IST-2004-027655

ESTRELLA

European project for Standardised Transparent Representations in order to Extend Legal Accessibility

Specific Targeted Research Project
Information Society Technologies

Deliverable 1.4

OWL Ontology of Basic Legal Concepts (LKIF-Core)

Due date of deliverable: 31 December 2006
Actual submission date: 22 January 2007

Start date of project: 1 January 2006 Duration: 30 months

University of Amsterdam

FINAL

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Executive Summary

This document is part of Deliverable D1.4 of the Estrella project. It reports the work performed under Task 1.5 of the Estrella Workplan: "*Defining an ontology of basic legal concepts, using the Ontology Web Language (OWL).*" The ontology itself is the other part of this Deliverable.

In this document we describe the methodology used to develop the ontology, which we have named *LKIF-Core ontology*. The workplan suggests that by starting from an already existing core-ontology – LRI-Core [Breuker et al., 2004b]— this development process could be a rather simple top-down approach.[1] This was not the case, as the number of legal concepts in LRI-Core was rather small; it was rather a top ontology covering abstract concepts of common-sense rather than the field of law. Moreover, it appeared that in law, and in particular in legal reasoning, complex patterns of concepts are used, which do not easily fit into an ontology. We adopted a rather restrictive view on what should be in an ontology and what kinds of knowledge are beyond the scope of an ontology proper, and should rather be separated from the ontology. This kind of knowledge structures are called: *frameworks*. This distinction was not so much inspired by conceptual puritanism, based upon experiences with earlier kinds of legal core ontologies [Breuker and Hoekstra, 2004a], but particularly by the fact that the expression of such structures in OWL-DL is often not, or only partially possible. This is well explained in Deliverable D1.1, where LKIF itself is discussed: for more complex or other types of knowledge than terminological knowledge we also need rule formalisms. Frameworks and rules are beyond the scope of D1.4, but they are described, because we foresee that they will play a role in the next phase of Estrella. For instance, frameworks that describe dependencies in reasoning with norms or that structure legal problem solving methods or argumentation are powerful components in artificial legal reasoning. This is presented in Chapter 2.

The development methodology is specified in Chapter 3. Requirements on 1) what legal knowledge has to be captured and 2) for what functions the ontology will be used play an important role in any methodology of ontological engineering [Staab and Studer, 2004, Part-II].

Users of legal knowledge We have identified three 'user' populations of legal knowledge and reasoning:

Citizens who have to understand legal sources and legal procedures to plan their activities.

[1] "Input for this task will be the prior knowledge of UvA on LRI-Core, a legal core ontology. "

Legal professionals who have roles in the legal system and whose role is in the first place to assess whether there is compliance and violation of the law in society, and what the legal consequences are.

Legal scholars who observe and reflect upon the work of legal professionals, especially on what happens in courts (precedent law) and what happens in politics (legislation).

Although these populations have different uses and views on legal concepts, we assumed that the different perspectives involved are not conflicting: the legal system – the 'professionals' – and society – the citizens are in permanent interaction. The results of this interaction are not limited to the participants in courtrooms but are a major subject in news reporting. The views of legal scholars are very important because they develop abstract terms that may articulate a legal core ontology.

We have collected 'basic legal' terms from these three sources. The collection (list) was about 250 terms long. These terms were scored by partners in the Estrella project on their relevance to law, their level of abstraction and their legal specificity, so that we could start with a basic set of about 50 terms.

Use of legal ontologies The perspectives on the sources of legal knowledge in terms of users of legal knowledge provide a specification of which terms with what definitions are used. This is also the context in which the *LKIF-Core* is to be used – in fact a rather broad legal context. However when it comes to actually implementing the ontology, the demands on its prospective use bring a number of technical issues. Lightweight ontologies, i.e. ontologies which have only a few properties to define their meaning, may very well play a crucial role in information management of documents. However, if they are also use as resources for inference and reasoning more demands are put on the 'thickness' of their semantic descriptions, The latter is the case in Estrella. In Section 3.2.2 we discuss five typical roles of ontologies.

Methodology Using the methodology described in Section 3.3, a number of central and relatively abstract clusters of concepts are identified and modelled more or less independently as 'modules' of an ontology. A cluster is a set of terms which have a high degree of (mutual) relations and can be conceptualised in relative isolation. Examples are notions about 'norm' and 'action'. We have identified ten basic clusters, which cover more or less the core list of legal terms. These clusters are described in Section 4.1.3. Also a formal definition of the terms in these clusters was produced to provide a firm foundation for further modelling. The OWL modules that represent these clusters were subsumed under major categories of the LRI-Core ontology as to provide a coherent structure for *LKIF-Core*. This operation involved also some minor modifications and adaptations of the original modules. This structure was the starting point for further refinements so that the current version of the *LKIF-Core* ontology of basic legal terms contains now about 200 classes (concepts). [2] The descriptions and definitions of these terms are reported in this document, the full *LKIF-Core* is

[2] It was foreseen on the basis of existing core ontologies for other areas that this number would be between 100 and 200.

available at `http://www.estrellaproject.org/lkif-core` [3], its documentation is available online at `http://www.estrellaproject.org/lkif-core/doc`.

Conclusions and future work The development of *LKIF-Core* has extensively relied on earlier work by partners both inside and outside the project (research literature and legal ontologies), so that we have some ground for the claim that *LKIF-Core* is currently the most comprehensive and well structured (core) ontology for law. As we have checked its formal consistency, its validity will be assessed in its use in the next phase of Estrella. Although in the planning of this phase the maintenance and further development of *LKIF-Core* is not foreseen as a separate task, it is tempting to further develop the insights we have acquired. These ambitions involve the following aspects:

- Construction of two or more frameworks:
  - A framework for normative reasoning that can handle norms and their application to legal cases.
  - A (set of) framework(s) that represent structures of argumentation. A proposal for co-operation with the AIF-Project (Argument Interchange Format) is included in this Section 5.3.
  - Frameworks that represent methods for solving legal problems, in particular for legal assessment (which uses the framework for normative reasoning). This work can be seen as a further development of earlier work on problem solving methods in CommonKADS [Breuker and Van De Velde, 1994, Valente et al., 1999b].
- Construction of a software reasoning architecture that allows handling the reifications that are implied by the representation of norms.

[3] This Deliverable is publicly available, and resides under the LGPL licence, `http://www.gnu.org/copyleft/lesser.html`

OWL Ontology of Basic Legal Concepts (LKIF-Core)

Deliverable 1.4

Authors:

Joost Breuker, Rinke Hoekstra, Alexander Boer (Eds) and Kasper van den Berg
University of Amsterdam

Giovanni Sartor and Rossella Rubino
University of Bologna

Adam Wyner and Trevor Bench-Capon
University of Liverpool

Additional contributions:

Monica Palmirani
University of Bologna

In this document, part of deliverable D1.4 of the Estrella project, we describe the requirements, design principles, applied methodology and content of an ontology of basic legal terms, called *LKIF-Core*. The work reported is performed under Task 1.5. The ontology itself can be accessed through `http://www.estrellaproject.org/lkif-core`, its documentation is available online at `http://www.estrellaproject.org/lkif-core/doc`.

Contents

1 Introduction **1**

1.1 Overview 2

2 Design Principles **7**

2.1 Ontologies and Frameworks 7

2.1.1 Situational frameworks 8

2.1.2 Mereological frameworks 10

2.1.3 Epistemological frameworks 10

3 Development Methodology **12**

3.1 Context of Use 12

3.2 Perspectives and Requirements 14

3.2.1 Roles and tasks of legal agents 14

3.2.2 Roles of *LKIF-Core* 18

3.3 Middle-Out Ontology Development 21

4 An outline of *LKIF-Core* – The Methodology Applied **24**

4.1 Building and Clustering 24

4.1.1 Collecting and scoring legal terms 24

4.1.2 Conventions 25

4.1.3 Clustering 26

4.2 Integrating other upper/core ontologies 38

4.2.1 Principles and main categories inherited from *LRI-Core* . . . 40

4.3 Evaluation 44

5 Modules **45**

5.1 Modules and Dependencies 45

5.2 Glossary of Classes and Properties 47

5.2.1 Action Module 47

5.2.2 Expression Module 50

5.2.3 Legal Action Module 54

5.2.4 Legal Role Module 56

5.2.5 Mereology Module 57

5.2.6 Modification Module 58

5.2.7 Norm Module 61

5.2.8 Place Module 67

5.2.9 Process Module 69

5.2.10 Role Module 70
5.2.11 Rules & Argumentation Module 71
5.2.12 Space-Time Module 72
5.2.13 Time Module 72
5.2.14 Top Module 73
5.3 Towards an argumentation framework: the AIF-project 75
5.3.1 Introduction 75
5.3.2 Outline of AIF Project 75
5.3.3 Key Elements of AIF and Extensions 75
5.3.4 Relevance to the Estrella Project 77
5.3.5 Recommendation to the Estrella Project 77

A Appendix 79
A.1 Scores obtained in assessing legal concepts 79
A.2 Initial Definitions 83
A.2.1 Place 83
A.2.2 Time 84
A.2.3 Process 85
A.2.4 Action 85
A.2.5 Role 86
A.2.6 Expression 87
A.2.7 Norm 89
A.3 Final Definitions 90
A.4 On reasoning about norms 120

ESTRELLA
p/a Faculty of Law
University of Amsterdam
PO Box 1030
1000BA Amsterdam
The Netherlands

tel: +31 20 525 3485/3490
fax: +31 20 525 3495
`http://www.estrellaproject.org`

Corresponding author:
Joost Breuker
tel: +31(20)525 3494; + 33 4 66774227
`breuker@science.uva.nl`

Chapter 1

Introduction

In this document we report the work performed under T1.5 which is described in the Estrella Technical Annex as:

> ”Defining an ontology of basic legal concepts, using the Ontology Web Language (OWL). Input for this task will be the prior knowledge of UvA on LRI-Core, a legal core ontology.”

OWL is part of the LKIF knowledge representation formalisms for building and exchanging legal knowledge bases (see [Boer et al., 2007]). Since it has become the recommended formalism for providing ‘semantics’ to the Semantic Web (February, 2004), it has become the *de facto* standard for expressing ontologies. Although ontologies have been inspired by the branch of metaphysics called ‘ontology’, the notion of ‘ontologies’ in information technology has a slightly different orientation. Where the basic philosophical question in ontology is ‘what exists’, the perspective in ontological engineering is rather to capture the basic ingredients of our knowledge: concepts. For instance, ontologies may refer to virtual worlds. An ontology consists of “definitions of the terms used to describe and represent an area of knowledge” [1]. One may find more liberal descriptions of what an ontology is. For instance, the much quoted definition of an ontology by [Gruber, 1994] states that an ontology is “a formal representation of a conceptualisation”. This would mean that any formal model would be an ontology, where in the definition by Heflin an ontology is limited to the definition of terms, rather than a full conceptualisation. In this document we will reserve the term ‘ontology’ in the restricted sense, i.e. as consisting of definitions of terms.

Terms stand for concepts and concepts are represented in OWL as classes. To be precise: in OWL a class is (theoretically) viewed as a set of individuals, and is as such stands for extensional definition of a term. However, in practice this set may not be enumerable at all, so classes in OWL (and for that matter in all description-logic based knowledge representation formalisms) are rather defined in terms of sufficient and/or necessary properties, which may relate classes to other classes.

[1] Jeff Heflin, OWL-Use cases, http://www.w3.org/TR/2004/REC-webont-req-20040210/

1.1 Overview

Design Principles As will be explained in Section 2 the granularity of what is a term (concept) may vary. For instance, the concept of car as a motorised vehicle assumes that a motor is an essential part (property) of a car. However, enumerating all the parts of a car, and including their topological connectivity as part of its definition would no longer be viewed as "defining" a car, The parts and its connections are its structural description. We will see this rather as a *framework* than as a simple class definition. Therefore, besides an ontology of basic legal concepts – to be referred to as *LKIF-Core* – we also extract frameworks that represent composite views on concepts. As it is dependent on the granularity of modelling chosen, the distinction between framework and ontology may not always be very clear cut. More about this in Section 2. It should be noted that the work reported in this Deliverable covers in the first place the *LKIF-Core* ontology. Frameworks are introduced here for the following reasons:

- Frameworks and descriptions of terms cannot always be sharply distinguished. Many problems in the development of an ontology are due to the fact that it is often difficult to distinguish between what one knows about a term and what meaning a term carries in a particular context. This is particularly relevant because the relation in a framework may be between a concept and a reified structure of concepts. For instance, a norm is a qualification over some statement (expression). OWL-DL does not accept such structures, and the modelling problem can be resolved by a reasoning architecture, rather than by knowledge modelling, as is explained in Section 2.1.1.

- In discussing the relationship between Task 1.2 and Task 1.5 it appeared that the use-context of formalisms respectively ontology needed a view on how the inference rules, respectively concepts were to be used in legal reasoning. Epistemological frameworks that describe problem solving and argumentation were needed as background knowledge. However, in drafting the Estrella workplan it was not foreseen that these frameworks could be an additional resource for reuse in constructing legal knowledge systems. This is not the case for legal argumentation, that takes a rather prominent role in D1.1 and is also described in this Deliverable (see Section 5.3).

As the representation of knowledge structures more complex than proper ontologies may require formalisms (rules) that go beyond the expressiveness of OWL, we have added a section on the relationship between descriptive and rule formalisms.

