Design patterns for legal ontology construction

Aldo Gangemi
Laboratory for Applied Ontology, ISTC-CNR, Roma, Italy
aldo.gangemi@istc.cnr.it

Abstract. Ontology design is known to be a difficult task, requiring much more than expertise in an area; *legal* ontology design, due to the complexity of its domain, makes those difficulties worse. That may be partly due to poor requirement analysis in existing tools, but there is also an inherent gap between the purely logical constructs and methods that are expected to be used, and the actual competences and thought habits of domain experts. This paper presents some solutions, based on *content ontology design patterns*, which are intended to make life of legal ontology designers easier. An overview of the typical tasks and services for legal knowledge is presented, the notion of ontology design pattern is introduced, and some excerpts of a reference ontology (CLO) and its related patterns are included, showing their utility in a simple legal modeling case

Keywords: legal knowledge engineering, ontology design, design pattern

1. Introduction

A new breed of semantically-explicit applications is getting momentum through the Semantic Web programme and beyond. Legal practice can take advantage from them e.g. in the form of dynamically integrated Semantic Web Services (SWS) (Motta et al., 2003), directed towards citizens, institutions, and companies.

The core of semantically-explicit applications is constituted by so-called *ontologies*, which are strongly-typed logical theories that formalize the assumptions underlying various kinds of knowledge, including physical and social objects, as well as legal procedures, norms, roles, contracts, etc. Ontologies are usually expressed in first-order languages or fragments of them, although some typical modal and meta-level primitives are usually added to them, e.g. in *description logics* like OWL(DL) (McGuinness and van Harmelen (editors), 2004).

Ontologies can be designed by means of various methodologies (e.g. (Gruninger et al., 1994)(Gangemi et al., 2004)), encompassing top-down expertise elicitation from humans, bottom-up learning from documents, and middle-out application of content patterns (specialized from domain-independent ontologies, elicited in a top-down way, or learnt from patterns found in experts' documents), which can be called *Content Ontology Design Patterns* (CODeP, also known as "conceptual" ontology design patterns) (Gangemi et al., 2005). In large-scale, realistic applications, CODePs are core components for ontology design.

The legal domain is very complex compared to others, because it involves knowledge of the physical and social worlds, as well as typical legal knowledge that actually creates a novel layer over the social world (Moore, 2002).
Due to the autonomy (on one hand) and dependence (on the other hand) of the legal knowledge on both physical and social knowledge, legal reasoning tasks have evolved in a peculiar way, which include e.g. the norm structure based on CODePs like *Requirement→Consequence* (if the factual knowledge is P, then the legal knowledge is Q), *Obligation→Right* (if A has an obligation towards B, then B has a right towards A), *Norm↔Case* (if a situation fulfils a the conditions for violating a norm, it becomes a legal case), *CrimeScenario* (a crime is committed by a perpetrator and comes to the attention of authorities that pursue a criminal process), etc. The CODePs that are assumed by legal experts can be formalized by specializing or composing other existing patterns for the social world.

This work introduces some use cases for legal ontologies, as well as some CODePs that can be specialized to support ontology-driven solutions to those use cases. The CODePs have been defined on top of the DOLCE foundational ontology library (Masolo et al., 2004)(DOLCE, -), the Core Legal Ontology (CLO) (Gangemi, Sagri and Tiscornia, 2004) (CLO, -), and JurWordNet (Sagri, 2003).

In section 1. some ontology design/engineering use cases in the legal domain are introduced. In section 2. the CODeP idea is presented. In section 3. The Core Legal Ontology is briefly summarized and some more legal CODePs sketched with an example on a use case.

2. Legal Ontology Engineering: Functionalities and Techniques

Within ICT, ontology design is dependent on (ontology) engineering applications, which involve the statement of functionalities and their implementation as techniques and tools. For a comprehensive framework of ontology design, and its relations to content, related data, formal languages, design patterns, social practices, organizations, teams, and functionalities, see (Gangemi et al., 2007). The ontology encoding of a metamodel for describing collaborative ontology design activities and data can be found at: `http://www.loa-cnr.it/ontologies/OD/codolib.owl`.

Ontology engineering deals with designing, managing, and exploiting ontologies (to be intended as *strongly-typed logical theories*) within information systems. Ontologies are usually hybridated with other components in order to build semantically-explicit applications; e.g., when used jointly with:

- theorem provers, *consistency checking* can be performed to logically validate the set of assumptions encoded in an ontology
- subsumption and instance classifiers against a logical language of known and manageable complexity, like OWL (in the Lite and DL species), *automatic inferences* can be derived from *taxonomical reasoning* as well as for the *classification of instances and facts* (Gangemi et al., 2001) (Gangemi, Sagri and Tiscornia, 2004)
- computational lexicons, NLP tools, and machine-learning algorithms, legal ontologies can enhance *information extraction* from semi-structured and non-structured data, adding a new dimension to knowledge management and discovery in Law (Gilardoni et al., 2005)
- planning algorithms, ontologies can assist or automatize *negotiation* or *execution* e.g. for *contracts, regulations*, *services*, etc. (Gil et al., 2005)
- case-based reasoners, ontologies can *formalize case abstractions* within more general frameworks, or can *classify cases* according to pre-designed descriptions (Forbus et al., 2002)
- rule-based engines, facts can be inferred e.g. for *causal responsibility assessment*, *conformity checking, conflict detection* and in general for *fact composition* (Gangemi et al., 2001).

Ontology engineering techniques are exploited in the context of "generic use cases" defined for a domain of application. The main types of use cases that can be implemented or assisted by means of semantically-explicit applications in the legal domain (Fig. 1) are summarized in the following.

Intersubjective agreement and meaning negotiation
Definition: the task of getting consensus (or of discovering disagreement) about the intended meaning of a legal term, legal text unit, etc.
Approach: the formal encoding of (part of) the intended meaning assumed by each of the parties involved in the task.
Issues: given the traditional practices of consensus reaching in Law,

this task is usually considered intrusive, and could require a mindset shifting in order to acquire some relevance. Nonetheless, encoding intended meaning of a legal text is preliminary to all other tasks presented here. This observation seems paradoxical, since, if we cannot consider the formal encoding of legal meaning as an interpretation with legal validity, all the other tasks result to be based on an arbitrary (in the worst case) or a weak (in the best case) set of assumptions. Due to the current state of legal ontology, most tasks are carried out *as if* that formal encoding had legal validity, thus providing results that can be considered only as heuristical means for legal professionals or citizens.

