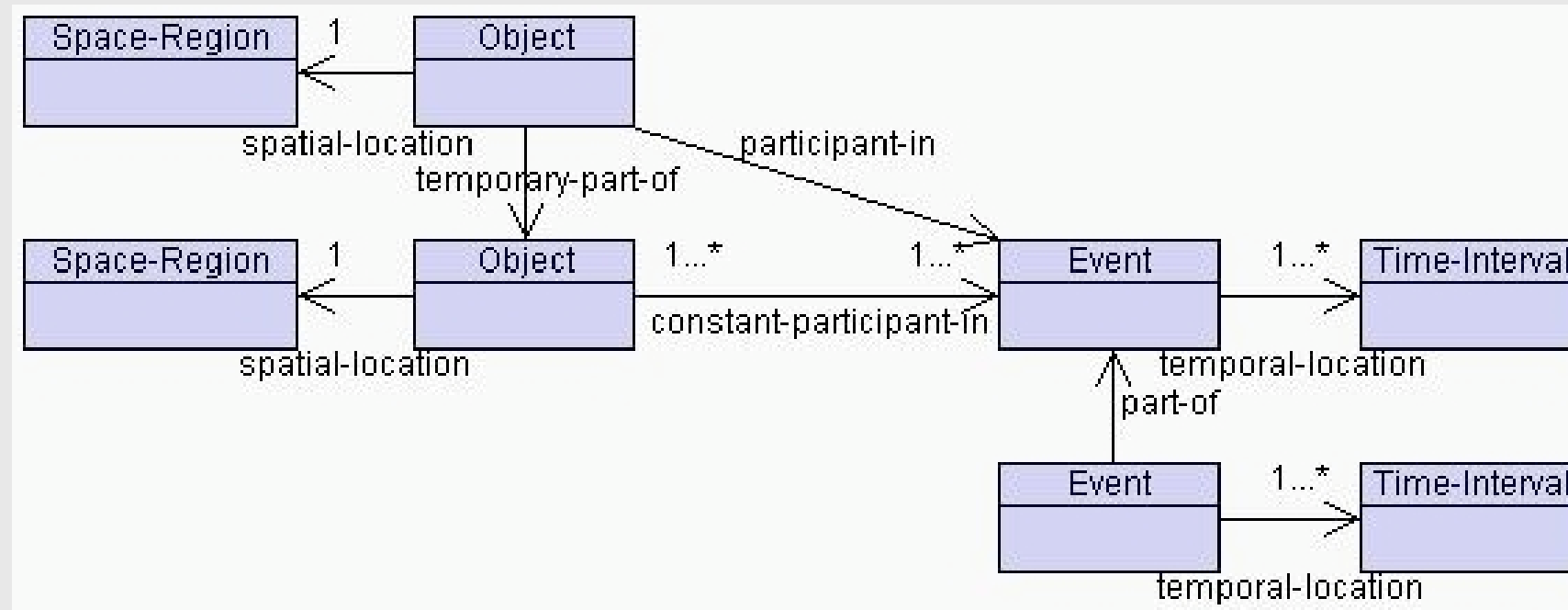


Summary

- Differences between ontology patterns and other patterns
- A use case registry for *content* ontology design patterns (CODEP)
- **Examples of, and operations on CODEPs**
- How to frame CODEPs (a meta-model)



Participation pattern (from DOLCE)



cf. <http://www.loa-cnr.it/ontologies/DOLCE-Lite.owl>

Examples:

“Members themselves will PARTICIPATE in the final selection”

“On the other hand, girls have not PARTICIPATED strongly in male-dominated subjects”

“Francesco Totti PLAYED the ball to Zambrotta and the full-back went on to FIRE a left-foot shot from outside the area”

“An object at rest tends to STAY AT REST and an object in motion tends to STAY IN MOTION with the same speed and in the same direction unless ACTED UPON BY an unbalanced force”

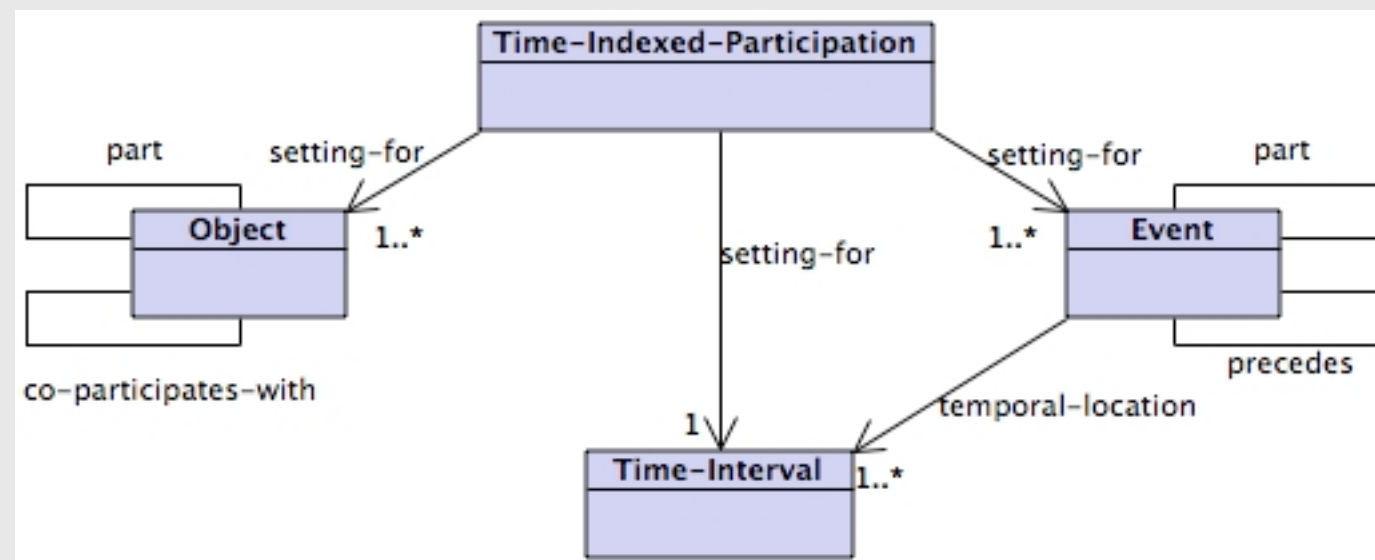
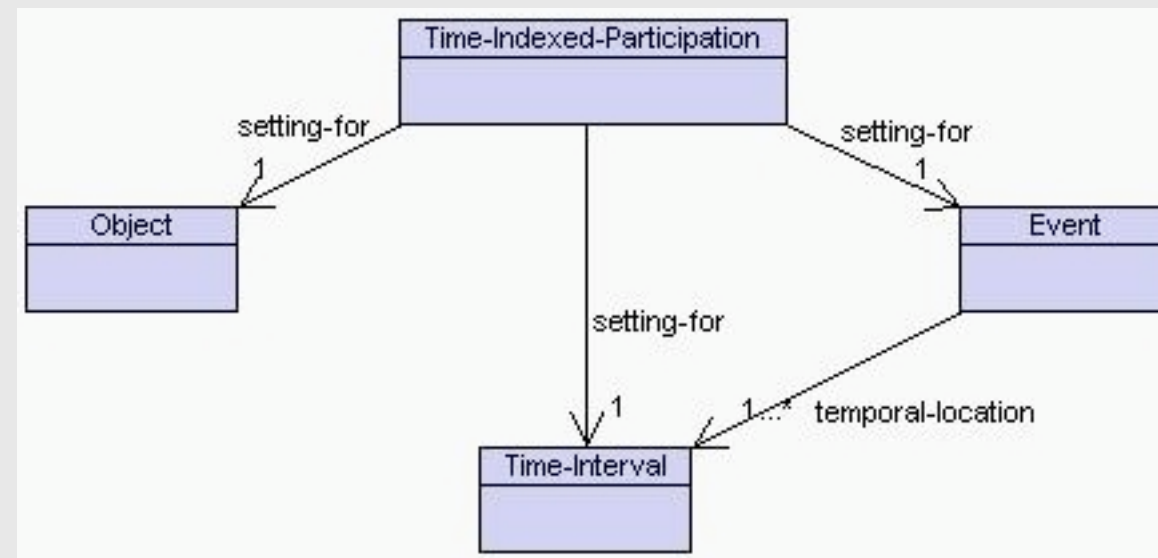