Methodology The methodology we have followed is described in Section 3. Ontologies contain definitions of terms, but these definitions are not independent of the context of use. For instance, a car can be defined as a motorised vehicle (traffic (code)), or as a commodity (trade (law)), or as a device (maintenance (contract)). One may argue that ontologies capture specific *senses* of terms rather than *meaning.*[2]

[2] In linguistics the semantic interpretation of a sentence refers to a particular sense of a word. This sense is a selection or derivation of what we *know* about a concept, i.e. its full meaning.

There are two kinds of perspectives of use. The first perspective is how Law and legal concepts are used in practice by humans (Section 3.2.1). The second perspective is about the prospective use of the LKIF-(Core)-ontology itself (Section 3.2.2).

Whose legal knowledge? There are three major views and interests: those of the citizens to whom the majority of law is directed. The second view is by the actors in the legal system itself (judges, lawyers, etc.). The tasks they perform, i.e. the kinds of problems they solve are different. Finally, legal theoreticians may have developed or refined concepts of law.

Citizens' use of law Aside from the (e-Government) issue of informing citizens about law, law – legislation and the working of the legal system – has to be understood in the first place. There is a common-sense view and awareness about law that is that least sufficiently accurate as to make conflicts with law and in particular the legal system rare. Citizens comply with the law, or at least they plan their activities in such a way as to avoid being sanctioned by the legal system. [3] Actions are planned by citizens under legal constraints, i.e. their understanding of law limits or guides their options in acting. For the citizen, law is part of their daily (social) life, and their knowledge and understanding of law has a strong common-sense flavour.

The lawyers view In this context it is professional legal knowledge that is in focus. For legal professionals law is what happens in court and to them law has the character of a permanent debate, even when they recognise that the debate should be ultimately grounded in legal sources. However, also these legal sources change. [4] Law is made by argumentation in dialogues, starting with political debate if it concerns legislation.

Legal scholars and jurisprudence This view of law as continuously evolving under changes in society and the legal debates, is also the point of departure for most legal theoreticians. Moreover, they reflect on these debates, adding their own debates to it, but also a new vocabulary and new, clarifying concepts.

The law is also the object of study, at the level of legal sociology, legal doctrine and jurisprudence. Legal doctrine in particular, aims at explaining and clarifying the law and to suggest what interpretations of legal texts and more generally what approaches to legal issues are the most appropriate ones, or the ones which are more likely to be successful. Jurisprudence works at a further level of abstraction, analysing the law as a whole, critically evaluating doctrinal debates, considering the links between the law and other disciplines.

The difference between the views of citizens and those of legal practitioners is well illustrated by the concept of 'liability'. In legal common-sense, it is reasonable that one is held responsible for one's own actions, but it may come as a surprise that a

[3] A noteworthy exception are those citizens who seek profit rather than compensation by summoning other parties to court.

[4] Also in this respect law is different for the citizen who finds that the law often changes too slowly and is too conservative as it reflects often a consensus that was already socially well established.

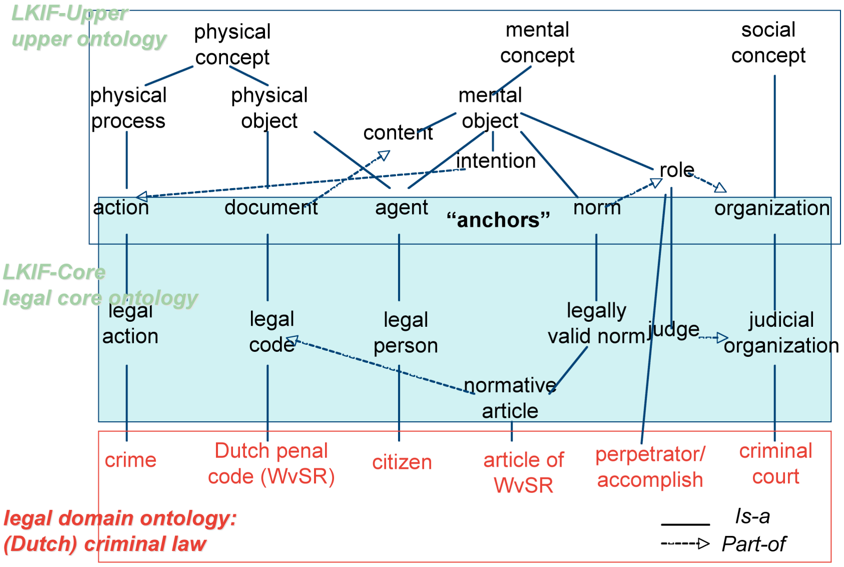

Figure 1.1: A legal upper/ core ontology to structure a domain ontology.

parent is called into court for trespasses of minors that are under his/her custody. Liability is not the same as responsibility. What liability exactly means or should mean is a long debate between legal theoreticians.

What will *LKIF-Core* be good for? Besides this perspective on human use of concepts in practical contexts, the second perspective is how the LKIF-ontology *itself* is to be used. This is described in Section 3.2.2. Upper- or core ontologies are in the first place intended to support top-down knowledge acquisition (modelling) of domain ontologies, see Figure 1.1. An upper/core ontology provides the initial structure (super-classes) for a domain ontology. Besides this pre-structuring, the properties defined for the super-classes are inherited and reused. As it is likely that the concepts defined in an upper/core ontology are highly abstract and relatively legal domain independent, one can expect that *LKIF-Core* can be used to assess similarities and differences between concepts in equivalent legal domains of different jurisdictions. For instance, it is no surprise that the various traffic codes of different (European) countries refer to the same objects and actions. Indeed experience shows that the traffic codes of the Netherlands (RVV-90) and of Minnesota share largely the same ontology, while the codes themselves are cast in totally different formats. [5] For traffic codes this may not be very surprising. However, also for criminal law, which is notoriously divergent in conceptions between national (European) jurisdictions of what constitutes a crime, we have found that a legal core ontology is very useful in tracing similarities and divergence. In other words: a legal core ontology also supports translation, alignment and even harmonisation between national ju-

[5] The codes are used as exercises for students following a Master course in knowledge representation and ontology at the University of Amsterdam.

risdictions. [6] Another role for an LKIF ontology – usually a domain-ontology – is to allow translation of terms used in knowledge bases of vendors, which helps to obtain meaning preserving translations.

Ontologies, in particular upper ontologies, provide a basis for special inference. Formally, ontologies are a special kind of theories, whose axioms can be seen as meaning or as terminological postulates. These postulates constrain and force a special class of inferences. An ontology that relates terms for describing time is for instance the basis for a calculus on time [Allen, 1984]. This calculus can be seen as inference constrained by the semantics of the terms included by the inference engine of the knowledge representation formalism (for OWL: a classifier). We may expect that for instance *LKIF-Core* definitions of terms of deontic qualifications allow us to establish all possible paraphrases of a norm: i.e. it may be useful to develop a deontic calculus.

Particularly in the context of the Semantic Web, ontologies are primarily used to enable a richer and more conceptual way of information retrieval and information management. Large collections of documents can be 'semantically' accessed by annotations and by search based on an ontology. The legal world is *par excellence* a world rooted in documentation. Information retrieval becomes information serving if the user is not only presented with the relevant (parts of) documents, but in particular when the information can be interpreted automatically as to provide answers to legal questions. Ontology based question answering techniques have been explored in previous European projects (e.g. CLIME [7]).

Basic terms of law Not only how we model, but also what we model is of importance. *LKIF-Core* should cover "basic concepts of law". This means that the terms selected should be both highly abstract as to cover all domains of law, and be relevant for law. The latter is a problem because many terms relevant for law are not exclusive for law. For instance, the terms used in French legislation ('codes') are indistinguishable from those used in newspaper text, which is different from other specialised fields of practice (e,g, medicine or engineering) [Lame, 2006]. This means two things. First that newspapers may contain a lot of information about legal issues, which is true when it concerns politics of legislation or reporting legal decisions (in particular from criminal courts). A second interpretation is that indeed law is deeply rooted in common-sense as it is intended to be understood and complied with by citizens. In Section 4.1.1 we describe how we collected and assessed this basic set of concepts of law.

Methodological approaches In constructing LKIF-ontology we followed two approaches, which are described in Section 3.3. The first one is a classical *top-down* approach, in which an upper-ontology provides the initial structure. As specified in the Technical Annex (TA), the upper ontology that is a first candidate is LRI-Core [8],

[6] These experiences and insights have been obtained in the 5th framework project "e-Court" (IST-2000-28199). See for details [Breuker et al., 2004a].

[7] CLIME was an European project (IST 25414, 1998-2001): see http://www.bmtech.co.uk-/clime/index.html

[8] http://wiki.leibnizcenter.org/index.php/LRI_Core

but we have also investigated what kind of modelling support could be provided by other kinds of upper ontologies, which is described in Section 4.2. A second approach can be characterised as a *'middle-out"* one. It is inspired by the work of [Hayes, 1985] where clusters of interdependent concepts are identified. For instance, understanding the notion of 'action' implies 'intention', which relates to 'expected-effect', which requires the notion of 'expectation'. Some clusters may be simply imported as 'modules'. For instance, one may find readily available ontologies for 'time', 'space', mereology, etc.

Chapter 2

Design Principles

As is also stated in the Technical Annex (TA) of the Estrella project, OWL is the obvious choice for representing ontologies. As the LKIF-ontology has to serve many purposes that include reasoning we need to guarantee that the inferences remain tractable, so we have to limit the expressiveness of the representation to OWL-DL. However, as we foresee that the support in constructing automated legal reasoning services cannot be strictly limited to a collection of general legal terms, but should also include 1) special legal inference (see Section 3.2.2) and 2) typical problem solving and argumentation methods. Although it has become common practice to describe these re-usable reasoning structures as 'ontologies', we will distinguish proper ontologies from these frameworks that describe dependency structures in reasoning rather than providing definitional descriptions of terms. The reason that ontologies and frameworks became mixed up is the fact that OWL allows one to express more than only terminological knowledge. The confusion may be due to the belief that because knowledge is cast in OWL it must therefore be an ontology. This is a misconception. In OWL one can express a large variety of knowledge, ranging from factual information ('individuals') to complex relational structures that stand for (stereo)typical *use* of knowledge.

Although OWL can be used for more than the representation of ontologies, its limited expressiveness is often a handicap. For instance, even with the added expressiveness of OWL-DL 1.1 it remains impossible to properly distinguish between individuals in class restrictions. A candidate for lifting the limitations of OWL is the Semantic Web Rule Language (SWRL)[1], which is intended to be closely compatible with OWL. In this Section we will outline how we foresee the extension of the LKIF-Ontology as to support also re-usable legal reasoning structures, and outline the requirements for the expressiveness of LKIF itself.

2.1 Ontologies and Frameworks

We adhere to a rather restrictive view on what an ontology should contain: terminological knowledge, i.e. intensional definitions of concepts, represented as classes with which we interpret (model) the world. In the actual practice of defining concepts, the class/subclass structure of ontologies is soon assumed background, and

[1] `http://www.w3.org/Submission/SWRL/`

"usual" or common relationships with other concepts become foreground. For instance, the definition of "eating-in-a-restaurant" may emphasise dependencies between actions of clients and service personnel as the major structure. However, 'eating-in-a-restaurant' is not some natural sub-class of 'eating'. It refers to some typical model of how eating is put in the context of a restaurant. We do not want to have a proliferation of all contexts of eating, such as eating-at-home, eating-with-familiy, eating-at-a-dinner-party, eating-outside, eating-in-the-kitchen, etc.(cf. [Bodenreider et al., 2004], [Breuker and Hoekstra, 2004a]). This perspective takes the context in which a term may occur as a some specialisation. Any appropriate context of occurrence would be a candidate for defining a sub-class, leading to an endless proliferation of sub-classes. We do not want to see that "eating-oysters-with-a-friend-at-the-Pic-Saint-Loup-on-a-sunny-day" is a concept. It is an event in a particular situation, which is modelled by knowing what eating, oysters, friends and sunny days are; you may have missed out that the Pic-Saint-Loup is not a restaurant but a remarkable hill near Montpellier.

However, if experiences re-occur and have a justifiable structure, it may pay to store these structures as generic descriptions, because they deliver a predictable course of events for free. Eating in a restaurant is a typical example and it served in the Seventies to illustrate the notion of knowledge represented by scripts [Schank and Abelson, 1977] or 'frames' [Minsky, 1975]. In this document we will use the term *framework* for representing concepts (classes) which refer to what they consist of, or are dependent of, rather than what they are.

This shift of attention from ontological (definitional) representation to relational, or dependency-oriented concept description (i.e. a framework) confuses a more fundamental discussion on the *meaning* of the terms that will be commonly used in LKIF statements. Furthermore, a strict ontological stance allows us to define a *LKIF-Core* ontology which is as independent as possible of particular legal theoretical views on e.g. normative statements. For instance, in [Boer, 2006b]'s view, norms can be seen as a preferential ordering of behaviour – a framework– , where in the LKIF-Ontology, a norm can be defined no more and no less as a generic description of behaviour that an agent or device should comply with.

To be sure, ontologies and frameworks cannot always be cleanly separated in practice: it is rather a methodological distinction than a dismissal of the representation of frameworks in OWL-DL. We distinguish between three kinds of frameworks: situational, mereological and epistemological frameworks. These frameworks are extensions of the *LKIF-Core* ontology.