Knowledge extraction
Definition: the process of extracting concepts, relations, named entities, and complex knowledge patterns from a database, a document, or a corpus.
Approach: data- and text-mining, machine-learning, and NLP algorithms that can extract linguistic objects from a corpus, and semi-automated methods that match them to semantic objects.
Issues: this task is highly incremental, because the approaches need a *training* phase or an extensive *data entry* procedure, so that the extracted knowledge can be used to build a repository of patterns that can be used to improve further extraction processes. Best results can be achieved on very large corpora (for statistical reasons), or on well-delimited, possibly semi-structured corpora and tasks; for example, typical expressions that are found in legal drafting can be used to formalize expectation patterns in corpora consisting of homogeneous texts (Basili et al., 2005).

Conformity checking
Definition: the task aimed at verifying if a social situation (known in some way compliant with legal regulations) satisfies a legal description (norm, principle, regulation, etc.). In the generalized case, also situations already known to be legally relevant for some reason (e.g. a crime situation) can be checked for conformity against a further legal description (e.g. an appeal judgment procedure).
Approach: the representation of social or legal situations as well as of legal regulations. Reasoners to cluster/classify/abstract situations.
Issues: Reasoning: the typical inferences supported by semantic web engines for OWL (McGuinness and van Harmelen (editors), 2004) (e.g. FaCT++ (Horrocks,)) currently include only *concept subsumption* and *instance classification*. The expressivity is also limited so that e.g. *fact classification* (also called *materialization*) is only performed by some engines (e.g. Pellet (Sirin et al., 2007)). Moreover, *compositions bet-*

ween facts cannot be inferred unless an additional rule engine is added (extensions for rule languages supporting fact composition are provided by additional languages: SWRL(Horrocks et al., 2005), SPARQL (Prud'hommeaux et al., 2005), F-Logic (Hustadt et al., 2004), and their related implementations, e.g. KAON2 (Hustadt et al., 2004). Moreover, clustering and abstraction of situations requires different reasoners, e.g. induction engines (Basili et al., 2005) and case-based reasoners (Forbus et al., 2002). Finally, approximate inferences (Domingos and Richardson, 2004) should also be supported in the generalized case of partial knowledge about situations.
Representation: a homogeneous language to represent both situations and the constraints from a legal description is highly desirable, otherwise a higher-order logic would be required to express constraints on contraints on constraints etc. on situations. The proposal in (Gangemi, Sagri and Tiscornia, 2004), briefly summarized in next sections, shows a viable approach to represent both constraints and instance data in a same, partitioned first-order domain of quantification.

Legal advice
Definition: the investigation of the relations between legal cases and common sense situations.
Approach: subsumption/instantiation classification, or case-based reasoning.
Issues: in large scale applications, legal advice involves crucial problems such as causality and responsibility assessment, open-textured concepts, interpretation aspects, which are still being investigated from an ontological perspective. A typical scope for legal ontology design is to encode only *weak constraints* for terminological clarification. Legal advice requires more than that.

Norm comparison
Definition: the matchmaking between different norms. Norm comparison includes tasks such as: (i) *normative conflict checking* and handling between norms about a same situation type, (ii) *discovery of implicit relations* between a norm and other norms from a known corpus.
Approach: approximate classification algorithms (i-ii), including legal text annotation and classification on large corpora (ii). (Gangemi, Sagri and Tiscornia, 2004) and (Gangemi et al., 2001) show simpler approaches to the classification of norm dynamics and conflicts within a finite set of norms after their first-order encoding.

Issues: in Civil Law corpora, the task (ii) is sometimes relevant as much as in Common Law corpora, because of the stratification of laws that do not explicitly delete or even refer to previous ones (e.g.

in Italy). In Common Law, implicit relations can be discovered more easily, because case abstraction has always a clear reference to a case, while in Civil Law, implicit relations appear at the purely normative level.

Norm rephrasing

Definition: expression of norms' content in different terms, which can be either translations in a different natural language, or in a different form within a same natural language, e.g. for the purpose of popularization.
Approach: translation between different languages requires a preliminary mapping between terms, like the EuroWordNet-oriented work performed in the LOIS project (Peters et al, 2006), classifications based on statistical NLP techniques, and subsumption classification for a close matching between content patterns and linguistic patterns.

Contract management and execution

Definition: a service assisting parties in the tasks of managing contract agreement and definition, and of following contract execution.
Approach: the semantic specification of contract content, as well as algorithms to manage the matching of parties' constraints and preferences, and a planning algorithm for the generation of optimal obligations that parties could undertake (Gil et al., 2005).

(Information) service matchmaking and composition

Definition: operations carried on the description of services, in order to check e.g. if an offered service matches the requested service, or to orchestrate two services to get a more complex one.
Approach: in the legal domain, these tasks require the semantic specification of services with reference to the legal knowledge involved in the execution of the service. Appropriate reasoners and planners are required.