cf. <http://www.loa-cnr.it/ontologies/ExtendedDnS.owl>

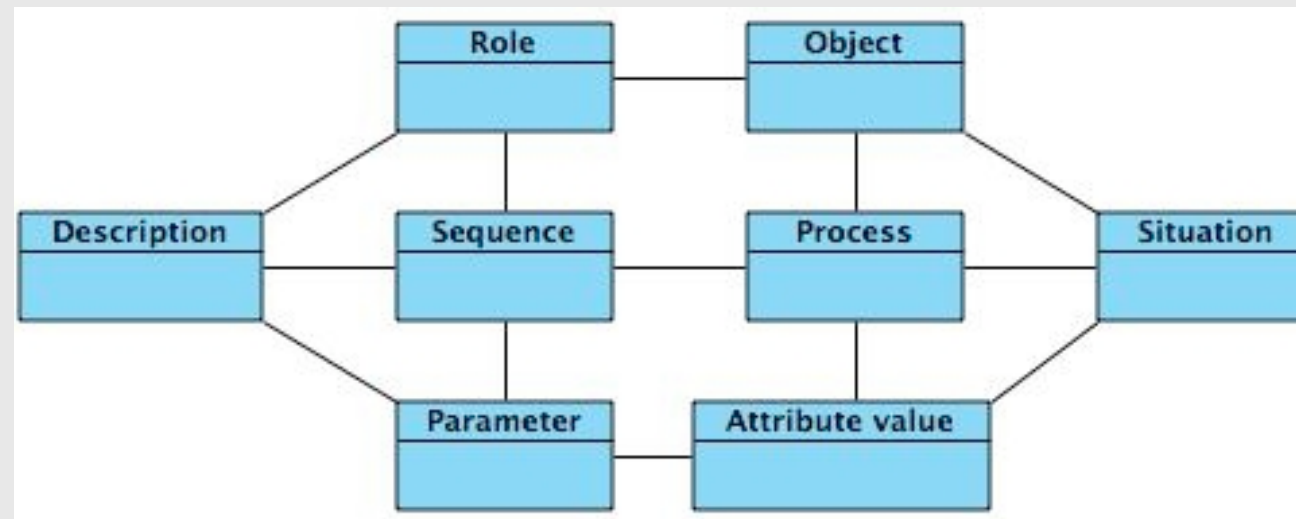
“Members themselves will PARTICIPATE in the final selection”

“On the other hand, girls have not PARTICIPATED strongly in male-dominated subjects”

“Francesco Totti PLAYED the ball to Zambrotta and the full-back went on to FIRE a left-foot shot from outside the area”



Description<->Situation, informally



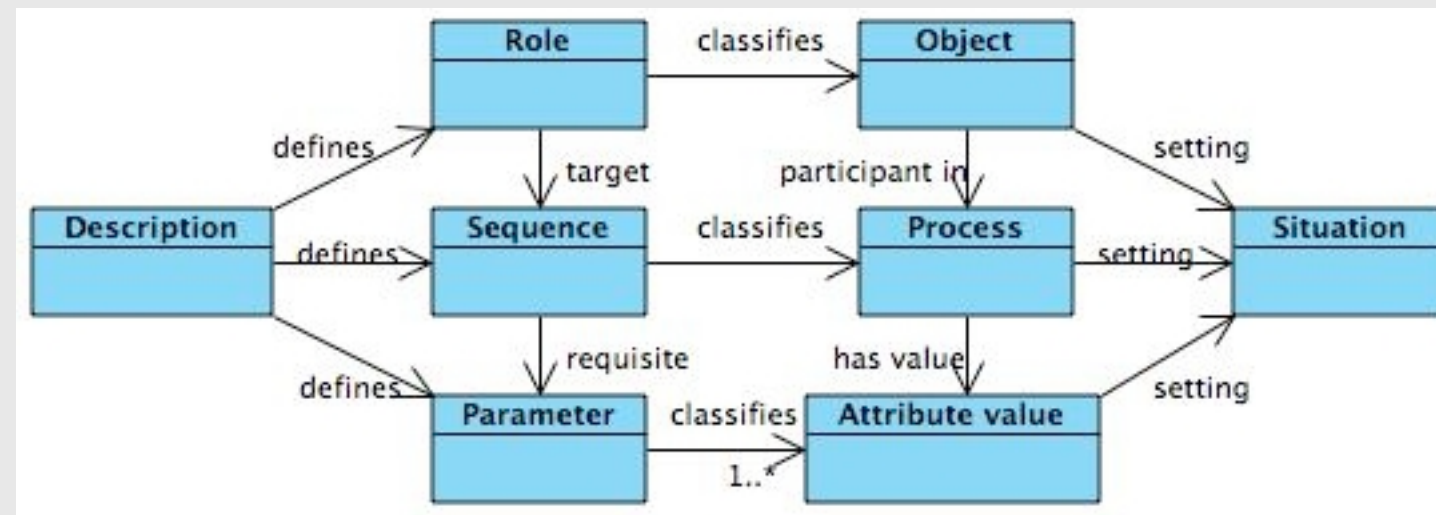
cf. <http://www.loa-cnr.it/ontologies/ExtendedDnS.owl>
<http://www.loa-cnr.it/ontologies/DUL.owl>

Examples:

- The European Union has created rights for air passengers, and is working to enforce them
- Two-boat operated purse seine is a catching technique involving the use of a purse seiner
- The insurers ended up in the bind of having to pay 15 years' taxes
- Discuss the things that guided you in making your decision
- The descriptions of situations and actions relevant to these situations are extended when new concrete situations are encountered
- Canada's Global Role: A Strategic Assessment of its Military Power
- Dive Into Greasemonkey is a book about programming with Greasemonkey, a Firefox extension for customizing web pages



With axioms



cf. <http://www.loa-cnr.it/ontologies/ExtendedDnS.owl>
<http://www.loa-cnr.it/ontologies/DUL.owl>