2.1.1 Situational frameworks

Situational frameworks are stereotypical plans for achieving some goal in a recurrent context. Making coffee may be such a plan. However, the plans may involve transactions in which more than one actor participates. In the 'eating-in-a-restaurant' actions of clients (ordering, paying) are interwoven with corresponding actions of patrons (noting, serving). In the legal world, such situational frameworks may be pre-scribed in articles of procedural ('formal') law. These prescriptions may be explicit or implicit. For instance, Dutch administrative law (RAWB) prescribes explicitly the requirements of an appeal to a decision of a governmental agency. This prescription assumes a transaction by which the appellant prepares an appeal and sends this to the agency. The agency re-motivates or re-considers its decision and replies to the appellant. [2] Although *stereotypical* plans ('customs') and *prescribed* plans may differ in their justification – rationality vs. authority – their representation is largely similar.

Norms, which are a key concept in law, prescribe behaviour. The prescription of this behaviour consists of a generic situation description (conditions) and some specific state or action. This description is qualified by some deontic term. For instance, the norm that "vehicles should keep to the right of the road" states that the situation in which a vehicle keeps to the right is the obliged one. Although this conceptual view appears self-evident, it should be noted that the deontic qualification – an obligation– is 'meta' to the situational description. Although the situation description may be expressed without problems as a framework cast in OWL-DL, but OWL-DL does not allow this deontic qualification to be expressed as a meta-class, or rather that the deontic qualification allows reified statements. There are three solutions to this problem. The first one is to go for OWL-Full/RDF which allows meta-classes, but is intractable. The second one is to express norms as rules, which has an intuitive appeal, as in common sense terms legal norms are synonymous with legal rules: one has to obey the rules. However, this does not immediately solve the tractability problem either as description logics and rule formalisms are not necessarily compatible (see also [Boer et al., 2007]). The third solution is a architectural one, in which two knowledge bases are used. One for expressing the situation descriptions and one for the deontic terms associated with the descriptions. This will only work if reasoning at the object-level – matching an individual situation with the (generic) situation description – does not (heavily) interact with reasoning at the meta-level. This is the case if we see that associating a qualification to a situation does not change that situation, i.e. there is no non-monotonicity between the two levels of reasoning. [3]

Qualifying situations is not exclusive for law, or for that matter: there are many other terms for qualification than deontic ones, and they should encounter similar problems in representation as deontic qualifications. For instance, we have aesthetic judgements about events and states, i.e. an assessment whether something is beautiful or not. However, we may believe that the beauty of a flower is not a qualification, but a quality of a flower, i.e. an attribute of a flower rather than some assigned belief. One may even hold that the colour of a flower is as much in the eyes of the beholder as its beauty. Moreover the notion of (dis)belief itself is a qualification of some piece

[2] In fact the obligation for the agency that follows from the action of the appellant is not explicit at all, but it is stated that the appellant will be satisfied if no timely or motivated reply is produced. The legal constraints serve as a rationale for acting.

[3] To a particular situation more than one norm may be applicable (match). For instance one may be both speeding and not keeping sufficiently to the right side of the road. The set of applicable norms may contain 'exceptions', i.e. norms with intersecting situation descriptions but deontic qualifications that exclude one another. For instance, the speeding car may be overtaking another car, in which case it is no longer obligatory to keep fully to the right. For the moment we are not concerned with the resolution of such conflicts – or rather: how to assign priorities to these conflicting norms. For a full account, see [Boer, 2006b]

of information. It is here not the place to go deeper into the (e.g. auto-epistemic logics involved in) reasoning about qualification and qualities. The point is that ontological commitments should refer to what is 'inherent' to a concept, rather than what one may attribute to it. Attribution is an epistemic problem; defining properties requires an ontological stance. Therefore we prefer the 'architectural' solution to the representation problems of norms rather than the OWL-Full or rule based approaches which are not only intractable, but do not represent a proper distinction between ontological and epistemological views.

2.1.2 Mereological frameworks

Entities may be composite entities. Not only objects, but also processes may contain other objects or processes. It is tempting to include in the definition of a concepts a mereological (part-of) view. For instance, one may include in the definition of a car that it has at least three, and usually four wheels., and at least one motor. However, a full *structural* description of all its parts and connections goes beyond what a car "essentially" is. Structural descriptions, such as the topology of a circuitry board are the models which are the major knowledge base for model-based and qualitative reasoning systems (see e.g. [Davis, 1984], [Hamscher et al., 1992]). As already stated in Section 2.1, the distinction between a framework and a defining description of a term (ontology) may sometimes be very thin. For instance, if we want to describe a bicycle as distinct from a tricycle, it is necessary to use the cardinality of the wheels as defining properties. Mereological frameworks may go under a large diversity of names: structural models, configurations, designs, etc. It should be noted that frameworks are generic, i.e. they act as pre-specified patterns that get instantiated (parametrized) for particular situations.

2.1.3 Epistemological frameworks

Reasoning structures may also be represented by frameworks. Although the terms 'reasoning' and 'inference' often used as more or less synonymous, we want to reserve the term inference for making explicit what is implicit in a knowledge base, given some inference engine: most generally speaking: a 'theorem prover'. To make efficient use of inferences, control over the making of inferences is required. That is what problem solving is about. In problem solving, there is a specific goal: a problem to be solved. It is not only the goal oriented (teleological) perspective that is used to constrain the inference making, but also the fact that a problem may be decomposed in smaller sub-problems or sub-tasks. This insight has inspired Descartes to write his famous "Discours de la Méthode". Breaking up the problem or 'task' makes the total problem space smaller. [4] A problem solving method (PSM) is not only a break-down of a problem, but also provides control over the making of inferences by assessing success and failure in arriving at the (sub)goals. As problem solving methods (PSM) have a heuristic, rational flavour, but have not necessarily a logical grounding. Their justification is a gain of efficiency in solving certain types of

[4] Usually, making inferences is thought of in terms of search rather than deriving new information in the classical AI literature.

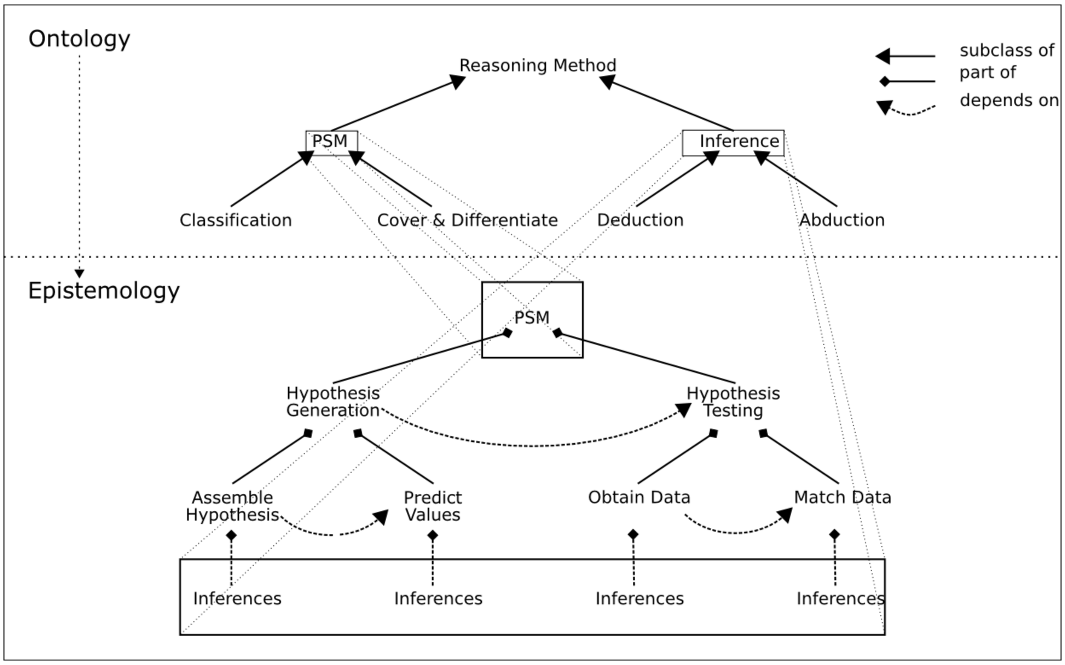

Figure 2.1: Ontology of reasoning terms vs framework for reasoning

problems (e.g. troubleshooting (diagnosis) of devices) given the particular structure of the domain knowledge and its representation. Problem solving methods have two major components: some method for selecting or generating potential solutions (hypothesis), and some methods for testing whether the solutions hold. Whether they hold may be due to the fact that they satisfy all the specified requirements (constraints) or whether they correspond with ('explain') empirical data. The former is particularly of importance in design and planning problems ('synthesis' problems); the latter is typical for analytic problems such as diagnosis or prediction. Particularly in the beginning of the nineties, a large variety of PSM have been constructed or identified in research on methodology and technology of knowledge systems (knowledge engineering). For instance, in the CommonKADS methodology libraries of PSM have been constructed to enable re-use and top-down knowledge acquisition [Breuker and Van de Velde, 1994], [Mottta, 1999]. [Schreiber et al., 2000]. [5] In Figure 2.1 we present a schematic explanation of the relationship between ontology and epistemological frameworks. The terms used in the framework – a PSM – find their definitions in an ontology in which the main structure is a class-subclass one, i.e. one that catches similarities and distinctions between terms.

[5] It was also in this community of researchers that work on ontologies was initiated. Ontologies were viewed as the complement of PSM, i.e. they were used to specify the required domain knowledge for the knowledge base of a knowledge system. As it appeared that ontologies could also be used to handle large quantities of documents – i.e. in knowledge or information management – their usefulness for a semantic based web technology became evident.

Chapter 3

Development Methodology

In this chapter we describe the methodological considerations underlying our approach to the development of a core ontology for the legal domain. The first section describes the context of use for the ontology, section 3.2 provides an overview of perspectives and requirements of *LKIF-Core*. Section Section 3.3 describes the principles underlying the middle-out approach in ontology development we have applied to *LKIF-Core*. The last section describes the various phases of development leading to the ontology presented in the next chapter.

3.1 Context of Use

The meaning of terms in an ontology is not independent of its context of use, in the same way as the meaning of a word in a sentence is largely determined by the discourse context [Valente et al., 1999b]. This context dependency can be seen as the selection of the relevant properties and relations between all the knowledge one has about a particular concept. This explains to some extent why the more abstract concepts are also the least context dependent one: the less properties there are, the less there is to select. Despite that, there is a remarkable lack of agreement on the major structure and terms in top ontologies (see Section 4.2) [Chandrasekaran et al., 1999]. This is not only the case in ontology engineering, but it is already the case for more than two millennia in philosophy (metaphysics) (for an overview see e.g. [Sowa, 2000, Chapter 2]). An important reason is that it is hard for us even to make explicit what we know about the most abstract concepts, even when these concepts are also the ones we use most frequently and generally. [1] These concepts like cause, duration, position, belief, process, etc. are so ingrained in our common-sense that they are hardly ever questioned but in philosophy, and recently, in the construction of upper/top ontologies. This is due to the process of knowledge acquisition, i.e. the methodology that is used to make concepts explicit and to model these using a formal language. For that reason, it became important to understand the context of use of the terms in two ways.

[1] This is not only the case for common-sense, but also for highly professional areas of knowledge. We have experienced that the construction of a top ontology for organic chemistry was blocked for a long time by the fact that the (eminent) chemists couldn't agree on the definition of central terms like 'chemical reaction' , 'functional group' , etc. [Castillo-Colaux and Krief, 2006]

- The context of use is in the first place the context of use of law in society. This context provides the sources from which we may obtain the required information. We have identified three different kinds of users of legal vocabularies: 1) the citizen, 2) the legal professional and 3) legal scholars. It should be noted that in legal practice and legal education this first category is usually not considered to be a valid source for legal terminology, while on the other hand law is in the first place a social institution for the citizens. Law pervades everyday life to such an extent that the need for legal information serving for (categories of) citizens has become as much a priority on the political agenda, as the need to enable legal practitioners to handle the vast collections of legal documents. This is reviewed in Section 3.2.

- Ontologies are used in the context of applications, tools and technology. Ontologies are knowledge bases that can provide a number of important services. Five major services and uses, particularly for *LKIF-Core* are described in Section 3.2.2. As most of these five kinds of uses apply to (core) ontologies in general, no special requirements follow from LKIF-Core. Maybe we should point out that we foresee a specific use of ontologies in reasoning (second role) by the fact that we may be able to construct frameworks from the definitions of highly interrelated, important legal terms, such as deontic qualifications, that allow special inference. However, this is 1) not completely new (but rarely done) and 2) the full implementation is beyond the current resources in Estrella. [2]

Modelling a relatively large collection of legal terms requires a strategy: an approach (Section 3.3). The 'middle-out' approach, used in the development of the ontology is inspired by [Hayes, 1985] and further developed by [Uschold and King, 1995] and [Grüninger and Fox, 1995]. We have collected terms that were viewed as 'basic-for-law' from representatives from all three groups as they participate in or related to Estrella. To obtain some selection on the 'basic-ness' of the concepts we had these assessed by partners in terms of legal relevance, abstraction and relevance for the core ontology. This is reported in Section 4.1.1. This approach resulted in a number of clusters, or modules, described in Section 5. The initial clusters were specialised and assessed by including more specific concepts from the collected terms. Working upwards from the middle-ground of the initial clusters results in a more global coherence in the ontology. The latter phase has been inspired by existing ontologies, which are reviewed in Section 4.2; in particular *LRI-Core*, a core ontology for law developed in the fifth framework project eCourt [Breuker et al., 2004b] has been our main inspiration (as also specified in the Estrella TA). *LRI-Core*, and the underlying principles are described in Section 4.2.1. [3]

[2] There are no resources/Tasks for this in the TA. This approach was not foreseen, but is the result of the insights obtained in (the discussions in) WP-1.