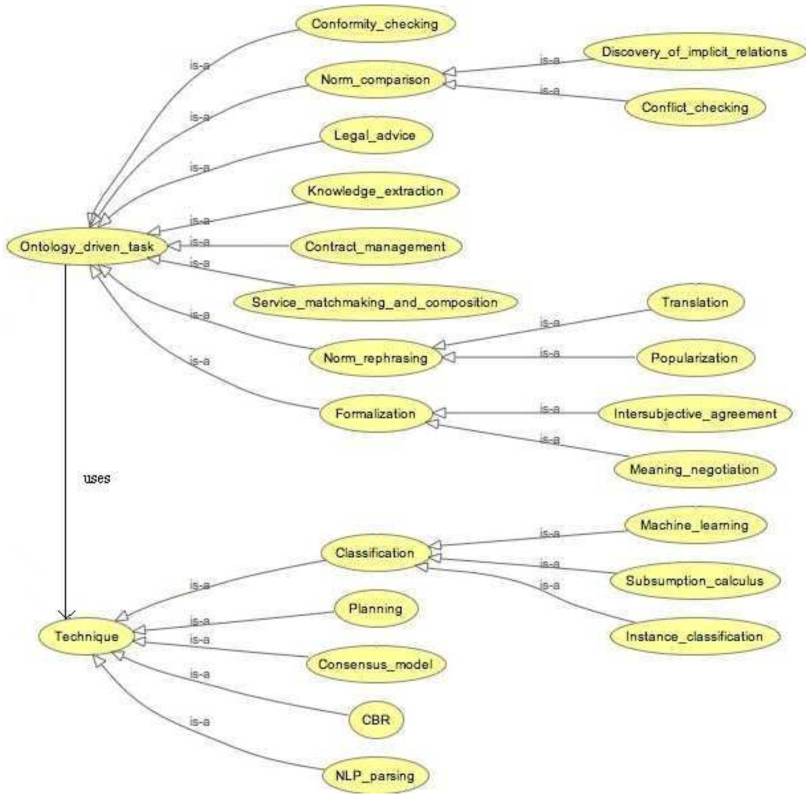


Figure 1. A taxonomy of ontology-driven tasks and related techniques for legal information

3. Content Ontology Design Patterns

Semantically-explicit applications in the legal domain present us with conceptual analysis and integration problems that require appropriately designed legal formal ontologies. Part of the design problems can be simplified by creating or extracting "Conceptual Ontology Design Patterns" (CODeP) for a domain of application (Gangemi, Sagri and Tiscornia, 2004) (Gangemi et al., 2004). An intuitive characterization of CODePs is provided here:

– A CODeP is a template to represent, and possibly solve, a modelling problem. For example, a $Norm \leftrightarrow Case$ CODeP (Fig.3) facilitates the modelling of legal norms and cases (as well as their components and dependencies) in logical languages that require constraint reification. E.g. in OWL(DL), relations with an arity $=2$ are not allowed, therefore OWL(DL) modelling requires a reification of those relations. A vocabulary for reification has been designed in the Descriptions and Situations ontology ("ExtendedDnS" in Fig. 4, see below), which is specialized in the $Norm \leftrightarrow Case$ pattern.

- A CODeP "extracts" a *connected fragment* of a reference ontology, which constitutes its "background". For example, a connected path of two relations and three classes (A(x) ∋ B(y) ∋ C(z) ∋ R(x,y) ∋ S(y,z)) can be extracted as a sub-theory of an ontology O because of its relevance in a domain. Therefore, a CODeP lives in a reference ontology, which provides its taxonomic and axiomatic context. E.g. in the $Norm \leftrightarrow Case$ CODeP a foundational distinction is reused from DOLCE (Masolo et al., 2004), while the cardinalities for the relations are provided by the Core Legal Ontology (CLO, (Gangemi, Sagri and Tiscornia, 2004) (CLO, -)). DOLCE and CLO together form the reference ontology for the CODeP.

- Mapping and composition of patterns require a reference ontology, in order to check the consistency of the composition, or to compare the sets of axioms that are to be mapped. Operations on CODePs depend on operations on the reference ontologies. However, for a pattern user, these operations should be (almost) invisible.

- A CODeP can be represented in an ontology representation language whatsoever (depending on its reference ontology), but its intuitive and compact visualization is an essential requirement. It requires a critical size, so that its diagrammatical visulization is aesthetically acceptable and easily memorizable. For example, the $Norm \leftrightarrow Case$ CODeP only includes eight classes, with several, systematic relations between them: this makes it *dense*, but *manageable*.

- 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 pattern is specialized. A hierarchy of CODePs can be built by specializing or generalizing some of the elements (either classes or relations). For example, the *participation* pattern (of an object in an event) can be specialized to the *taking part in a public enterprise* pattern (of an agent in a social activity with public relevance).

- 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 and best practices that "make rational thinking move" for an expert in a given domain for a given task.

- A CODeP can be often built from informal or simplified schemata used by domain experts, together with the support of other reusable CODePs or reference ontologies, and a methodology for domain ontology analysis. Typically, experts spontaneously develop schemata to improve their business, and to store relevant know-how. These schemata can be reengineered with appropriate methods (e.g. (Gangemi et al., 2004)).
- A CODeP can be similar to a database schema, but a pattern is defined wrt to a reference ontology, and has a general character, independently of system design. In this sense, it is closer to so-called *data modelling patterns* (Hay, 1996), but a CODeP should be contextualized in a reference ontology, making it more interoperable than a data modelling pattern, at least in principle.

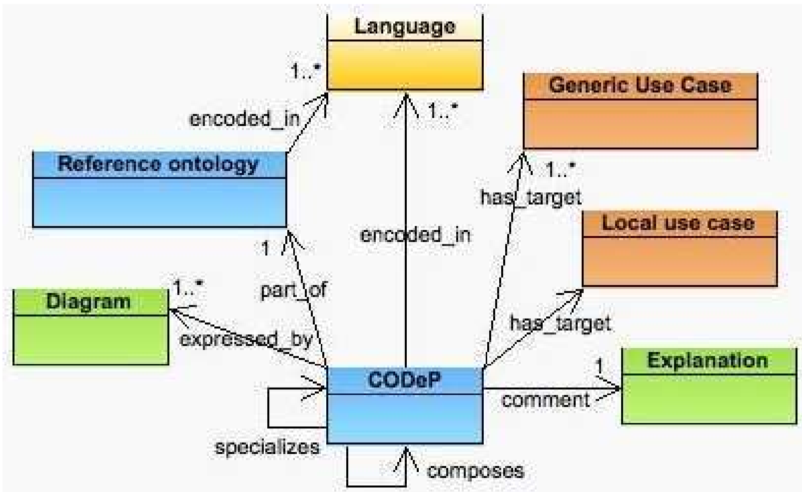


Figure 2. The CODeP annotation pattern

Conceptual Ontology Design Patterns (CODePs) are a resource and design method for engineering ontology content over the Semantic Web.

A template (Fig. 2, also available in OWL from (CODeP,)) can be used to annotate CODePs as sub-theories of reference ontologies, in order to share them in pre-formatted documents, as well as to describe, visualize, and make operations over them them appropriately.

The CODeP template consists of:

- Two slots for the *generic use case*, and the *local use cases*, which includes a description of context, problem, and constraints/requirements.