Examples:

The insurers ended up in the BIND of having to pay 15 years' taxes

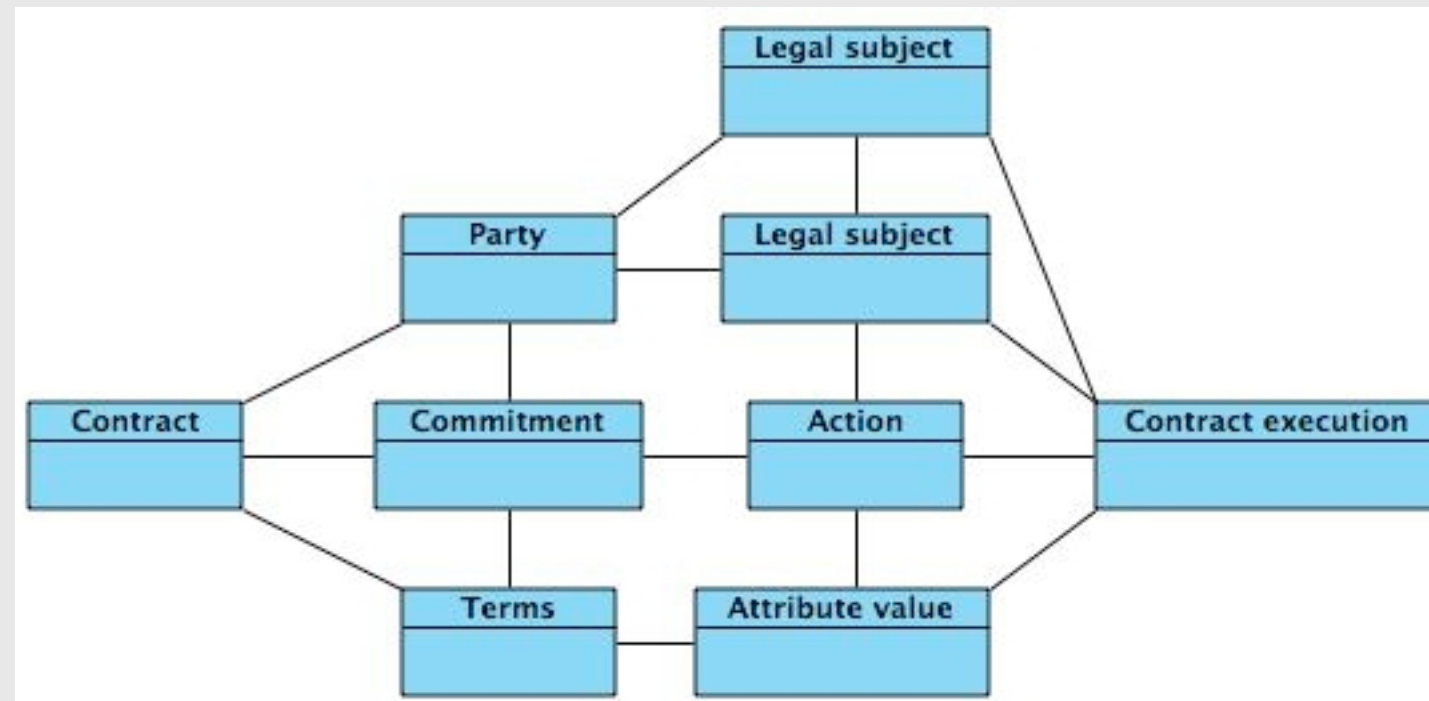
Discuss the things that GUIDED you in making your decision

The descriptions of situations and actions relevant to these situations are extended when new concrete situations are encountered

Canada's Global Role: A Strategic Assessment of its Military Power



Legal specialization: Contract<->Execution



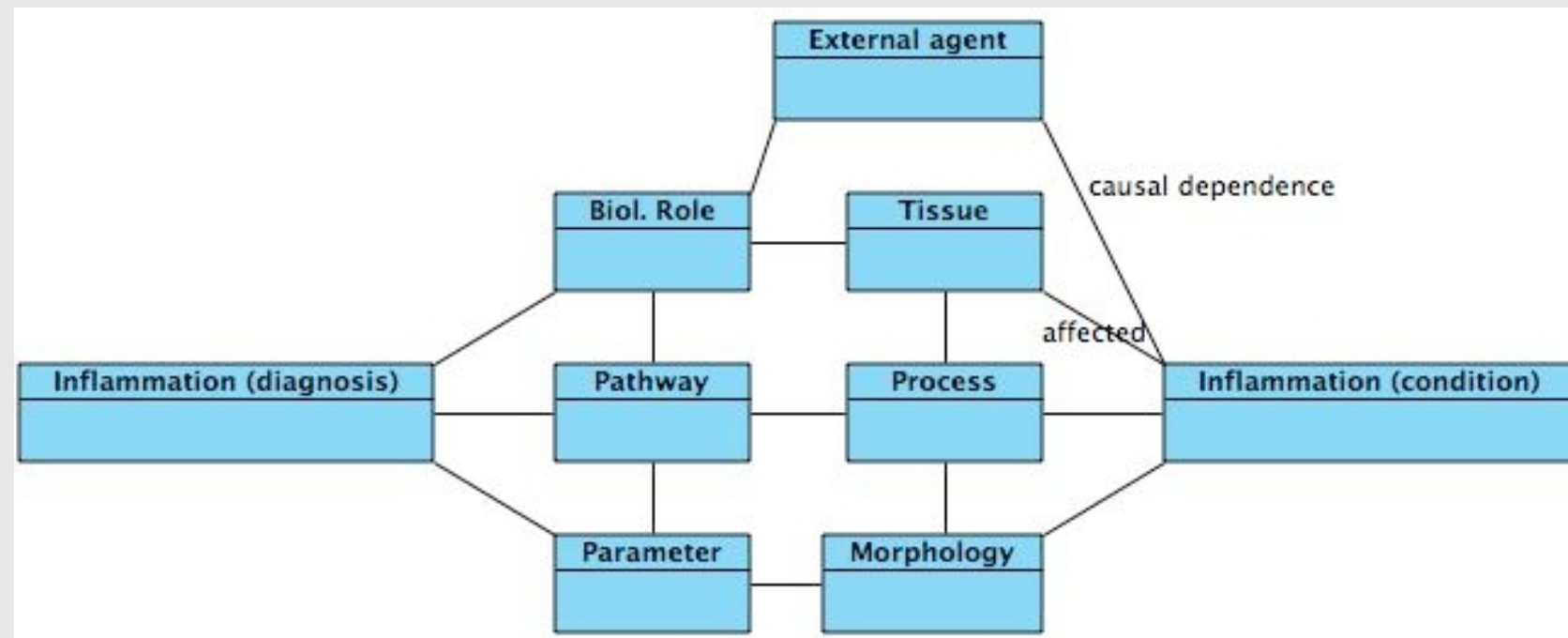
cf. <http://www.loa-cnr.it/ontologies/CLO/CoreLegal.owl>

Examples:

- The contracted parties oblige themselves to handle all information and data obtained from the other party in the course of executing this contract in a ...
- Italian maximum urban speed limit is 50kmph [social norm]
- The European Union has created rights for air passengers, and is working to enforce them



Biomedical specialization: Diagnosis<->Condition



Example: The pathogenesis in acute disseminated encephalomyelitis results in perivascular edema, inflammation, and demyelination. It is also postulated that there may be associated vasculitis.