[3] This core ontology could not be used directly, as the work was not finished in eCourt and it also contained unchecked structures. We needed to clean and adapt this ontology to structure the LKIF clusters (modules).

3.2 Perspectives and Requirements

The 'use-context' or perspective provides constraints on the knowledge acquisition for *LKIF-Core*. These constraints can be specified as answers to the following questions:

- Elicitation. Given an area of practice or use, which documentation and/or kinds of expert-consultants can be identified as sources for selecting terms and obtaining understanding of their meaning and use?
- Modelling. How are the selected concepts used, i.e. in what kinds of tasks and in what roles are the terms selected used and to be used.

The answers to these questions have two purposes:

1. There should be a good correspondence between the use of terminology in current practice and the ontology: the ontology should be an ecologically valid one.
2. The prospective use of the ontology put demands on the 'semantic deepness' of the ontology. For instance, in many forms of information retrieval support, an ontology is used as a structured vocabulary, and there is no or hardly any inference competence required. However, if it is used to support reasoning and problem solving, we need a semanticly 'deep' ontology, i.e. one with well defined properties of classes as a basis for making inferences (see [Hayes, 1985]).

In Section 3.2.1 we will discuss the three populations of 'users' of law: citizens, legal professionals, and legal scholars, providing information about current human use of legal knowledge. In Section 3.2.2 we present five different uses of legal (core) ontologies, and in particular the (potential) use of *LKIF-Core* in Estrella.

3.2.1 Roles and tasks of legal agents

The three identifiable kinds of users of legal knowledge may not only have different vocabularies and perspectives, but they have certainly different tasks and problems to solve with respect to the law.

Law for the citizen

The largest population of users of law are citizens. As behaviour in society is more and more constraint by legal constraints and procedures, the legalisation of social life has forced citizens more and more to think in legal terms. There may be not a clear view on what law *is*, and what its justification and implementation in legal institutions and practice is, but they are aware of the fact that law is *about* their domains of practice, ranging from daily practice (e.g. traffic) to the rules of democracy and to their professional life. In fact, it will be difficult to find domains of common practice of which law has nothing to say. This grounding in common (-sense) domains of law makes that professional legal practice is still intimately related common-practices. We may expect legal knowledge to have many roots in common-sense. This is not only confirmed in studies on the terminology in legislation, but also in earlier attempts in constructing a legal core ontology [Breuker and Hoekstra, 2004a].

Citizens may not be interested or involved in the process of making law (both in the sense of legal drafting and of the debate of cases in courts), they have to deal with the *sources* of law, and in particular with legislation. Besides the fact that they should be aware of a legal context of their activities, and problems about access to these sources, the first task is to *understand* legislation. Even if this understanding is mediated by legal professionals (lawyers, accountants, civil servants, etc.) there remains the requirement of a basic understanding of the constraints imposed by law. The problem is however, that legislation does not *explain*. A legislation assumes that the reader is familiar with the domain the law is about. It appears to be superfluous to explain to the layman what traffic is, even if the legislator has a particular view on traffic. [4] Law about a particular domain is written under the assumption that the addressee is well acquainted with the domain itself. This assumption may be correct, but it means that also the legal drafter/legislator has to have a (good) understanding of the domain: they have to share a common understanding:a common sense. This understanding is facilitated by the fact that the norms in a regulation are cast in such a way that they describe some situation in generic terms and specify what is obliged, forbidden, or permitted about it.

This understanding of legal constraints provides also constraints for the *planning* of activities. In legal drafting this planning view of citizens has only recently (i.e. over the last decades) been made explicit. Legislators and their legal drafters have become aware of the fact that planning may involve more than simple compliance with law. It may mean changes of plans that follow alternative, more 'costly' routes (e.g. tax evasion), illegal but hidden steps etc. This may make the relationship between legislator and citizen look like the evaluative developments from conflicting interests and symbiosis between organisms and parasites. The benefits of semi- or not complying with (the intentions of) the law may involve the use of legal professionals who are not part of the legal system itself (courts, solicitors, government), but who may even plan legal actions for the pure private benefit of it. As a reaction, legislation and precedent law may become more and more in-transparent, as justification and explanation are not obligatory parts of legislation. This is also where intelligent legal information serving may help to (re-)establish some balance. Not only may these tools provide the less resourceful citizen with the knowledge and know-how of more resourceful citizens, but also it allows the legislator to check more or less exhaustively the potential reasonable actions of citizens. Of course, not in all domains of law rationality explains the legal and the illegal behaviour, but it certainly does to a large extent.

Tasks of the legal system

The legal system is the executive organisation of law in society. It consists of a social organisation and a culture. The culture – knowledge and know-how – consists of

[4] For instance, traffic regulations have a very reduced view on time (only 'moments') and space (reduced to positions of parts of the road), while for the citizen-driver, traffic is about real time (duration and moments) and real space (distances).

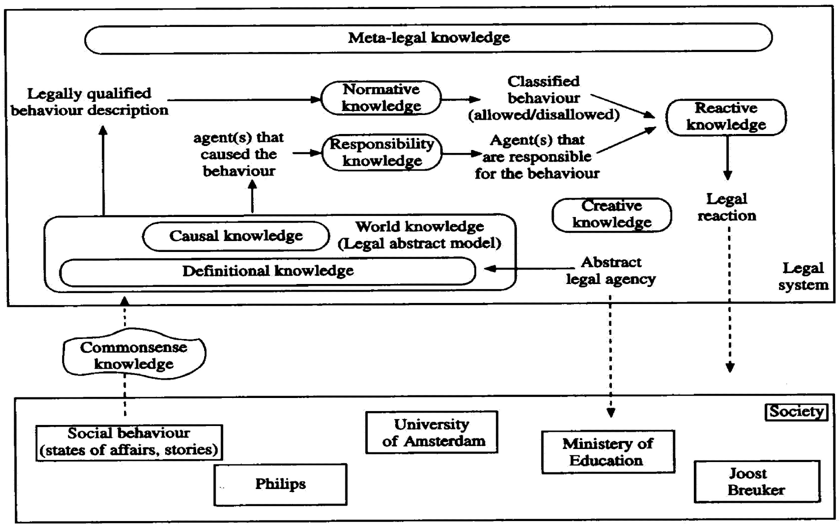

Figure 3.1: FOLaw: a functional ontology for law [Valente, 1995]

legal sources and methods. The legal organisation consists of legal roles (e.g. judge) and sub organisations. It has the power to assess the legitimacy of social behaviour and in that context to solve conflicts between citizens and between citizens and government. Therefore the major task of the legal system is *assessment* of legitimacy in the control of legal compliance in society. [5] This feed back function is illustrated in Figure 3.1. This Figure represents a schematic view of what dependencies between kinds of legal knowledge are involved in assessing legal cases. The assessment task is split in two streams: assessing whether there is some violation of norms, and the answer to the question who can be held responsible for the violation(s) (for a full description see [Valente, 1995] or [Valente et al., 1999b]). Within these kinds of knowledge we can distinguish two crucial clusters:

- For assessing trespasses of law *case descriptions* are compared with *norms.*
- For assessing who can be held responsible one needs a *causal/intentional* account of the events of the cases matched against legal notions about *guilt, liability* etc. This causal account involves also notions of *action, intention, prediction,* etc.

Representing legal decision making over cases as a two streams assessment task simplifies what is actually going on in courts and in legal professional activities. The assessment takes place in the context of debate in which different interpretations of cases and the applicability of legal terms (norms, principles) are disputed. This debate characterises legal practice to such an extent, that legal professionals see law rather as evolving insights and knowledge due codified in precedent law. For that reason this deliverable includes an account of what is involved in debate, i.e.

[5] The legal system itself does not perform the full control cycle, as for instance there are additional functions for the police and other civil servants.

dialogical *argumentation.* A framework for argumentation is proposed and described in Section 5.3.

Besides this major task, the legal system – or at least legal professionals – are involved in drafting regulations. Their role is to translate political aims and intentions into legislation that fits the conventions and functions of the legal system. Legal drafters are an interface between the world of legitimated power (e.g. a government) and the legal system). [6] One may say that this interface is an interface to a larger control cycle: the cycle of political control in a democratic society.

Jurisprudence perspectives

Legal concepts in legal doctrine and jurisprudence. An essential aspect of both legal doctrine and jurisprudence consists in the study of legal concepts. In particular, legal doctrine analyses what the sufficient or necessary preconditions are for the application of a certain concept (for instance, tort, self defence, homicide, good faith, contract, and so on) and what consequences follows from its applications. It decomposes legal concepts in their constituent elements (for instance, a tort may be decomposed into an action, a damage caused by the action, the fact that the action violates the law, the fault of the author, etc.), it studies commonalities and differences between different legal concepts etc. This work is intended to help legal practitioners who will refer to legal doctrine when applying the law to individual cases (namely, when interpreting cases or qualifying cases according to concepts). The analyses performed by legal doctrine are usually evaluative, in the sense that the legal scholar will propose certain characterisations of legal concepts knowing what inferences will then be derived from the concepts, and thus will frame legal concepts in such a way that these inference are legally justified and lead to outcomes corresponding to legal values. Legal theory usually builds its concepts on the top of legal doctrine, the main difference being that the conceptual analysis of legal doctrine are generally tied to the language of legal sources (and doctrinal analysis is meant to indicate the meaning of the occurrence of certain terms in legal sources) while legal theory also characterises terms that are not present in legal sources.

Integration or choice?

The question is now whether we should choose one of these three perspectives to inspire *LKIF-Core*, or whether they are sufficiently compatible to allow some integration. The basic assumption we start the modelling with is that the perspectives of citizens and the legal professionals can be more or less integrated for the following reasons.

- Law is *about* domains of social practice and therefore a lot that legal norms refer to are 'common-sense', or at least not law specific terms, but the technical or common sense terms of some field of practice. Of course this does not apply to norms for the legal system itself.

[6] Their task is somewhat analogous to that of a knowledge engineer who has to translate 'expert' knowledge into some technical vocabulary.

- In assessing legal cases, the normative and responsibility knowledge involved has only a small number of concepts which be unknown or less precise in common sense. A good example is the notion of liability.

For the terminology of legal scholars a special role is indicated. Their concepts are usually not typical common-sense, but these concepts are not only very abstract, due to the reflective role of legal scholars, but their role is often to classify the terms typically used by legal professionals, and may therefore very well appear in the top of a legal core ontology.

In summary, it appears that from all three kinds of users of legal knowledge we may acquire knowledge for the construction of the *LKIF-Core* as the three 'perspectives' appear to be compatible rather than containing conflicts.

3.2.2 Roles of *LKIF-Core*

We have identified five main uses or roles for ontologies: (a) organise and structure information; (b) reasoning and problem solving; (c) semantic indexing and search; (d) semantics integration and interoperation; and (e) understanding the domain. Specifically in AI & Law, these uses have manifested as follows:

Organise and Structure Information In this role ontologies organise and structure information in the domain. Ontologies here are tools to describe things or phenomena in the domain of interest. The ontology thus plays the role of vocabulary, answering two main questions: (a) which terms can be used? (i.e., ontology as a lexicon); and (b) which valid sentences can be expressed (i.e. ontology as a semantics). Typical examples are professional terminologies: for reasons of maintainability and use in (semantic web based) automated transactions, these terminologies are being transformed or labeled now as ontologies, in particular when cast in OWL [Stevens et al., 2004, Kashyap and Borgida, 2004]. In AI & Law, this role is shown in the use of ontologies to define legal vocabularies. In this way, the ontologies are not so much really legal ontologies, that contain a typical legal vocabulary, but the universe of discourse of the world or domain the law is working on, e.g. taxes, crime, traffic, immigration, etc. Two examples of this use are the Jur-Wordnet ontology [Gangemi et al., 2005] and CRIME.NL, an ontology of Dutch criminal law that was constructed with the specific aim to be re-used and adapted for Italian and Polish criminal law [Breuker et al., 2004a].

Top/core ontologies like *LKIF-Core* have an important role in pre-structuring the development of a domain ontology (see Figure 1.1). They allow for top-down knowledge acquisition whereby many/most/all domain specific terms can be sub-classes or specific combinations of more abstract terms. For instance, the definition of 'norm' in *LKIF-Core* provides the structure of the specific norms in some legal domain. In fact we have followed to some extent a top-down approach in developing *LKIF-Core* itself by using *LRI-Core* as a top structure to collect the various modules developed in a middle-out approach (see Section 3.3)

Reasoning and problem solving The basic role of ontologies in this case is to represent the knowledge of the domain so that an automated reasoner can represent problems and generate solutions for these problems. The ontology here works as a terminological part of the knowledge base to enable the understanding of assertions about the problem situation to be solved. Ontologies here are not only constructed for a particular knowledge system, but as ontologies contain generic knowledge, cost-effective knowledge engineering may benefit from its re-use potential. Indeed, one can argue that the use of ontologies in AI comes from research in the late eighties and nineties that aimed at improving knowledge engineering by creating "well-structured" knowledge bases that would not only solve the problem at hand but be more maintainable, easier to extend, etc. In this sense, ontologies are then very much an engineering tool. This role of ontologies implies the use of an inference engine that is used to conclude specific goals. An interesting problem that arises is the introduction of an inference bias. [Valente et al., 1999b] show that ontological choices are strongly influenced by the purpose of the ontology. That is, the same knowledge will be structured or formalised differently depending of how it will be used by the reasoner in reaching the desired conclusions in a specific context. This indicates that reusability is a good idea, but it can never be accomplished completely.