– Two slots for the addressed *logic*, and the *reference ontologies* used as a background for the pattern.

– Two slots for -if any- the *specialized* pattern and the *composed* patterns (by inheritance, and inverse inheritance, it's possible to obtain the closure of specialized and expanding patterns).

– Two slots for the *maximal relation* that encodes the case space, and its intended *axiomatization*: a full first-order logic with meta-level is assumed here, but the slot can be empty without affecting the functionality of a CODeP frame.

– Two slots for *explanation* of the approach, and its *encoding* in the logic of choice.

– A last slot for a *class diagram* that visually reproduces the approach.

The template can be easily encoded in XSD or in richer frameworks, like semantic web services (e.g. Motta et al. 2003) or knowledge content objects (Behrendt et al. 2005), for optimal exploitation within Semantic Web technologies. The high reusability of CODePs and their formal and pragmatic nature make them suitable not only for isolated ontology engineering practices, but ideally in distributed, collaborative environments like intranets, the Web or the Grid.

CODePs can also be used to generate intuitive, friendly UIs, which can present the user with only the relevant pattern diagram, avoiding the awkward, entangled graphs currently visualized for medium-to-large ontologies.

The advantages of CODePs for ontology lifecycle over the Semantic Web are straightforward: firstly, patterns make ontology design easier for both knowledge engineers and domain experts (imagine having a menu of pre-built, formally consistent components, pro-actively suggested to the modeller); secondly, patterned design makes it easier ontology mapping - perhaps the most difficult problem in ontology engineering. For example, the *time-indexed participation* presented in this paper requires non-trivial knowledge engineering ability to be optimally represented and adapted to a use case: a CODeP within an appropriate ontology management tool can greatly facilitate such representation.

The CODeP examples and the related frame and methods introduced in this paper have been applied for two years (some of them even before) in several administration, business and industrial projects, e.g. in fishery information systems (Gangemi et al., 2004), insurance CRM, biomedical ontology integration (Gangemi et al., 2004), anti-money-laundering systems for banks (Gangemi et al., 2001), service-level agreements for information systems, biomolecular ontology learning (Ciaramita et al 2005), legal norms formalization (Gangemi, Sagri and Tiscornia, 2004)(Sagri et al., 2004).

Current work focuses on building a tool that assists development, discussion, retrieval, and interchange of CODePs over the Semantic Web, and towards establishing the model-theoretical and operational foundations of CODeP manipulation and reasoning. In particular, for CODePs to be a real advantage in ontology lifecycle, the following functionalities should be available:

– Categorization of CODePs, based either on the use cases they support, or on the concepts they encode.

– Pattern-matching algorithms for retrieving the CODeP that best fits a set of requirements, e.g. from a natural language specification, or from a draft ontology.

– Support for specialization and composition of CODePs. A CODeP 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 resp. property from p_1, while the remainder of the CODeP is identical. A CODeP p_2 *expands* p_1 when p_2 contains p_1, while adding some other class, property, or axiom. A CODeP p_3 *composes* p_1 and p_2 when p_3 contains both p_1 and p_2. The formal semantics of these operations is ensured by the underlying (reference) ontology for the patterns. Notice that CODePs –differently from "knowledge patterns" in (Clark et al., 2000), which are characterized as *invariant under signature transformation*– are intended to be *downward conservative under signature transformation*, meaning that a pattern semantics is structure-preserving when the pattern is *specialized, expanded,* or *composed*, but this conservativeness holds only in the downward taxonomical ordering. On the contrary, *logical* ontology design patterns are conservative under signature transformation both down- and up-wardly.

– Interfacing of CODePs for visualization, discussion, and knowledge-base creation.

– A rich set of metadata for CODeP manipulation and exploitation within applications.

4. The Core Legal Ontology and its Related Patterns

The need for an extended typology of legal entities is becoming a pressure, even from traditionally “bottom-up” approaches. For example, the need to pair case-based reasoning with an ontology of first-principles should be investigated in order to represent the two kinds of structures employed in reasoning: abstraction from cases, and satisfaction of constraint sets (e.g. norms) (Forbus et al., 2002).

The level of granularity is also a core issue in developing formal ontologies, specially because a decentralized architecture is emerging for ontologies as well: how to compare/integrate/transform two ontologies about a close domain, but with a different detail encoded in their vocabulary and axioms?
The Core Legal Ontology (CLO) (Gangemi, Sagri and Tiscornia, 2004) (CLO, -) is developed on top (Fig.3) of *DOLCE* (Masolo et al., 2004) and *Descriptions and Situations* (Gangemi and Mika, 2003) (Masolo et al., 2004) ontologies within the DOLCE+ library (DOLCE, -). CLO allows for the representation of first principles (by means of a rich axiomatization), and granularity (by means of its reification vocabulary and axioms) in the legal domain.

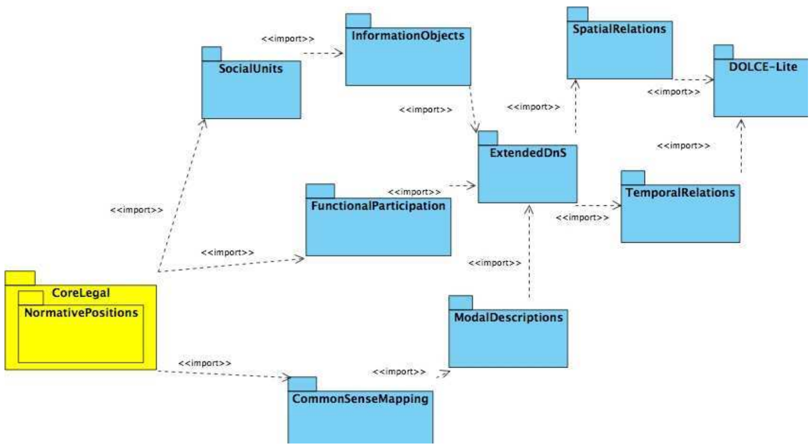


Figure 3. CLO depends on other ontologies: DOLCE, ExtendedDnS, InformationObjects, Temporal and Spatial relations, etc. The *Norm↔Case* CODeP is a component within the CoreLegal (CLO) module.