Diagnosis<->Condition as underlying the polysemy of *inflammation*

1. “Inflammation segregates external agents” (*physiological function*)
 2. “The inflammation has a diameter of 5 cm.” (*portion of a body part*)
 3. “The inflammation has changed its shape” (*abnormal morphology*)
 4. “The inflammation evolved during three weeks” (*clinical condition*)
 5. “The inflammation is severe” (*diagnosis*)
- inflammation#2 participates in inflammation#1
 - inflammation#3 is an attribute of inflammation#2
 - inflammation#4 is the setting for the inflammation#1, #2, #3, possibly satisfying inflammation#5
 - inflammation#5 defines concepts that classify entities in #4

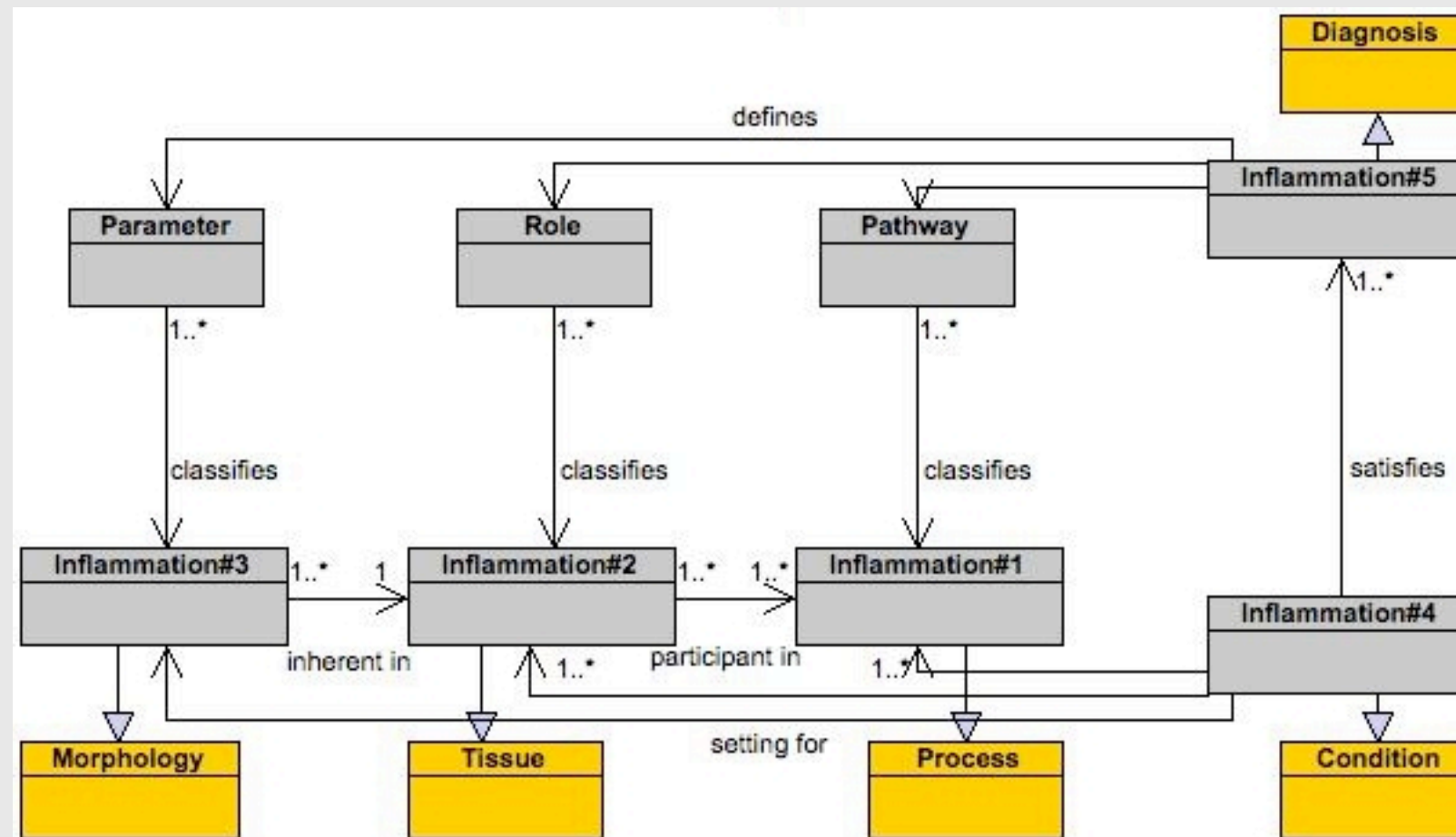


Hepatitis-related services and patterns

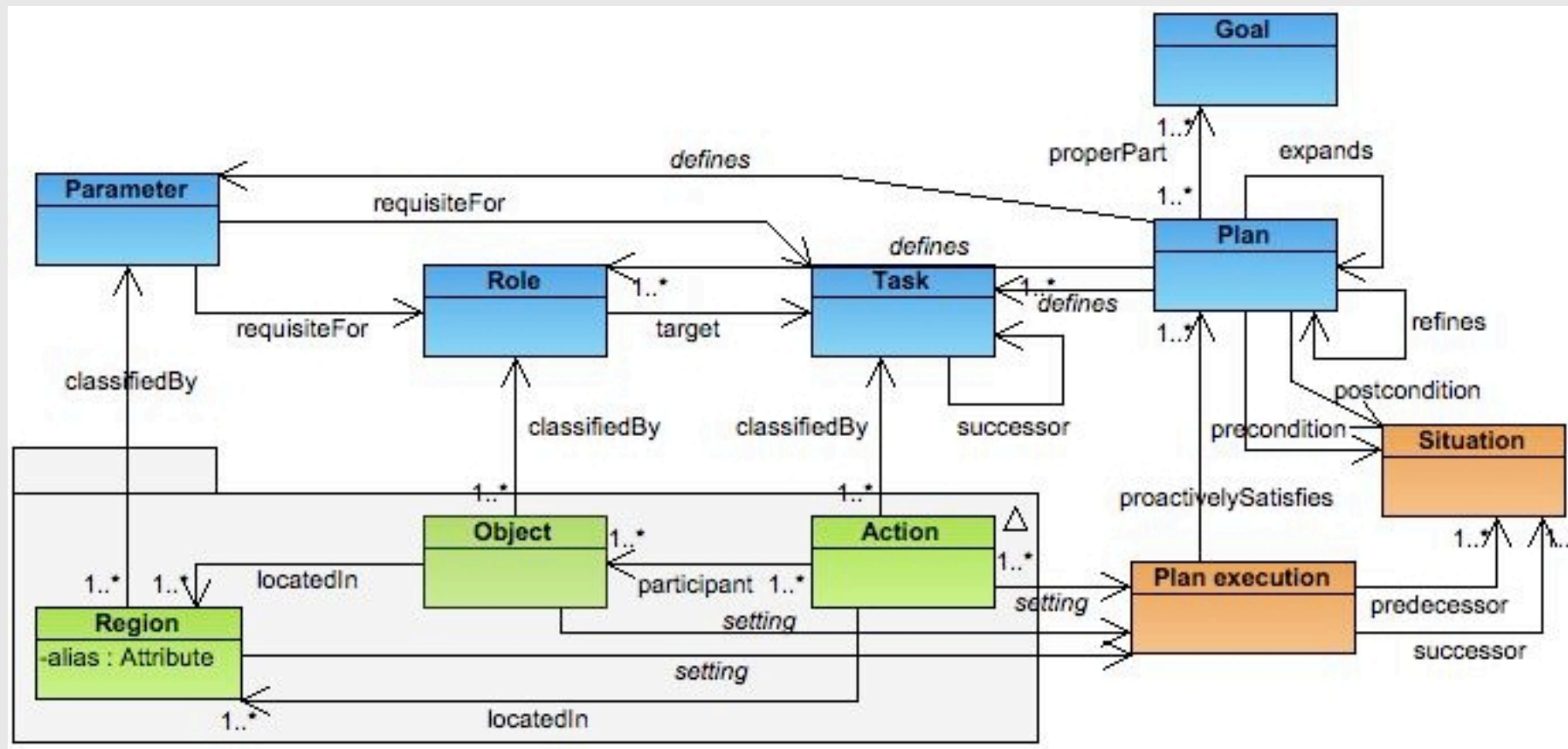
- According to task, our selection criteria may change greatly
- Service#1: how much does a hepatitis cost to the hospital?
 - Conceptualization needed: *hepatitis as clinical condition*
- Service#2: what lab tests are routinely prescribed for a hepatitis?
 - Conceptualization needed: *hepatitis as process/tissue*
- Service#3: what records must be filled into a patient record for a hepatitis?
 - Conceptualization needed: *hepatitis as condition/diagnosis/process/tissue/morphology*
- Service#4: how to integrate the other services?
 - Conceptualization needed: complete *hepatitis* semantics; i.e. all above notions, *plus* (unique) relations among them