A more specific use in Estrella of the *LKIF-Core* is the fact that a well defined ontology of interrelated terms can be the basis of a special *calculus* (inference). If terms get definitions in which their mutual relations and exclusions from which all tautologies can be generated by the use of an inference engine. For instance, we can, like in the axioms of deontic logic, write the relationships between terms that provide the deontic qualifications like obligation and permission in norms. A reasoner can use these axioms for instance to detect exceptions in a regulation consisting of norms. [7]

Semantic Indexing and Search The basic role of ontologies in this case is to represent the contents of documents or other "soft" information sources (picture, movies, etc.). The ontology here works as the semantic index of information that enables semantic search for content. Law and legal practice produce vast amounts of knowledge in the form of documents, charts, schemes, etc. There is a key need to organise and be able to find these documents. Ontologies can be used to represent and search semantically the content of documents to go beyond word or keywords. The traditional example that shows the need for this use of ontologies is the existence of multiple meaning of words. Ontologies can also be used in a more intentional way, as a mechanism for creating annotations, i.e. allowing a person to semantically mark content so it can be found later. An example of this use is the work of [Benjamins et al., 2004]. In fact this use of ontologies in managing and accessing legal information will probably be a primary use of legal ontologies. The majority of applications of

[7] In fact, as is explained in [Boer et al., 2007]. and in Appendix Section A.4, due to the special status of 'permissions' that have most often the role of exceptions, a simple calculus is not possible if applied to norms. It has to be combined with a reasoning architecture.

ontologies as resources for the Semantic Web have this role, and particularly for legal domains with its overwhelming amount of documented sources, this kind of use is indicated for such important roles as e-Government and access to cases of precedent law. Moreover, these ontologies also allow by its semantic base to change information retrieval as currently provided by search engines into real *legal information services*, where users can negotiate there needs and will receive tailored information – even answers to their questions – instead of collections (parts of) potentially relevant documents.

Semantic Integration/ Interoperation The basic role of ontologies in this case is to support applications to exchange information electronically. The ontology here works as an inter-lingua that defines a (narrow) vocabulary to be used to interchange information. This use is less common in the legal domain, but there is potential for use in law enforcement, e.g., organisations exchanging information about criminals. There is also a lot of use in quasi-legal situations such as in complex systems in large bureaucracies that need to interoperate (e.g., the European Union). Since they can be seen as a semantic information schema, these ontologies may reuse parts of ontologies created for other uses. In the context of Estrella, this role is particularly relevant when it concerns the interoperability of knowledge bases of the vendors. The exchange format provided by LKIF as a base inter-lingua, is only a formal language mapping, by which formal categories of a vendor language are mapped onto those of LKIF to map these back to the formalism of another vendor. Aside from the fact that in this two step *translation* mismatches may occur twice, an ontology in an LKIF language (OWL-DL 1.1) may help to maintain the integrity of the semantics in this process. Of course, this means that for the specific domain an ontology is developed (using *LKIF-Core* as explained above).

Understand a domain The basic role of ontologies in this case is to provide a view of what a domain is about to try to make sense of the variety of knowledge in that domain. The ontology here works as a map that specifies what kinds of knowledge can be identified in the domain. This type of ontology can be used as a basis for designing specialised representations. Because it tries to get close to the nature of the domain, it frequently connects and draws from theories of that domain (e.g., theories of law). These types of ontologies have been called core ontologies. An example of this type of ontology is the Functional Ontology of Law (FOLaw: [Valente et al., 1999a]).

3.3 Middle-Out Ontology Development

In order for the *LKIF-Core* ontology to meet the requirements described in the previous sections, its construction follows a combination of methodologies for ontology engineering. Already in the mid-nineties, the need for a well-founded methodology was recognised, most notably by [Gruber, 1994, Grüninger and Fox, 1995, Uschold and King, 1995, Uschold and Grüninger, 1996] and later [Fernández et al., 1997]. These methodologies follow in the footsteps of earlier experiences in knowledge acquisition, such as the CommonKADS approach ([Schreiber et al., 2000]) and others, but also considerations from naive physics and cognitive science, such as [Hayes, 1985] and [Lakoff, 1987], respectively.

[Hayes, 1985] describes an approach to the development of a large-scale knowledge base of naive physics. Instead of rather metaphysical top-down construction, his approach starts with the identification of relatively independent *clusters* of closely related concepts. These clusters can be integrated at a later stage, or used in varying combinations allowing for greater flexibility than monolithic ontologies. Furthermore, by constraining (initial) development to clusters, the various – often competing – requirements for the ontology are easier to manage.

Whereas the domain of [Hayes, 1985]'s proposal concerns the relatively well-structured domain of physics, the combination of commonsense and law does not readily provide an obvious starting point for the identification of clusters. In other words, for LKIF-core, we cannot carve-up clusters from a pre-established middle ground of commonsense and legal terms. Furthermore, the field does not provide a relatively stable top level from which top-down development could originate. In [Uschold and King, 1995], who are the first to use the term 'middle-out' in the context of ontology development, it is stressed that the most 'basic' terms in each cluster should be defined before moving on to more abstract and more specific terms within a cluster. The notion of this basic level is taken from [Lakoff, 1987], who describes a theory of categorisation in human cognition. In the context of ontology engineering [Uschold and King, 1995] deem the following ideas of importance ([Lakoff, 1987, p. 12 and 13]):

Basic-level categorisation Categories are organised so that the categories that are cognitively basic are 'in the middle' of a taxonomy. Generalisation proceeds 'upwards' from this basic level and specialisation proceeds 'downwards'.

Basic-level primacy Basic level categories are functionally and epistemologically primary with respect to (amongst others) knowledge organisation, ease of cognitive processing and ease of linguistic expression.

Functional embodiment Certain concepts are not merely understood intellectually; rather, they are used automatically, unconsciously, and without noticeable effort as part of normal functioning. Concepts used in this way have a different, and more important psychological status than those that are only thought about consciously.

[Uschold and Grüninger, 1996] describe the following elements for a 'skeletal' methodology for ontology development. Although their paper does not present these elements as phases in ontology development, but rather as characteristics of the methodology itself, these elements apply to the development of a single ontology as well. For the purpose of the LKIF ontology, we have made slight adjustments to this framework.

Guidelines for each Phase establish guidelines for each development phase. [Gruber, 1993, Uschold and Grüninger, 1996] describe a number of design criteria we adhere to in the development of the LKIF ontology:

Clarity An ontology should be as unambiguous as possible, distinctions between concepts should be motivated, and examples should be provided to enable the user of the ontology to better understand formal definitions and less strictly defined concepts. Definitions should be documented using natural language.

Coherence The ontology should be (at least logically) consistent.

Extensibility The ontology should be designed to anticipate future extensions, for a range of anticipated tasks. The terms included in the ontology vocabulary should be usable for defining new terms, in a way that does not require the revision of existing definitions

Minimal ontological commitment 'An ontology should require the minimal ontological commitment sufficient to support the intended knowledge sharing activities'.

Minimal encoding bias Concepts should be specified at the knowledge level, without too much bias to some symbol-level encoding, e.g. constructions for convenience of notation or implementation.[8]

Documentation It may be desirable to have guidelines in place for developing the documentation during the development phases. F

Identify Purpose and Scope in this phase the requirements for the ontology are identified: *why* the ontology is developed, its possible *use* and *users.*

Building the Ontology The building phase consists of three closely related tasks, which can be alternated as need arises:

Ontology Capture is the identification of key concepts, the production of precise definitions in natural language of concepts and relations, and the identification of terms to be used to refer to these concepts.

- Produce all potentially relevant terms by means of a brainstorming session.
- Group the terms loosely into work-areas, corresponding to naturally arising sub-groups, i.e. clusters.

Ontology Coding is the representation of these concepts and relations in a formal language. This involves commitment to a particular representation formalism.

[8] This restriction is for a large part overcome by adhering to the OWL representation formalism

- Address each work area in turn, or divide work across the clusters
- Proceed in a middle-out fashion. Address the most fundamental concepts first, before moving up or down the hierarchy.

Integration with Existing Ontologies involves the identification and integration of existing ontologies into the results of the previous two steps.

Evaluation of the ontology, as it fulfils the requirements formulated in the second phase.

The following chapter describes how the building and integration phases have materialised in the context of *LKIF-Core*. Furthermore, a process for future evaluation is outlined.

Chapter 4

An outline of *LKIF-Core* – The Methodology Applied

This chapter describes how the methodology from the previous chapter was applied to the development of *LKIF-Core*. We first describe the building and clustering phase, followed by an overview of the ontologies under consideration for inclusion. The chapter ends with a description of follow-up evaluation steps.

4.1 Building and Clustering

4.1.1 Collecting and scoring legal terms

The *LKIF-Core* ontology should contain 'basic concepts of law'. As explained in Section 3.2.1 it is dependent on the (potential) users what kind of vocabulary is aimed at. We have identified three main groups of users: citizens, legal professionals and legal scholars. Although legal professionals use the legal vocabulary in a far more precise and careful way than laymen, it appears that for most of these terms there is still a sufficient common understanding to treat them more or less as similar [Lame, 2006], except for a number of basic terms that have a specific legal-technical meaning, such as 'liability' and 'legal fact'. We have added terms from legal scholars (legal philosophers) because they may have acquired insights not only in the 'essential', abstract meaning of terms in law, but also because their terms may be used as abstract organisers of legal terms.

The Estrella consortium includes representatives of the three kinds of experts. Every partner was asked to supply their 'top-20' of legal concepts. Combined with terms we collected from literature (jurisprudence and legal text-books) we obtained a list of about 250 terms. As such a number is unmanageable as a basic set for modelling, we asked partners to assess each term from this list on a number of scales. We identify the following five scales.

1. level of *abstraction* (1= highly abstract... 5 = very concrete)
2. *relevance for legal domains* (1= highly relevant 5= not relevant at all)
3. degree to which a term is legal rather *common-sense* (1= exclusively legal 3= both used as a legal term and in a common-sense sense 5= the common-sense meaning is also the meaning when used in a legal context)

4. degree to which a term is a *common legal term* vs. a term that is specific for some sub-domain of law (e.g. copyright law, tax law, ...) (1= common ... 5=specific to subdomain). Note that for this dimension you may have to use the 0 (e.g. for a number of 5 scores on scale 3.).

5. degree to which you think this term should be *included in the ontology* (1= really important, 5 = not important at all)

Five partners returned the list with scorings. The average values for the 'top-50' terms of the three major scales are presented in the Appendix (Section A.1). These 50 terms plus those re-used from LRI-Core (see Section 4.2) have been the initial set for the representation of the *LKIF-Core* ontology.

4.1.2 Conventions

As *LKIF-Core* was developed by a heterogeneous group of people, we specified a number of conventions to uphold during the representation of terms identified in the previous phases:

- Words in class and property names are separated by an underscore (*Some_Named_Class* and *some_property*).
- Naming convention for properties:
  - Property names are lowercase, e.g. *property*.
  - Nouns are preferred over verbs.
  - If a verb is used, we use third person singular active form (*verb+s*).
  - Properties *never* have the prefix 'has' , e.g. *has_noun* should be *noun*.
- Naming convention for inverse properties:
  - If a word exists for the inverse of a property, we prefer to use that word.
  - If the property identifier is a noun, the inverse is "*noun_of*", "*noun_in*", "*noun_for*", etc.
  - If the property identifier is a verb, the inverse is "*verbed_by*".
  - Note that for some words, constructing the identifier of the inverse property as described above might result in a rather strange name. Sensible names are preferred.
- Naming convention for class names:
  - Each word in a class name starts with a capital; other letters are lowercase, e.g. *Class*).
  - Class names never have the suffix 'Thing', e.g. *Interesting_Thing* should be *Interesting*.
- Classes and properties should be provided an adequately descriptive comment in the form of an *rdfs:comment* field.

- Classes should be represented using necessary & sufficient conditions, as much as possible.

4.1.3 Clustering

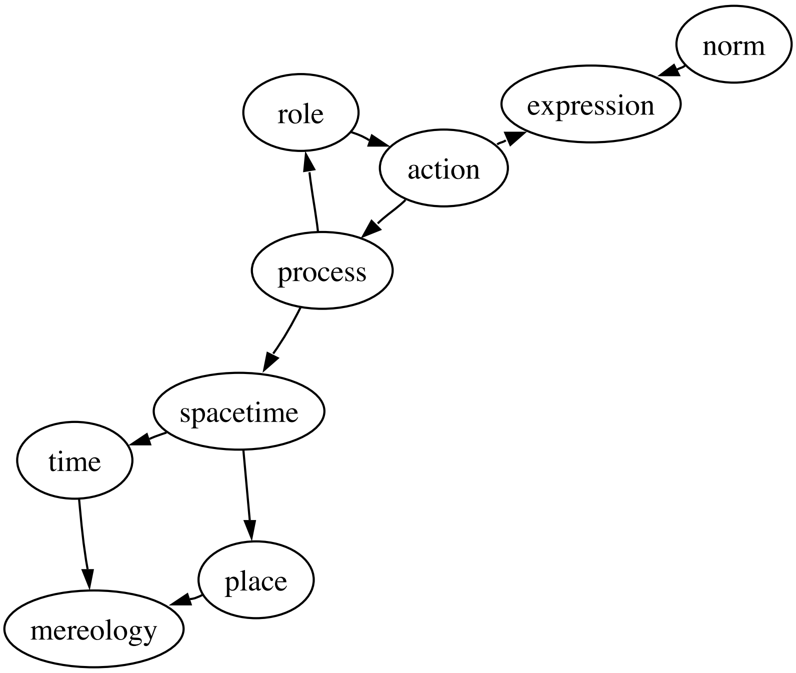

Figure 4.1: Top-level clusters

The list of terms and insights from the requirements-phase resulted in a set of initial ontology modules specified in OWL-DL, using TopBraid Composer (`http://www.topbraidcomposer.com`) and Protege 3.2 (`http://protege.stanford.edu`).