The two pillars of CLO as a plugin to DOLCE+ are: *stratification* and *reification*.

Based on the stratification principle, CLO provides types and relation for the heterogeneous entities from the legal domain, be it about the physical, cognitive, social, or properly legal worlds (cf. (Moore,

2002) (Gangemi, Sagri and Tiscornia, 2004)). According to stratification, entities from different layers can be *spatio-temporally co-located*, yet being completely different and (mutually or one-way) dependent. For example, a physical person pertains to the physical world as a biological organism, but the properties of the organism are not sufficient to characterize it as a social person. On its turn, the properties of a social person are not sufficient to characterize it as a legal person. Clearly, there are dependencies among those properties, but if the properties from each layer are simply summed up in a same entity, an ontology designer can get undesirable results, e.g. it would be possible to infer that an organism (physical layer) can be "acting" after its death because of the legal existence (legal layer) of a person until its legal effects disappear.

In DOLCE, the solution includes a very general pattern, expressed as a disjointness axiom between the class of physical vs. social objects, whereas legally-relevant entities are mostly in the social realm (see Fig.s 5,6,7,8,9, which also include some use of CLO for the JurWordNet lexical ontology (Sagri, 2003) (Gangemi, Sagri and Tiscornia, 2004)). Among social objects, *agentive* and *non-agentive* are disjoint, and among non-agentive ones, *legal descriptions, concepts, facts, collectives, persons*, and *information objects* are also disjoint. Among legal descriptions, *constitutive* vs. *regulative* descriptions are distinguished from *principles, rationales, modal descriptions* (e.g. *duties, powers, liabilities,* etc.), as well as mixed regulations such as contracts and bundles of norms. Among legal facts, *natural, human, cognitive,* and strictly *legal cases* are distinguished. Legally-relevant *circumstances* are further distinguished from legal facts as being ancillary to primary facts. Among legal persons, *organizations*, *natural persons*, and *legal subjects* are also distinguished.

Based on the reification principle, CLO enables an ontology engineer to quantify either on legal rules or relations (type reification) (Masolo et al., 2004) (Galton, 1991), or on legal facts (token reification) (Gangemi and Mika, 2003) (Galton, 1991). CLO extends the *Descriptions and Situations* vocabulary for reification. For example, *intensional specifications* like norms, contracts, subjects, and normative texts can be represented in the same domain as their *extensional realizations* like cases, contract executions, agents, physical documents.

A reification-based pattern models the structure of an intensional specification, called *Description* in (Masolo et al., 2004), as composed of its concepts and their internal dependencies. For example, the structure of a norm (a *legal description*) employs that pattern. Another pattern models the matching between a description and its extensional realization, called a *Situation* in (Gangemi and Mika, 2003), which can

be described as the configuration of a set of entities according to the structure of a description. A legal application of this pattern can be found in the dependencies among the *rules in a contract*, when they can be matched to a *legal case* (a legal situation or fact) or a *contract execution*. The matching is typically performed when checking if each entity in a legal fact is compliant to a concept in a legal description.
CLO is currently used to support the definition of legal domain ontologies (Sagri et al., 2004), the definition of a juridical wordnet (Sagri, 2003), and the design of legal semantic web services.

In order to describe some of the ontological expressiveness of CLO, a complex CODeP is introduce here which has CLO as its reference ontology. This is is the $Norm \leftrightarrow Case$ CODeP (Fig. 4): it is used as a pattern for representing legal cases, is specialized for types of norms and cases, e.g. for crime investigation, and is composed with other patterns, e.g. for norm conflict checking.

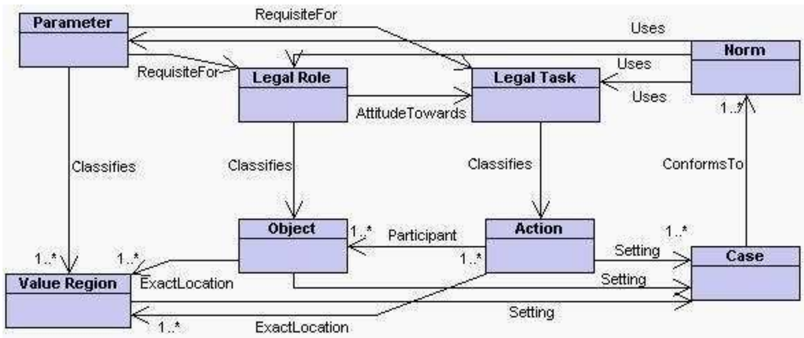
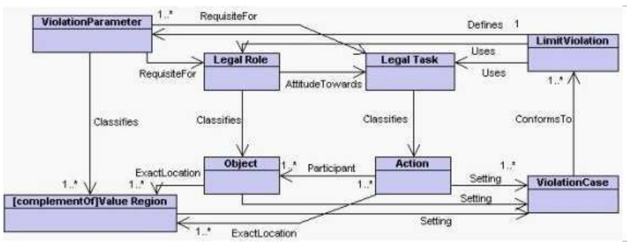


Figure 4. The $Norm \leftrightarrow Case$ CODeP: norms *use* tasks, roles, and parameters; legal cases *conform* to norms when actions, objects and values are *classified* by tasks, roles, and parameters respectively. Moreover, relations between legal roles, tasks and parameters correspond to relations between objects, actions and values. For example, an *obligation* for a role towards a task should correspond to a *participation* of an agent (object) in an action; a spatial parameter that is *requisite* for an object should correspond to an *exact location* of an object in a spatial value region that is *classified* by that parameter.

In the following template (Tab. I, see Fig. 2 for its datamodel), the $Limit \leftrightarrow ViolationCase$ CODeP is compactly introduced as a specialization of the generic $Norm \leftrightarrow Case$ CODeP:

Table I.