Mapping *infl.* polysemy to the Diagnosis<->Condition pattern



Plan<->Execution pattern



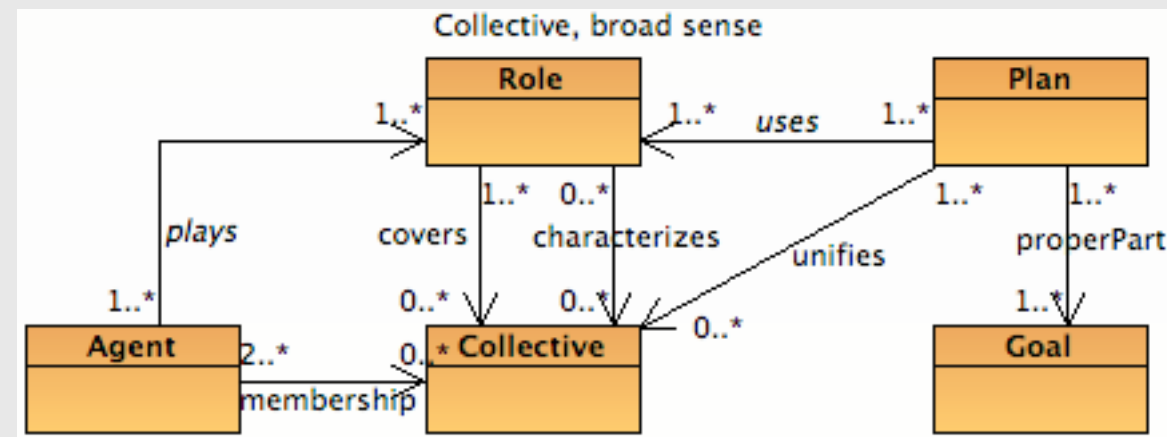
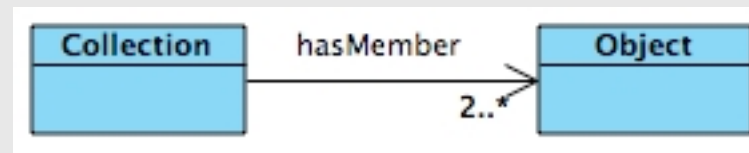
cf. <http://www.loa-cnr.it/ontologies/Plans.owl>

Examples:

- The school will continue with its PROJECT to end all violence related to racism, xenophobia and intolerance
- The prime GOAL of the programme was to help develop processes of scientific thinking in children
- Michael EXPECTED Abby to demand examples
- FIAT workers have decided a strike



Collection<->Entity pattern



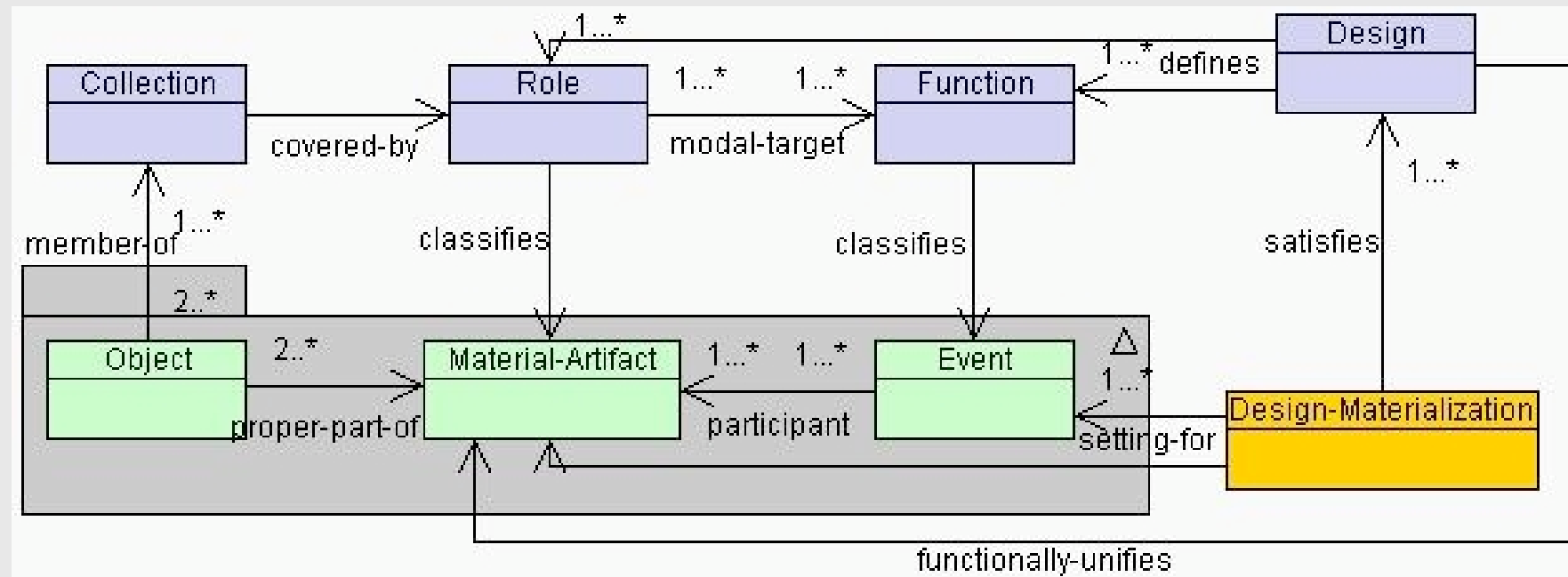
cf. <http://www.loa-cnr.it/ontologies/SocialUnits.owl>