The following list of original clusters refers to the sections in the Appendices where the text can be found.

Expression covers a number of representational primitives necessary for describing relational mental states that connect a person to a proposition: e.g. beliefs, statements etc. (see Section 4.1.3)

Norm defines the concepts most central to LKIF: e.g. norm, obligation, prohibition etc. (see Section 4.1.3)

Processes describes concepts related to (involuntary) change (see Section 4.1.3)

Action covers concepts related to actions and their relation to physical change and intentions, e.g. action, agent etc. (see Section 4.1.3)

Role describes constructs that underlie the roles being played by agents (see Section 4.1.3)

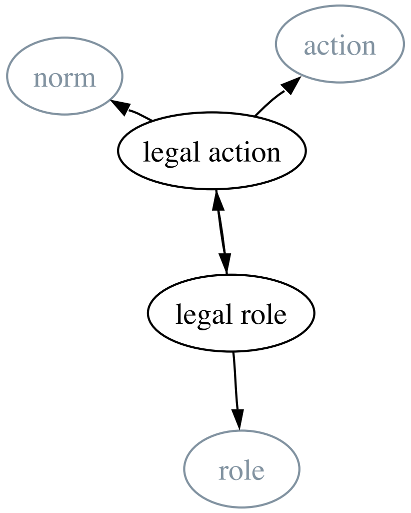

Figure 4.2: Legal clusters

Place defines representational primitives for describing places, locations and the relations between them (see Section 4.1.3)

Time covers representational primitives for describing time intervals (see Section 4.1.3)

Mereology describes classes and properties that allow to express mereological relations, e.g. parthood, components etc. (see Section 4.1.3)

Figure 4.1 shows the structure of the dependencies between these modules (note that the imports relation is transitive). At a lower level of abstraction, the clusters are used to specify frameworks for typical legal actions and roles (see Figure 4.2). The concepts in these clusters were formalised in a middle-out fashion: for each cluster the most central concepts were represented first. The sections 4.1.3-4.1.3 describe each of these initial clusters in some more detail (see also [Breuker et al., 2006]). The appendices contain a formal description of a number of the initial modules (Section A.2).

As a consequence and due to discussions, further literature study and importing already existing modules, the terminology of the original set of clusters has slightly changed. We have maintained the original views that were used in identifying the clusters, as the explanations and justifications are still valid and applicable to the current version of *LKIF-Core*, reported in the next chapter.

Place Module

The initial ontology for places is based on the work of [Donnelly, 2005], who uses Newton's distinction between relative places and absolute places. The location of a relative place is defined relative relative to some object. The location of an absolute place, on the other hand, is not defined relative to some other place. An absolute place is part of absolute space, it has fixed spatial relations with other absolute places.

As the spatial relations a relative place has with other places can change over time, many of the relations defined by Donnelly have a temporal component: i.e. they are indexed in time. We choose not to do so, as OWL does not allow for ternary predicates. One of the positive side effects of temporal aspects not being in the scope of the relative places cluster is that the ternary relations between two places and a moment described in Donnelly's paper are reduced to binary relations.

The most fundamental concept in this cluster is PLACE, it can be split into ABSOLUTE_PLACE and RELATIVE_PLACE as discussed earlier this section. A LOCATION_COMPLEX is a set of places that share a reference location. The spatial relations between objects in a single location complex are more stable than spatial relations between two arbitrary relative places.

The *meet* property is used to define many of the other properties. Place A meets place B when there exist a point $a \in A$ and a point $b \in B$ such that the distance between a and b is 0. We assume that the meet relation is the most basic one and will not define distance. Using *meet* we can define the following subproperties: *abut*, *partially_coincide*, *cover*, *covered_by*, and *exactly_coincide*. The exact relations between these properties are specified as expressions over points contained in the places. Unfortunately we cannot express this in OWL-DL

RELATIVE_PLACES are defined as places having some value for the *reference* property. The property *reference* is used to define *reference_of*, *location_complex*, *location_complex_of*, *relatively_fixed*. The combination of *meet* and *reference* is used to define *connect*.

This initial ontology of places does not include concepts about direction and orientation. Being able to express properties like *in_front_of*, *left*, etc. might be useful. An other option might be to express direction and orientation in degrees.

Time Module

The initially proposed ontology for time is based on the work of Allen[Allen, 1984, Allen and Ferguson, 1994]. As the purpose of this ontology is to be useful in client applications, rather than providing a full time calculus, we have chosen to impose as few restrictions as possible on these client applications. As the ontology is represented in OWL-DL, we have to cope with its limits of expressiveness.

The ontology distinguishes between two basic concepts, INTERVAL and MOMENT. Intervals have an extent (duration) and contain (sub-)intervals and moments. Moments are points in time and therefore have no duration and do not contain other moments or intervals. The super class of intervals and moments is called TIME_PERIOD.

The relations between periods are what defines time. Allen uses the relation *meet* to define all other relations. Allen's meet relation differs from Donnelly's meet relation, as the spatial meet relation is symmetric and Allen's temporal meet holds only in one direction. To avoid confusion with the spatial meet relation, we use the property *immediately_before*. This property allows us to define *before*, *immediately_after*, *after*, *between*, *during*, *finish*, *overlap*, *precedes*, *span*, *start*, and *transition_point*. Allen uses set-theory, predicate logic, and natural language to define these relations. As we are restricted to OWL-DL, this implies that not all relations can be completely defined.

The limited expressiveness of OWL-DL conflicts with the requirement to define the concepts well. To solve this, the ontology is divided over the files `time.owl` and `time-rules.owl`. The file `time.owl` contains the part of the ontology expressible in OWL-DL. The file `time-rules.owl` extends this with axioms only expressible in SWRL or in OWL full. The file `time-test.owl` contains tests for checking that the ontology is define correctly and results in the expected inferences.

Within the W3C, the Semantic Web Best Practices and Deployment Working Group is working on ontology design patterns. They are working on "Time Ontology in OWL" [Hobbs and Pan, 2006]. There are some issues with this W3C draft which render it not immediately suitable for our use. However, a future mapping between the W3C time ontology and LKIF Core is possible. Client applications can choose to use this mapping.

Mereology Module

The mereology module contains a set of concepts shared both by relative places and time. Mereological relations allow us to express a systems oriented view on concepts. The module contains concepts and properties like PART, COMPONENT, SYSTEM, *contain*, *part_of*, etc. Some properties inherit over some *part_of* relations and others will not. Unfortunately, we cannot express this in the current version of OWL-DL. However, the next version of OWL-DL, OWL 1.1, will allow these constructs to be expressed.

Process Module

The **process module** describes concepts related to involuntary change. It relies on descriptions of **time** and **place** for representing the duration and location of changes. This module forms the basis of definitions of dynamic concepts, i.e. concepts that capture *change* such as actions, tasks, inferences, workflow, processes, causation, history, stories. A change is a transition that brings about a difference between the situation before and after the change occurs. Changes that occur according to a certain recipe or procedure, i.e. changes that follow from causal necessity are *processes*: processes introduce causal propagation. Processes are bound in time and space: they have a duration, and take place at a time and place. A process can be a functionally coherent aggregate of one or more (sub) processes. We furthermore distinguish *physical* processes which operate on *physical objects*.

The process module does not commit to any view as to what the propositional content of a situation might be. Furthermore, at this level we do not commit to a particular theory of causation or causal propagation. The **process module** and **action module** are closely related.

Action Module

The **action module** covers both the addressees of regulations and the behaviour that is to be regulated. It extends the **process module**, which has a causal perspective, with intentional concepts such as *agent*, *action* and *artifact*.

An agent is intentional and can be an *actor* in an action. It is the medium of expressions such as beliefs, intentions, observations etc. Both *organisations* and *persons* are agents. Persons are individual agents, an organisation is a group of other organisations or persons which acts 'as one'. Artifacts are physical objects *designed* for a specific purpose, for some function as instrument in a specific set of actions.

An action is a change which is brought about by some agent, playing a role. The agent is the medium of some intended outcome of the action: an action is always intentional. The intention of the agent has, usually bears with it some expectation that the intended outcome will be brought about: the agent believes in this expectation. The actions an agent is expected or allowed to perform are constrained by the the *competence* of the agent, sometimes expressed as *roles* assigned to the agent. The **action module** does not include an elaborate characterisation of action frames and thematic roles.

Role Module

Roles play a pivotal part in a teleological perspective on the world. Not only do roles allow us to categorise objects according to their use (i.e. their *function*) and intended behaviour, they also provide the means for categorising the behaviour of people around us. Roles are a necessary part of making sense of the world: it allows for describing social organisation. Because roles are idealisations of behaviour, players can play a role incorrectly.

Roles are behavioural requirements on role execution and on qualifications of role taking. These requirements are prescriptions, i.e. they are normative. In modern society many roles have formal requirements enforced by law. Legislation addresses actors by the roles they play[1]. If actual behaviour deviates from the norms attached to these roles, we violate the law. Violations are based upon the distinction between the prescription (role) and role performance. Therefore, in court, it is the actor of the role who is made responsible: as a person; not as a role. Even the fictitious concept of legal-person for social organisations turns into concrete responsibilities of the liable persons who have mis-performed their roles.

The notion of roles has played an important part in recent discussions on ontology. Most notable in this respect are the work of [Steimann, 2000, Masolo et al., 2004], and the Ontoclean methodology ([Guarino and Welty, 2002]). In [Loebe, 2003], roles are categorised according to their context. This results in the identification of three types of roles:

Relational Role The arguments of a relation, i.e. the domain and range of a property in OWL. Prototypical examples of relational roles are usually taken from the family domain: *father, mother, parent, child* etc.[2] Relational roles are usually not only bi-directional but complementary as well. This nature of roles might explain the complimentarity of rights and duties.

[1] An exception to this rule is in criminal law.

[2] In many cases these roles are presented as *father_of, mother_of* and so on. However, we prefer a naming convention which emphasises the role instead of the relation. This, amongst others, allows for a more generic representation of the semantics of the roles/relations themselves (i.e. using subPropertyOf constructs)

Participant Role The role an entity plays with respect to a certain process taking place. Examples are the *speaker* and *hearer* in a *telling* process. [Loebe, 2003] describes functions of devices as participant roles. *Epistemological* roles are participant roles.

Social Role Roles which exhibit their own properties, socially: *student, professor* etc. Social roles are normative prescriptions of behaviour: they enable or prohibit the role-player to participate in processes, i.e. to fill certain prescribed participant roles. Social roles are related to social 'positions'. Law addresses agents mainly at the level of social roles.

Roles can be divided into two categories: relations and concepts. Roles as concepts specify required properties and behaviour. Role relations are

The **role module** captures the *roles* and *functions* that can be played and held by agents and artefacts respectively. Roles and actions are closely related concepts: a role defines some set of actions that can be performed by an agent, but is also defined by those actions. The role module focuses on *social* roles, rather than traditional thematic or relational roles ([Steimann, 2000, Loebe, 2003]). This allows us to define roles as classes.

Expression Module

Legal reasoning is based on a common sense model of intelligent behaviour, and the prediction and explanation of intelligent behaviour. It is after all only behaviour of rational agents that that can be effectively influenced by the law.

The **expression module** covers a number of representational primitives necessary for dealing with *propositional attitudes* (viz. [Dahllöf, 1995]). A propositional attitude is a relational mental state connecting a person to a proposition. Many concepts and processes in legal reasoning and argumentation can only be explained in terms of propositional attitudes, and even the norm is defined in this ontology in terms of propositional attitudes.

The full details of the model of intelligent behaviour implicit in legal reasoning is exposed best in the area of law that demands the greatest precision. This is undoubtedly criminal law.

Mens rea is the Latin term for "guilty mind" used in criminal law. The test of criminal liability is expressed in the Latin phrase *actus non facit reum nisi mens sit rea* or "the act will not make a person guilty unless the mind is also guilty". So there must be an *actus reus* accompanied by some level of mens rea to constitute the crime with which the defendant is charged.

This is not the case in many smaller violations of the law. The *actus reus* must always have taken place, but the guilt of the mind of the actor, and sometimes even the identity of the actor responsible for the violation, can be completely irrelevant. A parking or speeding ticket is for instance in many countries directly issued to the registered owner of the car involved, regardless of whether the owner was involved in the violation.

Although there are significant differences between jurisdictions in the levels of "guilty mind" recognized, there appears to be one constant: each distinguishes:

1. facts,
2. intentions, and
3. expectations.

Each of these have a *propositional content* (are about something), and a *violation must* be present on one or more of these levels. The violation must be at the very least present in the facts. The second test is whether the violation was, or should have been, foreseen by the involved agent(s), or in other words whether it is part of the expectations of the agent. The third test is whether the violation that took place is also present in the intentions of the involved agent(s).