Slot	Value
Generic use case	Legal situations to be checked for (non)conformity to existing norms that establish limits.
Local use cases	Legally-defined roles, functions, and parametric values exist to control the social life of a country. When talking about agents and social action, there is a network of senses implying a dependence on roles, functions (or tasks), and parametrized value ranges within a normative description. Intended meanings include the possible roles played by certain objects and agents, the actual actions occurring during social life, as well as parametric limits over value types, such as age limits, speed limits, etc. Therefore, both class- and instance-variables are present in the maximal relation for this pattern.
Logic addressed	OWL, DL species
Reference ontologies	DOLCE-Lite-Plus, Core Legal Ontology
Specialized CODeP	$Norm \leftrightarrow Case$
Composed CODePs	Time-Indexed-Participation, Concept↔Description, Description↔Situation
Formal relation	CaseConformsToLimitViolation(φ,ψ,χ,x,y,z,t,c_1,c_2,c_3,d,s), where $\phi(x)$ is an agent class, $\psi(y)$ is a process class, $\chi(z)$ is a class of values within a value range, t is a time interval, c_1, c_2 and c_3 are three reified intensional concepts, d is a reified intensional relation, and s is a reified extensional relation.
Sensitive axioms	CaseConformsToLimitViolation(s,d) $=_{df}$ $\forall x, y, z(\varphi(x) \wedge \psi(y) \wedge \chi(z) \wedge$ participantIn(x,y,t) $\wedge$ locationOf(z,y,t) $\wedge$ (Object(x) $\vee$ Agent(x)) $\wedge$ Action(y) $\wedge$ Speed(z) $\wedge$ TimeInterval(t)) $\leftrightarrow \exists c_1,c_2,c_3$(CF($x$,$c_1$,$t$) $\wedge$ MT(c_1,c_2) $\wedge$ CF(y,c_2,t) $\wedge$ $\neg$CF(z,c_3,t) $\wedge$ REQ(c_3,c_2) $\wedge$ (DF(d,c_1) $\wedge$ DF(d,c_2) $\wedge$ DF(d,c_3) $\wedge$ $\forall s$(SAT(s,d) $\leftrightarrow$ (SETF(s,x)$\wedge$ SETF(s,y) $\wedge$ SETF(s,t)))
Explanation	Since OWL(DL) does not support relations with >2 arity, reification is required. The Description↔Situation pattern provides typing for such reification. Since OWL(DL) does not support classes in variable position, we need reification for class-variables. The Concept Description pattern provides typing for such reification. Similarly, since participation is time-indexed, we need the time-indexed-participation pattern, which is here merged with the previous two patterns (time indexing appears in the setting of the general normative situation).
OWL(DL) encoding (abstract syntax)	Class(CaseConformsToLimitViolation complete Description restriction(defines someValuesFrom(Object)) restriction(defines someValuesFrom(Action)) restriction(defines someValuesFrom(TimeInterval))) Class(LegalAgent complete Role restriction(used-by someValuesFrom(CaseConformsToLimitViolation)) restriction(classifies allValuesFrom(Object)) restriction(attitude-towards someValuesFrom(LegalTask))) Class(LegalTask complete Task restriction(used-by someValuesFrom(CaseConformsToLimitViolation)) restriction(classifies allValuesFrom(Action)) restriction(attitude-target-of someValuesFrom(LegalAgent))) Class(LimitViolation complete Parameter restriction(used-by someValuesFrom(CaseConformsToLimitViolation)) restriction(classifies allValuesFrom(complementOf LimitValueRegion)) restriction(requisite-for someValuesFrom(unionOf(LegalTask LegalAgent))) Class(ViolationCase complete Situation restriction(satisfies someValuesFrom(CaseConformsToLimitViolation)) restriction(setting-for someValuesFrom(Object)) restriction(setting-for someValuesFrom(Action)) restriction(setting-for someValuesFrom(Limit)) restriction(setting-for someValuesFrom(Time-Interval)))
Class diagram	

5. Conclusions

An overview of the relevance of ontology patterns for legal knowledge engineering (LKE) has been presented.

For some generic LKE use cases (conflict checking, information extraction, etc.), some solutions and issues, both on the reasoning and (content) modelling sides, have been mentioned. While the reasoning side of LKE is a fast-moving target, with interesting solutions coming from e.g. hybridating different inference engines and classifiers, the modelling side is far less developed, despite the huge literature that focuses on legal and jurisprudential content, let alone the work in formal ontology and beyond. This is not surprising. Currently, very few ontologies are actually *reused*, against the great expectations that have been grown in the field of reusability of semantic components.

In the past, the need for structures that grant systematicity to content modelling have mainly focused on so-called *top-level* ontologies, but the power of a small set of categories is not enough for realistic ontology projects like those presented to LKE. A top-level can even be a problem when its categories are brittle with respect to the domain task.

A more sophisticated approach, which ensures a much higher level of cognitive interoperability, is constituted by *foundational* ontologies like DOLCE. Nonetheless, although their rich axiomatization makes foundational ontologies ideally suited for building a partial-order of reference ontologies, they require a substantial cognitive load to be accessed and then successfully reused.

A new dimension of reusability has been introduced in ontology engineering (and here extended to LKE) which revisits some good practices from AI (*knowledge patterns*) and databases (*data model patterns*), by providing a meta-model, some operations, and a generalized characterization to *conceptual ontology design patterns* (CODePs). A foundation to CODePs is supposed to enhance the construction of tools for ontology design, as e.g. envisaged in the NeOn project (NeOn,), and to facilitate the collaborative and distributed negotiation of meaning across the members of a same or of sufficiently close communities.

For LKE, CODePs appear slightly more complex than in other domains, mostly because the use cases in LKE involve a layering of meaning (from the physical to the social, cognitive, and finally legal realms), which also require an extended reification of entities such as norms, contracts, cases, legal text corpora, etc. A complex network of dependencies between roles and tasks, agents, *normative positions*, validity parameters, assumed goals, cognitive attitudes, etc. makes the modelling task in LKE harder than elsewhere. A few steps toward the creation of a repository of legal CODePs have been sketched, together with some possible directions for further development.

References

Gangemi, A., Sagri M.T., Tiscornia D.: A Constructive Framework for Legal Ontologies. In R. Benjamins, P. Casanovas, J. Breuker, A. Gangemi (eds.): Law and the Semantic Web, Springer, Berlin (2004).

Gangemi, A., Borgo, S. (eds.): Proceedings of the EKAW*04 Workshop on Core Ontologies in Ontology Engineering. `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-118/` ((2004)).

Gangemi, A., F. Fisseha, J. Keizer, J. Lehmann, A. Liang, I. Pettman, M. Sini, M. Taconet: A Core Ontology of Fishery and its Use in the FOS Project, in (Gangemi and Borgo, 2004) (2004).