Examples:

- FIAT workers have decided a strike
- The Presidency of the Security Council is held in turn by the members of the Security Council in the English alphabetical order of their names
- The Collection of Laws for Electronic Access (CLEA) is a unique electronic database providing easy access to intellectual property legislation
- An ARMY of postal workers descended on my office



Design<->Artifact pattern



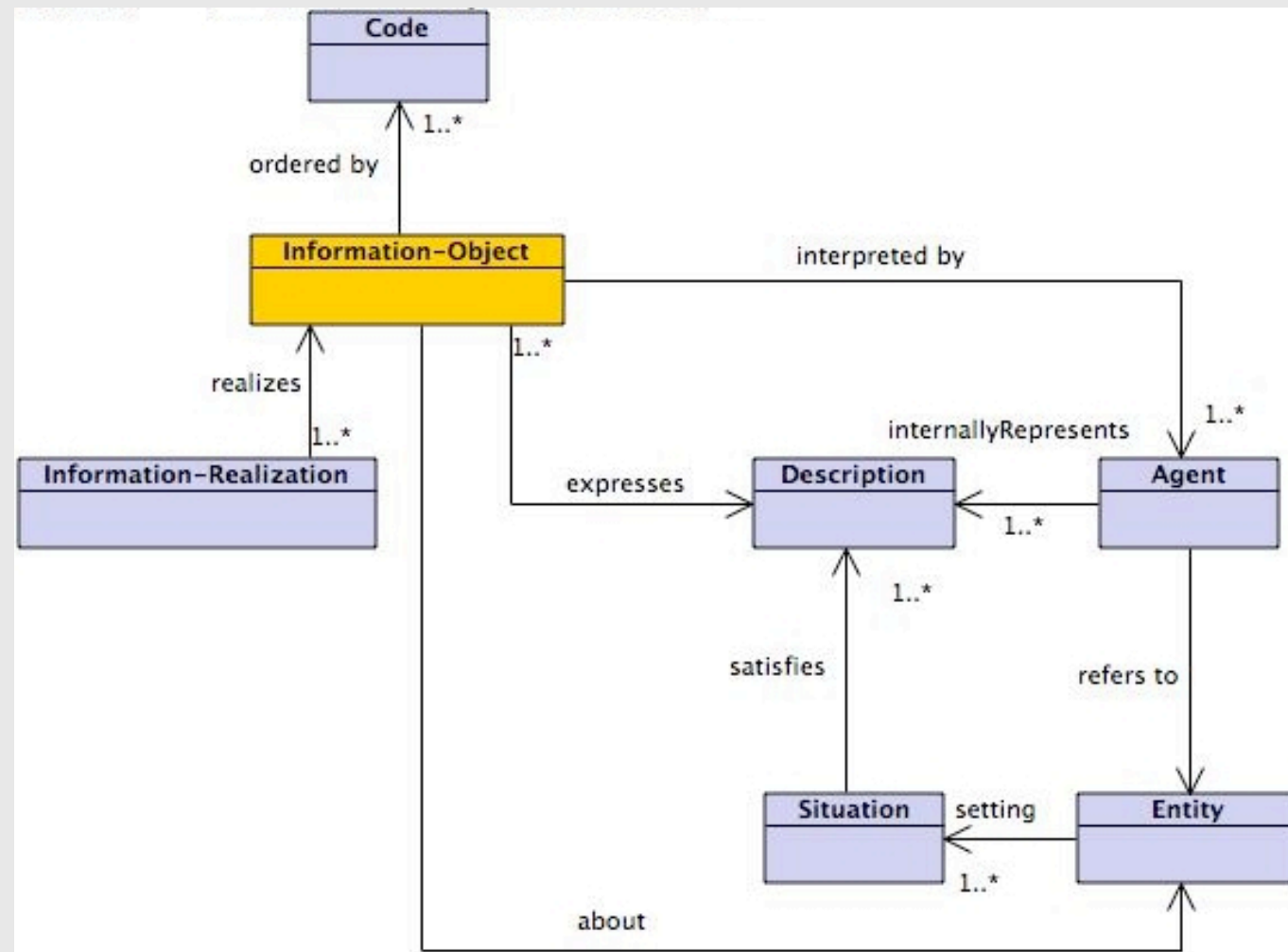
cf. <http://www.loa-cnr.it/ontologies/Systems.owl>

Examples:

- I would be told it's an original design, and not supposed to represent any actual car
- Assigning an early role to the instructional design function (internal or external) can save many - headaches, dollars, and delays down that path
- The design of the product model is very difficult since it must be adaptable to various phases of the design and manufacturing procedures



Information objects and realizations



cf. <http://www.loa-cnr.it/ontologies/InformationObjects.owl>

Examples:

- This book contains useful information about writing CVs
- A more recent edition of this book is available from Amazon



Specializing patterns

- *Same structure down the taxonomy hierarchy*
- A CODEP p2 specializes another p1 when at least one of the classes or properties from p2 is a sub-class or a sub-property of some class resp. property from p1, while the remainder of the CODEP is identical.
- *Participation*
 - *Taking part in a public enterprise*
 - *Giving a grant to a Semantic Web project*
- *Co-participation*
 - *Having a social relationship*
 - *Being bunkmates*