More subtle distinctions can be made: the difference between *direct criminal intent* and *conditional criminal intent* (*opzet bij oogmerk* and *voorwaardelijk opzet* in the Dutch WvS or the difference between *purposely* and *knowingly* committing a crime in the US Model Penal Code) is for instance that to establish conditional intent the crime that was intended need not be the same event as the crime that happened. Think for instance of a car bomb that killed or wounded the wrong victim.

More complicated are the distinctions made in relation to expectations. As a general rule, whatever is intended, is also expected. If you want someone to die and use Voodoo against that someone, but do not expect that someone to die as a result of Voodoo, you do not intend to kill. Not every *desire* is therefore an intention: an intention is acted on, and the actions should be rational way to realize the intention.

Besides the things you intended to happen, there are also a lot of things that could happen as a result of your actions and that you *should have* foreseen. Classical is the distinction between *recklessness* and *negligence* (for instance in the US Model Penal Code) and the notion of the man on the Clapham Omnibus as a test for what should have been expected by a normal and reasonable person, which will not be discussed here.

It is important to note that there is an *internalist* and an *externalist* way to use intentions and expectations. The external observer can only ascribe intentions and expectations to an agent based on his observed actions. The external observer will make assumptions about what is *normal*, or apply a a *normative* standard for explaining the actions of the agent. Intentions and expectations can also be used directly as a model for intelligent decision making and planning[3]. In this case the expectation of a violation of the law should weigh against a candidate decision or plan.

It is obviously also common, when making a plan, to predict the actions of other agents based on the assumption that they will act in accordance with the law. If you drive towards an intersection, and you see another car approaching for the left, and there is a norm which states that one should yield for traffic from the right, one may drive on in the legitimate expectation that the other driver will yield. This use

[3] Regardless of whether it is a psychologically plausible account of decision making. Daniel Dennett's notion of the *Intentional Stance* is interesting in this context (cf. [Dennett, 1987]). Agents may do no more than occasionally apply the stance they adopt in assessing the actions of others to themselves.

of the law as a predictive instrument is as important as its use as an obstacle in decision making. This is also an alternative account of what is called the "directed obligation" in Sartor's [Sartor, 2006]: an obligation to one side in a two-sided action is a *legitimate expectation*[4] to the other side. The teleological explanation also often works, because the legitimate expectation is often the very reason that we act in certain ways. If someone offers a painting for sale for a certain amount of money, and we give that amount of money to the salesman for the painting, then the expectation of receiving the painting is the very reason why we gave the money in the first place.

The file **expression module** defines a number of primitives necessary for classifying the various mental postures one can adopt towards an expression. These mental postures are subtypes of *expression*, which has some *medium* and simply expresses something. *Belief*, *intention*, *observation*, and *expectation* are components of a mental model, and the medium of these expressions is an *agent*[5]. *Statement*, *declaration*, and *assertion* are expressions communicated by one agent to one or more other agents. This classification is loosely based on Searle (cf. [Searle and Vanderveken, 1985]).

The **expression module** is intentionally left underdefined. A rigorous definition of propositional attitudes relates them to a theory of reasoning and an argumentation theory. The argumentation theory is supplied by the **argumentation module**. The theory of reasoning depends on the type of reasoning task (assessment, design, planning, diagnosis, etc.) LKIF is used in, and should be filled in (if necessary) by the user of LKIF.

The **expression module** is also used to define norm in the **norm module**.

Of special interest is the qualification, which is used to define norm. We mean here an evaluation, a value statement, value judgement, evaluative concept, etc. I.e. only the type of qualification which is an attitude towards the thing qualified, and not for instance the redness of a rose, as in [Gangemi et al., 2002] and others.

In many applications of LKIF the attitude of the involved agents towards a proposition will not be relevant at all. Fraud detection applications will only care to distinguish between potentially contradictory observations or expectations relating to the same propositional content.

Norm Module

The **norm module** defines some of the concepts that play a central role in LKIF: *norm*, *obligation*, *prohibition*, *permission*, *allowed*, *disallowed*, *obliged*, and *violation*.

The norm is a statement combining two performative meanings: it is *deontic*, in the sense that it is a qualification of the (moral or legal) acceptability of some thing, and it is *directive* in the sense that it commits the speaker to bringing about that the addressee brings about the more acceptable thing (cf. [Nuyts et al., 2005]), presumably through a sanction. These meanings do not have to occur together. It

[4] The legitimate expectation is a "right" in a colloquial sense, correlative to a duty, not to be confused with an exercisable right, that imposes a correlative duty. A violation of the duty to yield is not actionable for the 'victim' of the violation, but damage from a resulting car crash is.

[5] Or they are are contained by the mind of an agent, but this intermediary container does not appear to be very relevant for legal reasoning.

is perfectly possible to attach a moral qualification to something without directing anyone, and it is equally possible to issue a directive based on another reason than a moral or legal qualification (e.g. a warning).

We can reify the violation, the sanction, and/or the duty. As described in [Boer, 2006a], both the legislator and the knowledge engineer often leave one or more of these interpretations implicit.

In discourse we encounter the use of normative notions like duty (obligation) both in a generic sense and in a context relativized to specific objects. Consider the following two examples:

1. If (seller) offers to sell some good to (buyer), and (buyer) agrees to buy the good then (seller) has a duty to sell the good.

2. I have a duty towards John, to sell him the painting I offered him for sale.

The first one is a universal statement on the terminological level. The second one is an instance of it. The problem is of course to reify the norm itself, separate from the expressions that describe what inferences can be made from the existence of the norm.

The usual presentations of deontic logic, whether axiomatic or semantic, treat norms as if they could bear truth-values. A fundamental problem of deontic logics, logics of norms, is to reconstruct it in accord with the philosophical position that norms prescribe rather than describe, and are neither true nor false. In other words, norms do not have truth values. This observation is generally attributed to mathematician and philosopher Walter Dubislav (viz. [Dubislav, 1937]).

This criticism may be waved away by noting that that concepts, triangles, and sets do not have truth values either. Only the statements *about* them do, by affirming or violating the ontological commitments we made regarding these objects.

One may for instance say that such and such a norm is (or is not) part of (or implied by, consistent with, etc.) such and such a normative code. For example one may say that the British driving code requires vehicles, in normal circumstances, to be driven on the left hand side of roads, while continental European codes requires them to drive on the right side. These statements report on the presence or absence of a norm in a given normative system, and are often taken to be the fundamental representation category in deontic logics. To mark the difference, they are sometimes called “normative propositions” or “normative statements”.

As suggested for example by Bengt Hansson in [Hansson, 1969], we need only read a formula $O\alpha$ or $P\alpha$ as saying that α is obligatory, or permitted, according to some fixed system N of norms. On the semantic level, it is enough to take one’s preferred possible-worlds semantics and reinterpret its deontic components as relativised to a given normative system N.

One may also state that such and such situation or behaviour is a violation of a norm, or that this norm is in conflict with that norm, or that some norm directs us to choose this alternative over that alternative in a certain situation.

What we cannot do is equate some proposition or logical sentence to a norm.

As Makinson notes in [Makinson, 1999] people do accept the theoretical problem, but in practice work goes on as if the distinction had never been heard of:

> In axiomatic presentations of systems of deontic propositional logic, the truth-functional connectives “and”, “or” and most spectacularly “not” are routinely applied to items construed as norms, forming compound norms out of elementary ones. But as [Dubislav, 1937, Jörgenson, 1937] already observed, if norms lack truth-values then it is not clear what could be meant by such compounding. For example, the negation of a declarative statement is understood to be true iff the item negated is false, and false iff the latter is true; but we cannot meaningfully formulate such a rule for norms.

Depending on the kind of logic and the type of problem considered, knowledge engineers choose for different conceptualizations. We have the functional approach of Valente (cf. [Valente and Breuker, 1995]), which distinguishes situations into the qualifications allowed, disallowed, and silent. This is only good for assessment, and even in this context limited because of its omission of context. Another flaw is that it defines the situations on the concept level and the norms on the object level, and then relates them through a relations that is outside boundaries of the description logic he uses.

We have the modal approach, which turns obligation, prohibition, and permission into modal operators and works from the observation that obligation and prohibition are interdefinable, and obligation implies permission. This approach suffers from the lack of distinction between norms as objects and normative statements as logical expressions without object identity. There are also several rule-based approaches that try to capture norms in rules with notions like violation or duty as antecedent or conclusion. The rule itself captures the meaning of the norm, so that the confusion between norm and normative statementis again retained.

A last option is to consider the norm as a preference statement (as in [Hansson, 1969, Boer et al., 2005]), again failing to distinguish the statement and the thing one is making the statement about.

Putting them together results in the structure described in figure 4.3. A norm applies to (or *qualifies*) a certain case, allows a certain case - the *obliged* case or *allowed* case - and disallows a certain case - the prohibited or *disallowed* case. The obliged and prohibited case are both subsumed by the case to which the norm applies. Besides that they by definition form a complete partition of the case to which the norm applies, i.e. all cases to which the norm applies are either a mandated case or a prohibited case. This is true of the obligation and the prohibition: they are simply two different ways to put the same thing into words.

Note that we can derive the nature of the comparative relation between mandated and prohibited case from the norm, or postulate the existence of the norm from the comparative relation. The permission, shown in figure 4.4, is different. The permission allows something, but it doesnt prohibit anything. The logical complement of the mandated case is here an opposite qualified case, about which we know only that it cannot be obliged.

Let β be a situation, and α an alternative in a menu[6]. The deontic operators

[6] Beware of interpreting α as an action: the alternatives may concern both descriptions of actions and situations, as long as situations can be conceived of as productive characterizations in the sense

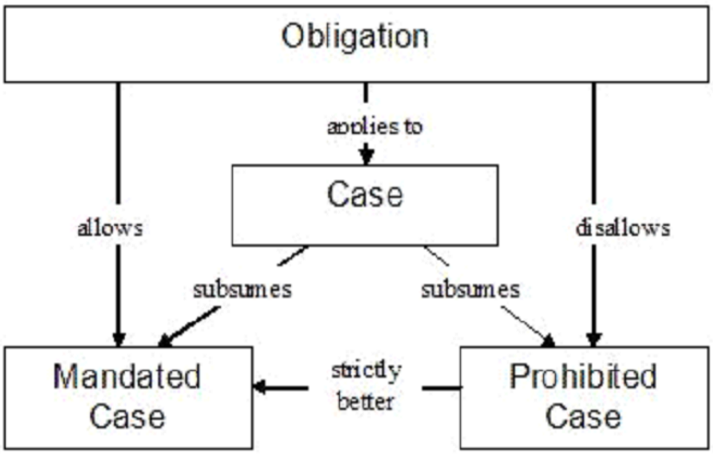

Figure 4.3: An entity-relationship diagram describing the salient structure of obligations and prohibitions.

are reduced to comparative statements as follows:

$O(\alpha \mid \beta)$: $\boxed{\beta, \alpha} \succ \boxed{\beta, \neg\alpha}$

$F(\alpha \mid \beta)$: $\boxed{\beta, \neg\alpha} \succ \boxed{\beta, \alpha}$

$P(\alpha \mid \beta)$: $\boxed{\beta, \alpha} \succeq \boxed{\beta, \neg\alpha}$

These can form the basis of preference-based reasoning system that can meet at least the following desirable characteristics of a deontic knowledge representation:

Observation 1 *Deontic choice $O(\alpha \mid \beta)$: if an agent has the choice between $(\alpha \wedge \beta)$ and $(\neg\alpha \wedge \beta)$ then the agent should choose $(\alpha \wedge \beta)$.*

Observation 2 *What is obligatory is permitted: $O(\alpha \mid \beta) \rightarrow P(\alpha \mid \beta)$*

Observation 3 *The impossible and the meaningless are not obligatory: $\neg O(\alpha \mid \alpha)$ and $\neg O(\neg\alpha \mid \alpha)$ are axioms.*

Observation 4 *There are no conflicting obligations. The obligations $O(\alpha \mid \beta)$ and $O(\neg\alpha \mid \beta)$ are inconsistent: $\neg(O(\alpha \mid \beta) \wedge O(\neg\alpha \mid \beta))$ is an axiom. Idem for $O(\alpha \mid \beta)$ and $P(\neg\alpha \mid \beta)$.*

Observation 5 . *If $\mid \phi \mid$ is the set of worlds w such that $M, w \models \phi$, then the sentences $O(\alpha \mid \top)$, $O(\beta \mid \alpha)$, $O(\neg\beta \mid \neg\alpha)$ are only satisfied by the ordering $\mid \neg\alpha \wedge \beta \mid \prec \mid \neg\alpha \wedge \neg\beta \mid \prec \mid \alpha \wedge \neg\beta \mid \prec \mid \alpha \wedge \beta \mid$.*

of [Sartor, 2006]. Key is that the situation can be brought about by an adressee.

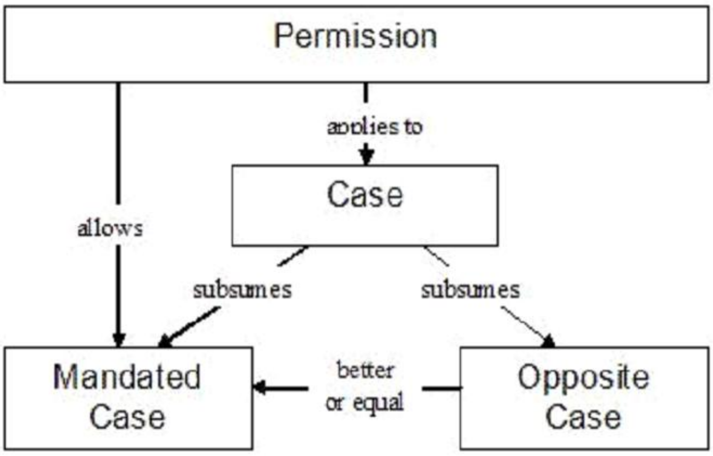

Figure 4.4: An entity-relationship diagram describing the salient structure of a permission. Obligation and prohibition are subsumed by permission.