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

Sagri M.T., Tiscornia D., Gangemi A., An Ontology-based Approach for Representing "Bundle-of-rights", M. Jarrar & A. Gangemi (eds.), 2nd International Workshop on Regulatory Ontologies, in OTM Workshops, Springer, 2004.

Sagri M.T.: Progetto per lo sviluppo di una rete lessicale giuridica on line attraverso la specializzazione di ItalWordnet, in Informatica e Diritto, ESI, Napoli, 2003.

Motta, E., Domingue, J., Cabral, L., Gaspari, M. (2003) IRS-II: A Framework and Infrastructure for Semantic Web Services. 2nd International Semantic Web Conference (ISWC2003) 20-23 October 2003, Florida, USA (2003).

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).

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

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

Galton, A.: Reified Temporal Theories and How To Unreify Them. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1991.

Clark, P., Thompson, J., Porter, B.: Knowledge Patterns. Proceedings of KR00 (2000)

Gangemi, A., Ontology Design Patterns for Semantic Web Content. Y. Gil, E. Motta, R. Banjamins, M. Musen, Proceedings of 4^{th} International Semantic Web Conference, ISWC05 (2005).

McGuinness D.L. and van Harmelen F. (editors), OWL Web Ontology Language Overview, W3C Recommendation, 10 February 2004, `http://www.w3.org/TR/2004/REC-owl-features-20040210/` (2004).

Moore M.S. Legal Reality: A Naturalist Approach to Legal Ontology. *Law and Philosophy* 21: 619–705, 2002.

Forbus, K., Mostek. T., Ferguson, R., An analogy ontology for integrating analogical processing and first principles reasoning, in Proceedings of AAAI02, 2002.

DOLCE + library, available in OWL from: `http://dolce.semanticweb.org` Directly loadable from: `http://www.loa-cnr.it/ontologies/DLP.owl`.

Core Legal Ontology, loadable from: `http://www.loa-cnr.it/ontologies/CLO/CoreLegal.owl`

Gangemi, A., Pisanelli, D.M., Steve G. A Formal Ontology Framework to represent Norm Dynamics. Proc. Of Second International Workshop on Legal Ontologies, Amsterdam, 2001.

Gil, R., Garcia, R., Delgado, J. An interoperable framework for IPR using web ontologies, J. Lehmann, E. Biasiotti, E. Francesconi, M.T. Sagri (eds.), Proceedings of the First LOAIT Workshop, Bologna, Italy (2005).

Evren Sirin and Bijan Parsia and Bernardo Cuenca Grau and Aditya Kalyanpur and Yarden Katz Pellet: a practical owl-dl reasoner, Available at http://www.mindswap.org/papers/PelletJWS.pdf

Gilardoni, L., Biasiuzzi C., Ferraro, M, Fonti, R., Slavazza, P. LKMS – A Legal Knowledge Management System Exploiting Semantic Web Technologies. Y. Gil, E. Motta, R. Banjamins, M. Musen (eds.), Proceedings of 4^{th} International Semantic Web Conference, ISWC05 (2005).

Prud'hommeaux E., Seaborne A. SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/

Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean SWRL: A semantic web rule language combining owl and ruleml, W3C Member Submission, 21 May 2004. Available at http://www.w3.org/Submission/SWRL/.

Basili, R., Pennacchiotti, M., Zanzotto, F.M. Language Learning and Ontology Engineering: an integrated model for the Semantic Web. B. Magnini (ed.), Proceedings of Meaning Workshop (2005).

Horrocks, I. Fact++ web site. `http://owl.man.ac.uk/factplusplus/`

Hustadt, U., Motik, B., Sattler, U. Reducing SHIQ Description Logic to Disjunctive Datalog Programs. Proc. of the 9th International Conference on Knowledge Representation and Reasoning (KR2004), June 2004, Whistler, Canada, pp. 152-162.

Domingos, P., Richardson, M. Markov Logic: A Unifying Framework for Statistical Relational Learning. Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (pp. 49-54), 2004. Banff, Canada: IMLS.

Peters W., Sagri M.T., Tiscornia D. and Castagnoli S. The LOIS Project. LREC 2006, Genova, 2006.

Hay, D.C. Data Model Patterns. Conventions of Thought. Dorset House, NY (1996).

CODeP Repository, available from `http://www.loa-cnr.it/codeps/index.html`

Gangemi A., Mika P. Understanding the Semantic Web through Descriptions and Situation, Meersman R, et al. (eds.), Proceedings of ODBASE03, Springer, Berlin, 2003.

NeOn Project EU-IST 27595. `http://www.neon-project.org`.

A. Gangemi, V. Presutti, C. Catenacci, J. Lehmann, M. Nissim, C-ODO: an OWL meta-model for collaborative ontology design. Noy et al. (eds.): Proceeding of First CKC workshop, WWW2007 Conference, Banff, 2007.

Appendix: some excerpts from the Core Legal Ontology

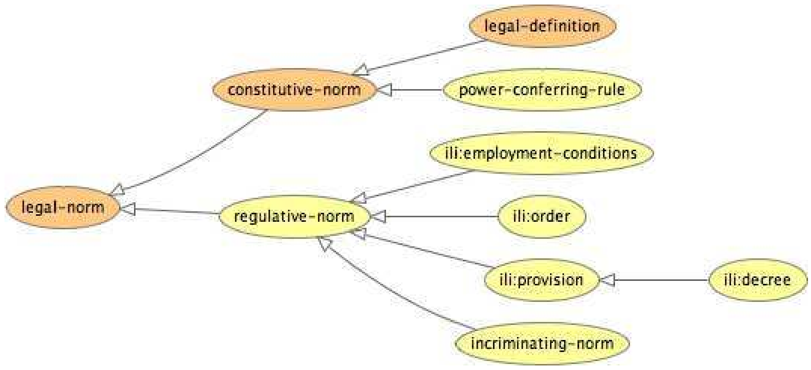


Figure 5. An excerpt of CLO taxonomy: legal norm types. "ili" classes are from the JurWordNet ontology (Sagri, 2003)(Peters et al, 2006). The $Norm \leftrightarrow Case$ CODeP is the generically related CODeP for legal descriptions