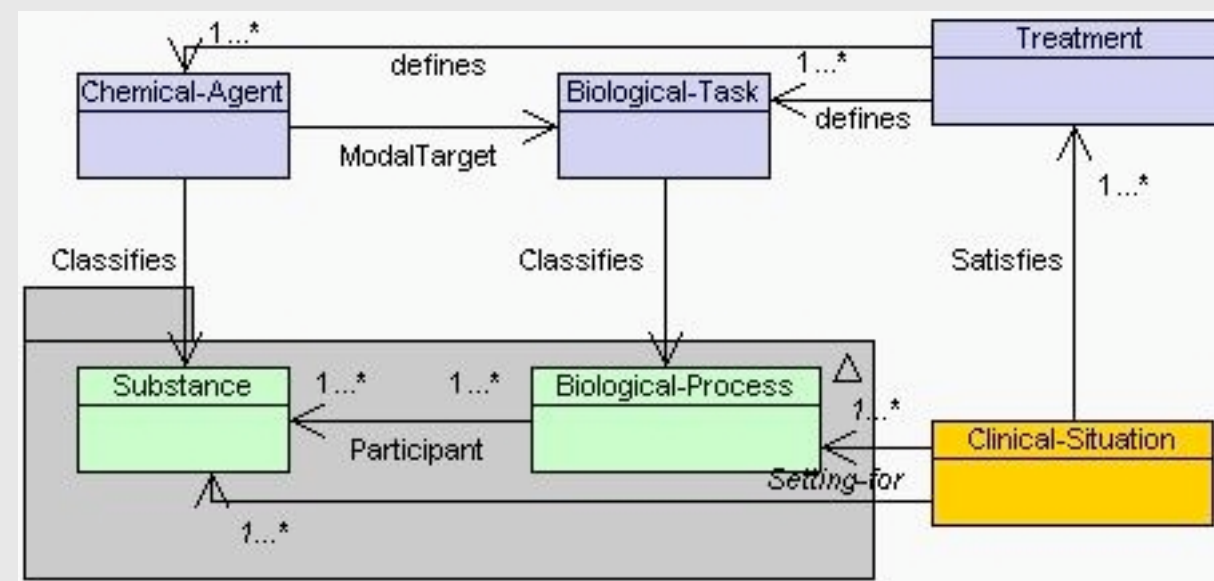
Composing patterns

- *Linking sensible classes on the background of a common (or integrated) reference ontology*
- A CODEP p2 expands p1 when p2 contains p1, while adding some other class, property, or axiom.
- A CODEP p3 integrates p1 and p2 when p3 contains both p1 and p2.
- A CODEP p3 merges p1 and p2 when p3 contains both p1 and p2, and there exist explicit links between at least two classes or properties of p1 resp. p2.
- Biochemical_Treatment → (Role<->Task ° Description<->Situation ° Substance<->Agent ° Time-indexed_participation)



The Biochemical Treatment pattern

Composes:
Role<->Task,
Description<->Situation,
Substance<->Agent
Time-Indexed-Participation

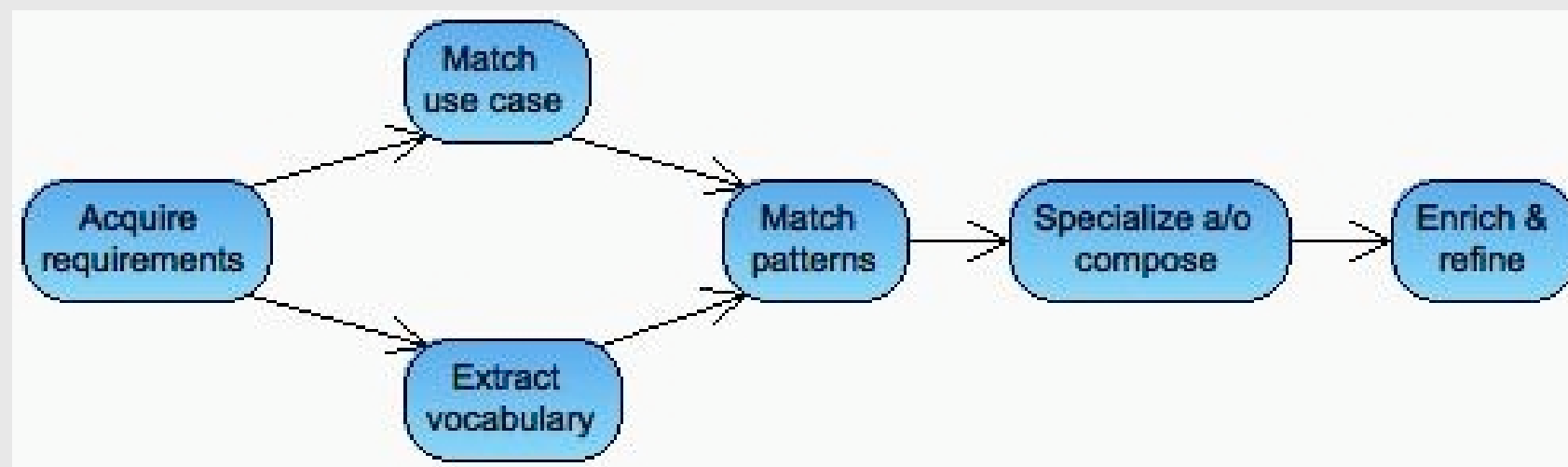


Example: One chemotherapeutic approach is the development of selectively toxic substances, i.e., substances that can destroy or inhibit infecting organisms



Minimal methodology

In ontology services, a GUC/LUC repository can help an agent (either human or artificial) to find the best patterns for a modelling requirement (either expressed in natural language or in a database or informal schema), with respect to expected services. Further services for pattern composition and specialization assist agents to draft the first detailed version of an ontology project.