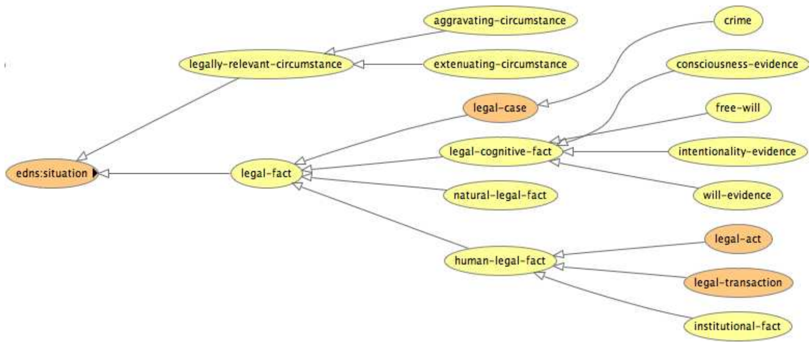


Figure 6. An excerpt of CLO taxonomy: legal fact and circumstance types as situations. The $Norm \leftrightarrow Case$ CODeP is the generically related CODeP for legal facts.

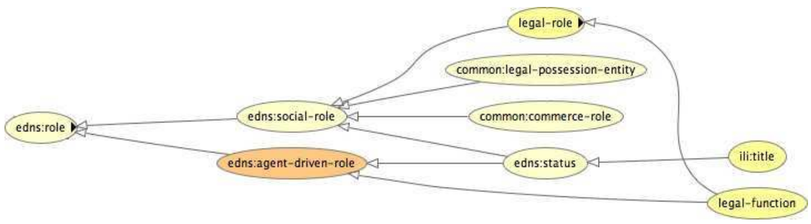


Figure 7. An excerpt of CLO taxonomy: legal roles. The $Norm \leftrightarrow Case$ CODeP is the generically related CODeP for legal roles.

Figure 8. An excerpt of CLO taxonomy: legal subjects (or persons). In Fig. 9 a related CODeP for socially-constructed persons is provided.

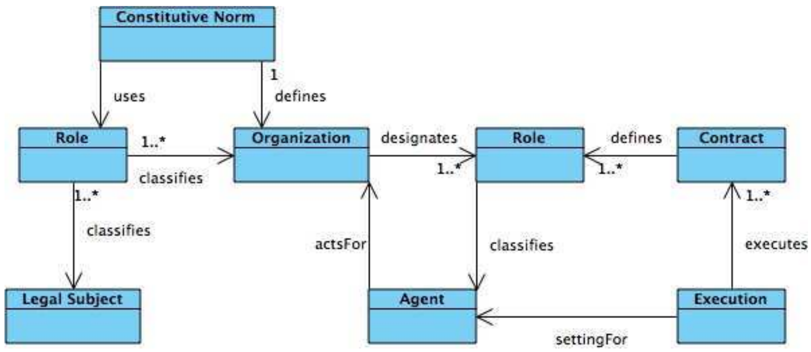


Figure 9. The *SociallyConstructedPerson* CODeP. Persons can be legal subjects (either natural persons or not), including organizations, or legal entities deputed to law enforcement. For example, an organization is defined by means of a constitutive norm that also uses concepts that either classify the organization or other legal subjects. An organization designates at least one role that can classify agents. For classified agents, we can say that they *actFor* the organization when they are in the *setting* of a contract execution that defines the organization's role.

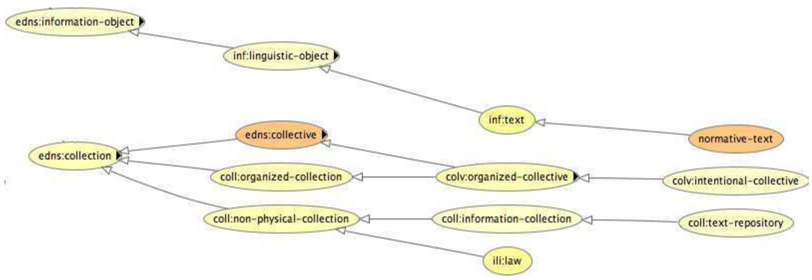


Figure 10. An excerpt of CLO taxonomy: legal informations and collections. In Fig. 11 a related CODeP is presented for information realizations and collections.

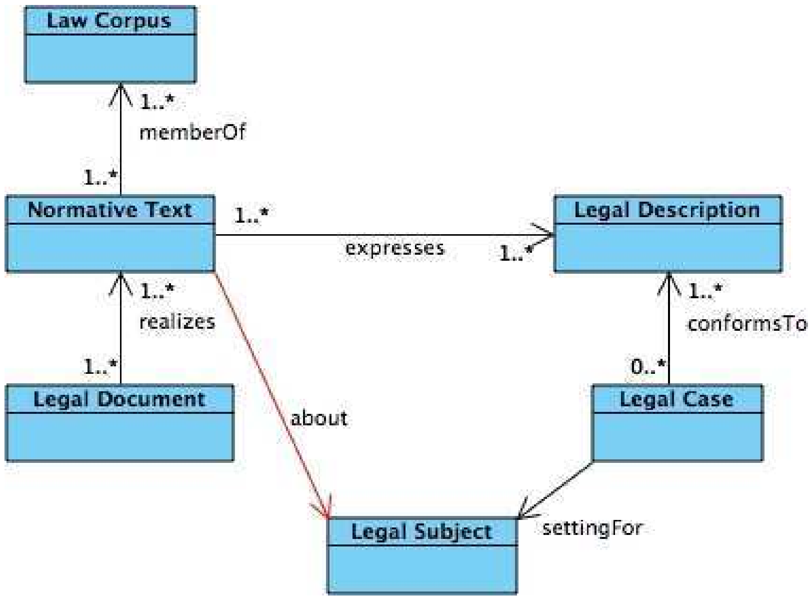


Figure 11. The *LegalInformationRealization&Collection* CODeP. A normative text is an information object, *member* of a corpus of laws, and *realized* by at least one legal document (its support). A text *expresses* a legal description (the public meaning of a law). For any legal case that *conforms* to the description, the text is *about* the legal subjects in the *setting* of the legal case.