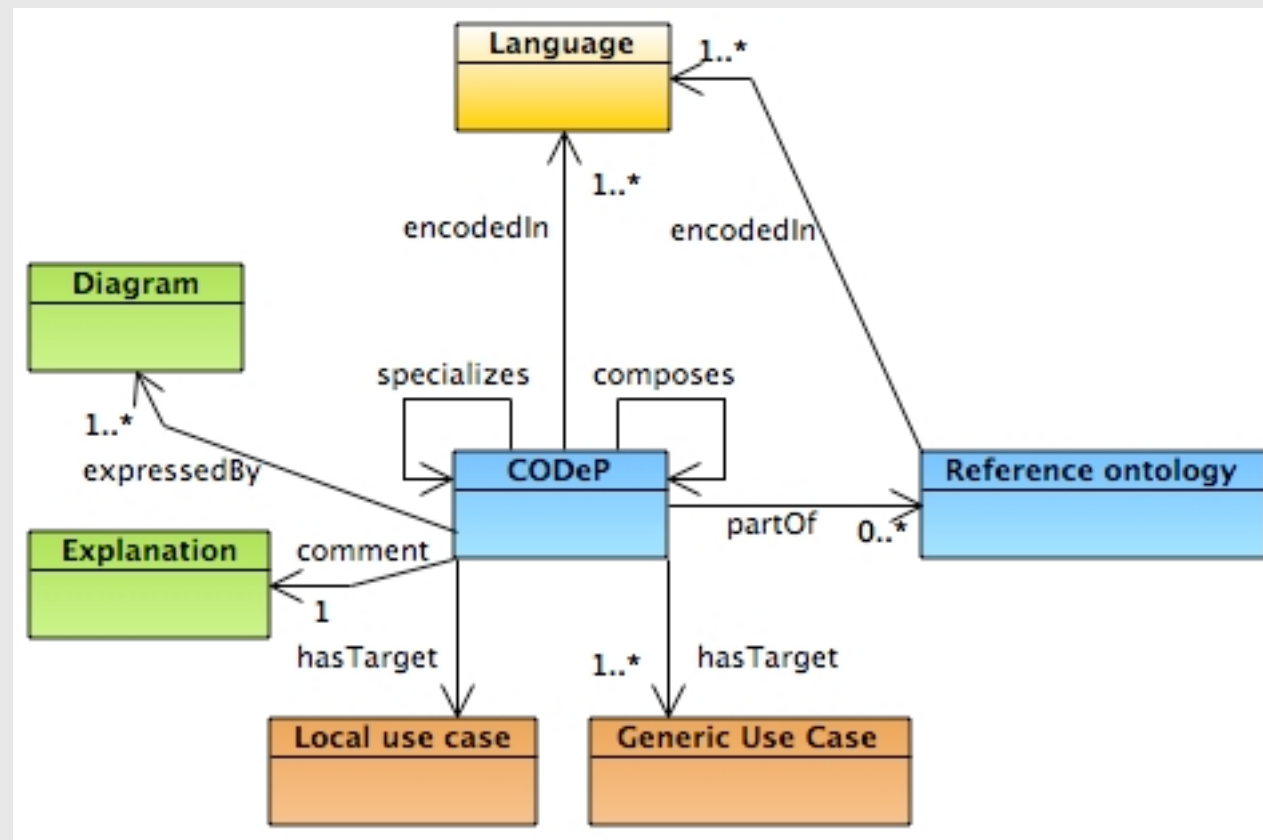



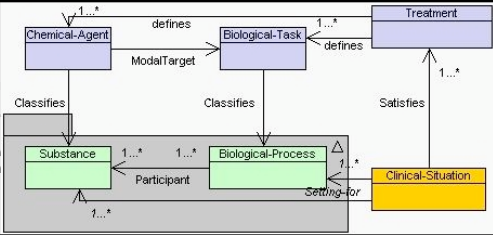
Summary

- Differences between ontology patterns and other patterns
- A use case registry for *content* ontology design patterns (CODEP)
- Examples of, and operations on CODEPs
- **How to frame CODEPs (a meta-model)**



The CDeP meta-model



Slot	Value
Generic use case (GUC)	Substances playing roles in processes.
Local use case(s)	<p>Various chemical agents, mostly drugs, are used to control biological processes within a chemotherapeutical treatment.</p> <p>When talking about drugs and processes, there is a network of senses implying a dependence on roles and functions (or tasks) within a clinical treatment.</p> <p>Intended meanings include the <i>possible</i> roles played by certain substances, as well as the actual administration of amounts of drugs for controlling actually occurring biological processes. Therefore, both class- and instance-variables are present in the maximal relation for this pattern.</p>
Logic addressed	OWL, DL species
Reference ontologies	DOLCE-Lite-Plus, NCI Ontology
Specialized CODEP	Role<->Task
Composed CODEPs	Concept<->Description, Description<->Situation, Substance<->Agent, Time-Indexed-Participation
Formal relation	$r_{\text{Chemical_or_Drug_Plays_Role_in_Biological_Process}}(\varphi, \psi, x, y, t, c_1, c_2, d, s)$, where $\varphi(x)$ is a chemical agent class, $\psi(y)$ is a biological process class, t is a time interval, c_1 and c_2 are two reified intensional concepts, d is a reified intensional relation, and s is a reified extensional relation.
Sensitive axioms	$r_{\text{Chemical_or_Drug_Plays_Role_in_Biological_Process}}(\varphi, \psi) =_{\text{df}} \forall x, y, t ((\varphi(x) \wedge \psi(y) \wedge \text{participant-in}(x, y, t) \wedge \text{Chemical-Agent}(x) \wedge \text{Biological-Process}(y) \wedge \text{Time-Interval}(t))$ $\leftrightarrow \exists c_1, c_2, d (CF(x, c_1, t) \wedge MT(c_1, c_2) \wedge CF(y, c_2, t) \wedge DF(d, c_1) \wedge DF(d, c_2) \wedge \forall s (\text{SAT}(s, d) \leftrightarrow (\text{SETF}(s, x) \wedge \text{SETF}(s, y) \wedge \text{SETF}(s, t))))$
Explanation	<p>Since OWL(DL) does not support relations with >2 arity, reification is required. The Role<->Task and the Description<->Situation patterns provide typing for such reification. Since OWL(DL) does not support classes in variable position, we need reification for class-variables. The Role<->Task pattern provides typing for such reification. Similarly, since participation is time-indexed, we need the time-indexed-participation pattern, which is here composed with the previous two patterns (time indexing appears in the setting of the general treatment situation). Finally, the Substance<->Agent pattern provides a specialization of the Role pattern usable for the use case.</p>
OWL(DL) encoding (abstract syntax)	<p>Class(Chemical_Plays_Role_in_Bio_Process complete</p> <p>Description</p> <p>restriction(defines someValuesFrom(Chemical-Agent))</p> <p>restriction(defines someValuesFrom(Biological-Task)))</p> <p>Class(Chemical-Agent complete</p> <p>Role</p> <p>restriction(defined-by someValusFrom(Chemical_Plays_Role_in_Bio_Process))</p> <p>restriction(classifies allValuesFrom(Substance))</p> <p>restriction(modal-target someValuesFrom(Biological-Task)))</p> <p>Class(Biological-Task complete</p> <p>Task</p> <p>restriction(classifies allValuesFrom(Biological-Process))</p> <p>restriction(modal-target-of someValuesFrom(Chemical-Agent)))</p> <p>Class(Chemical-in-Biological-Process_Situation complete</p> <p>Situation</p> <p>restriction(satisfies someValuesFrom(Chemical_Plays_Role_in_Bio_Process))</p> <p>restriction(setting-for someValuesFrom(Substance))</p> <p>restriction(setting-for someValuesFrom(Biological-Process))</p> <p>restriction(setting-for someValuesFrom(Time-Interval)))</p>
Class diagram	<div>  <div>ODP Tutorial</div> </div> 

Implementation

- While the meta-model remains an acceptable requirement model, ontology patterns and use cases can be specified and implemented in different ways, e.g.:
 - As ontologies (current solution, applies 'skins' in case of RO-dependency)
 - As modules or partitions of RO (still unclear semantics; owl:imports is an issue for readability and complexity)
 - As meta-models annotating ontologies (owl:restriction is an issue)
 - As unit-tests (ongoing work)
 - As D&S ontologies, similarly to OntoFrameNet ontology (massive reification is an issue, but most patterns are equivalent to axiom schemas for >2-ary relations)
- **How to link CODEPs to Use Cases?**
 - By generic properties between registry entries
 - By meta-models annotating CODEP elements to Use Case elements
 - Via D&S descriptions, by matching ABox structures, e.g. similarly to Q/A



Applications of content ontology design patterns

- Ontology development by pattern specialization and composition
- Migration of informal schemas to ontologies
- Heuristics for ontology learning and Q/A
- Refinement of extracted patterns (e.g. from ontology learning)
- Paraphrase synonymy discovery and relation learning
- Ontology mapping (macro-matching), e.g. by generating bridging rules clusters
- Ontology evaluation (identification of hub nodes and their accuracy wrt to domain/task)
- Service discovery and composition
- Education



Ongoing/related work and open issues

- Optimal CODEP encoding for the SW
- Experiments to evaluate CODEPs (cf. new oQual evaluation methodology)
 - Relation learning, social networks, user feedback (*KnowledgeZone* - Stanford Un.), cognitive evaluation, OntoFrameNet
- Using CODEPs for evaluating ontologies wrt use cases
 - Representing CODEPs as unit tests for ontologies
- Tools to optimize CODEP matching (e.g. via clustering) against a GUC/LUC registry (taxonomy) and informal requirements
- Tools to manage CODEP specialization and composition
- Deploying CODEP properties to optimize reasoning
- Detailed comparison to PSM paradigm
- Follow the *NeOn FP6 Integrated Project*: <http://www.neon-project.org>