A similar characterization can be given that depends on a functional mapping from propositions to the set $\{allowed, disallowed, silent\}$ as in [Valente and Breuker, 1995]:

$$O(\alpha \mid \beta) \; : \; f(\{\beta, \alpha\}) = allowed, f(\{\beta, \neg\alpha\}) = disallowed$$

$$F(\alpha \mid \beta) \; : \; f(\{\beta, \neg\alpha\}) = allowed, f(\{\beta, \alpha\}) = disallowed$$

$$P(\alpha \mid \beta) \; : \; f(\{\beta, \alpha\}) = allowed, f(\{\beta, \neg\alpha\}) = silent$$

This representation (without using the comparatives) is less powerful for a number of reasons. For one, it cannot be used to detect contrary-to-duty obligations.

One of the attractive features of the representation in the form of a comparative relation is that it produces triangles between a 'context' β, and two good ($\alpha \wedge \beta$) and bad ($\alpha \wedge \beta$) alternatives that are complete partitions of the context. For any norm, *Normatively_Qualified* $\equiv$ *Disallowed* $\sqcup$ *Allowed.* It naturally fits in a graphical representation of taxonomies, and knowledge acquisition methods like the repertory grid[7]. The disadvantage of this representation is that it splits up the meaning of the norm in several different definitions (the things described by the *Normatively_Qualified*, *Disallowed*, *Allowed* attached to the norm), and that the constraint *Normatively_Qualified* $\equiv$ *Disallowed* $\sqcup$ *Allowed* cannot be enforced in OWL DL. The user should also note that the norm is an individual: a concept with one instance in any given context.

[7] It actually fits the repertory grid better than the generic taxonomies this method is often used for. One of the persistent mistakes of students who first learn to make taxonomies is that they naturally assume that subtypes are disjoint, which is not normally the case.

4.2 Integrating other upper/core ontologies

Earlier experience, as in e.g. [Breuker and Hoekstra, 2004b, Breuker and Hoekstra, 2004a], suggests a commonsense basis for distinguishing main categories in an ontology for law. The intentional nature of the core concepts for the LKIF ontology (see e.g. sections 4.1.3,4.1.3) emphasises the distinction with other more (meta)physically inclined top ontologies such as SUMO[8], Sowa's upper ontology [Sowa, 2000] and DOLCE[9] [Gangemi et al., 2002]), but shows similarities with the distinction between *intentional*, *design* and *physical* stances described in [Dennett, 1987]. As some of these top- or upper ontologies (SUMO, Sowa) have not a common -sense basis – e.g. mental and social entities are poorly represented – they could neither be used as a top for *LKIF-Core*, nor as a source of descriptions of legal terms. The upper part of the CYC [10] ontology and DOLCE [Gangemi et al., 2003, Massolo et al., 2002] are claimed to have a common-sense view, but this common-sense view is rather arbitrarily based upon personal intuition than on empirical evidence. LRI-Core on the other hand is to a large extent based upon empirical studies in cognitive science.

We also collected terms from existing core or upper ontologies that contained legal terms. In this way we could re-use even their formal (OWL-DL) definitions. For an overview of legal ontologies, see [casanovas et al., 2006]. However, it turned out that the amount of re-use and inspiration was rather limited. The following core ontologies for law where consulted: both for their potential contribution for creating a coherent top for *LKIF-Core*, and specifically legal terms already modelled.

Language for Legal Discourse, LLD [McCarty, 1989] is a first attempt to define legal concepts in the context of legal reasoning, using formulae and rules. Properly speaking, LLD is not an ontology but a framework (it was conceived before "ontologies were invented") but it is a relatively rich source for legal terms and their definitions. We have included concepts from LLD in *LKIF-Core*.

LRI-Core [Breuker et al., 2004b, Breuker and Hoekstra, 2004a] is intended as a core ontology for law, and provide the top-structure for an ontology of Dutch criminal law (CRIME.NL), but the number of typical legal concepts in this legal core ontology was disappointingly small. However, its top structure appeared to be valuable in constructing *LKIF* as is further described in Section 4.2.1.

DOLCE-CLO (Core Legal Ontology) Core Legal Ontology (CLO) is a core ontology used to support the construction of legal domain ontologies [Gangemi et al., 2005]. CLO organises juridical concepts and relations on the basis of formal properties defined in DOLCE+. Although purpose and layers are largely similar to those of LRI-Core, the top structures differ considerably, as can be seen by comparing Figure 4.7 (LRI-Core) on page 41, and Figure 4.5.

[8] Suggested Upper Merged Ontology; `http://ontology.teknowledge.com`

[9] Descriptive Ontology for Linguistic and Cognitive Engineering; `http://www.loa-cnr.it/DOLCE.html`

[10] www.cyc.com

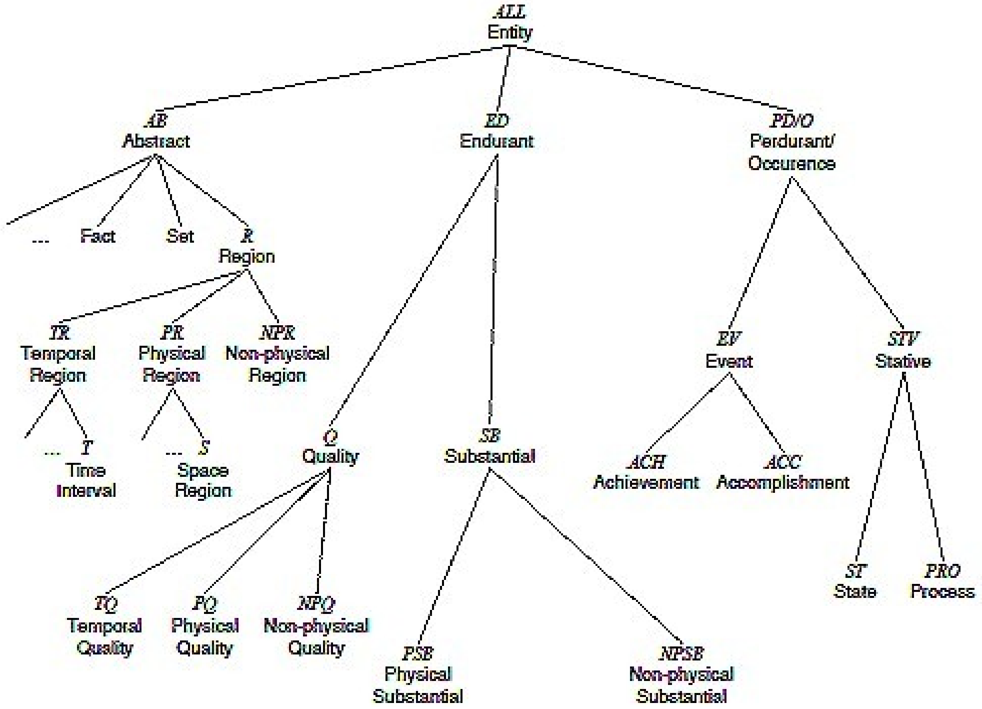

Figure 4.5: Top classes of of DOLCE [Gangemi et al., 2005]

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) is a top-level foundational ontology developed originally in the EU WonderWeb Project[11]. DOLCE+ is an extension of DOLCE containing the distinction between descriptions and situations (also called D&S).

As shown in Figure 4.6, a `description` is a `social object`, which represents a conceptualisation. Examples of descriptions are regulations and laws.

Social objects are similar to physical objects but differently from them, social objects are dependent on some agentive physical object that is able to conceive them. A `concept` is also a social object, which is defined by a description. `Role`, `course` and `parameter` are the primary concepts reified in D&S which respectively classify the three different entities in DOLCE: `endurant`, `perdurant` and `region`. For each of these concepts a sub-relation of the classifies relation exists: `played by`, `sequences` and `valued by`. A role can be, for example, president of republic. Examples of courses are routes, tasks which can have as requirement speed limits or temporal constraints.

`Figures` are other social objects defined by descriptions, but differently from concepts, they do not classify entities. Examples of figures are organisations or political geographic objects.

Endurant and Perdurant are linked by the relation of *participation*. Endurants

[11] `http://wonderweb.semanticweb.org`.

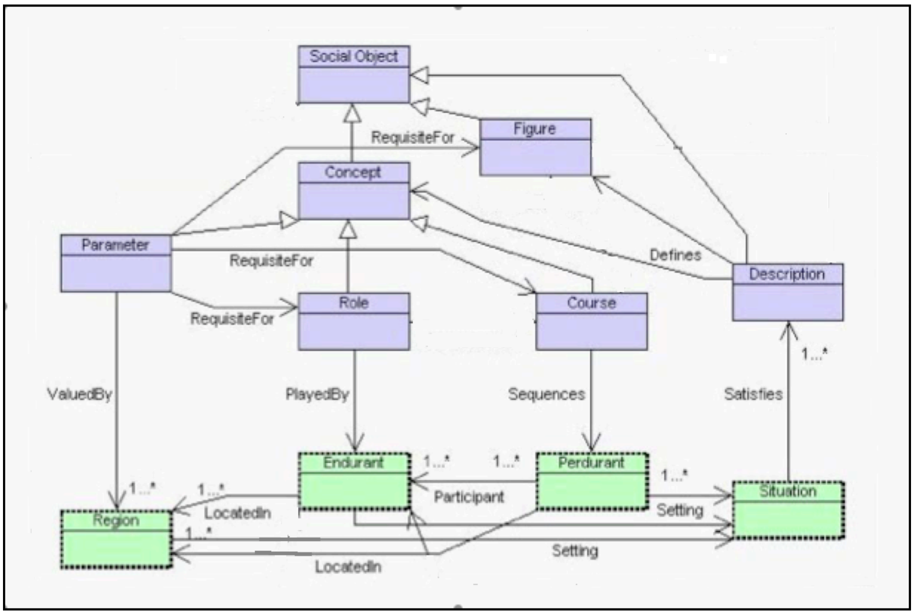

Figure 4.6: D&S design patterns as UML diagrams

are *localised* in space, and get their temporal location from the perdurants they participate in. Perdurants are localised in time, and get their spatial location from the endurants participating in them. Parameters and roles, figures, or courses are related by a `requisite for` relation, expressing the kind of requisites entities that are classified by roles or courses should have.

A `situation` is an entity that appears in the domain of an ontology when there is a description whose components can carve up a view on that domain. For example, legal cases are situations. Between situations and entities holds the `setting` relation.

4.2.1 Principles and main categories inherited from LRI-Core

In the top-down approach we have imported and adapted *LRI-Core*. We have used the top classes of this upper ontology [12] for *LKIF-Core*. These top classes can be described as follows.

The top of *LRI-Core* consists of five major categories: each referring to a 'world'. These five are: physical and mental concepts, roles, abstract concepts and terms for occurrences. [13] These categories follow from an assumed evolution of human (and animal) conceptualisations of reality.

Figure 4.7 presents the top two layers of *LRI-Core*

Occurrences An ontology should not be structured according to the *way* things occur in physical, mental, or fantasy worlds, but rather to *what* the things 'essen-

[12] LRI-Core was intended to be a legal core ontology. However, the 'legal' layer is largely underspecified, so that we will use here mainly its 'upper' layer.

[13] Most likely we have to add a sixth category: life, or organism.

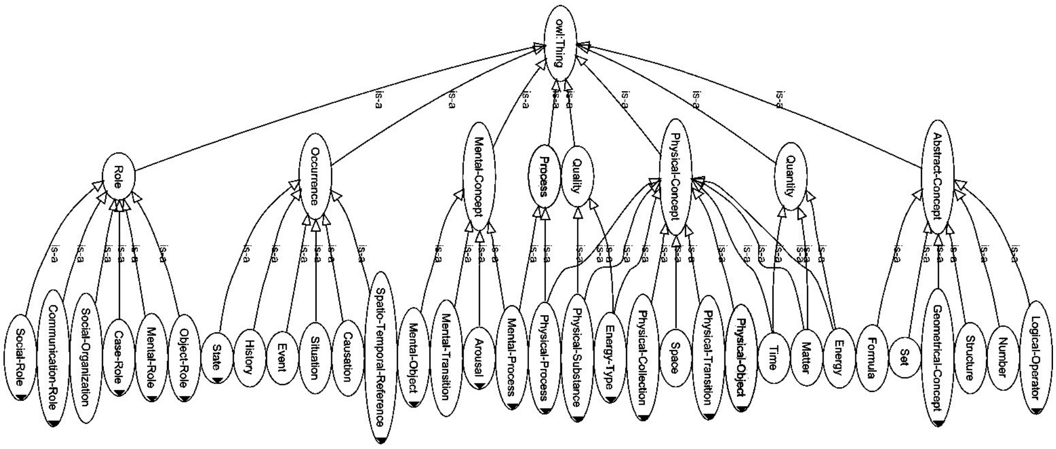

Figure 4.7: LRI-Core, top two layers

tially' are. Ontology has a Platonic flavour in the sense that it specifies the ideas with which we understand the world as it passes by.

Ontologies define and deliver the building blocks for the construction or interpretations of actual situations and histories: partial models of real or imaginary worlds. Histories describe the life line of individual entities, and situations are diachronic spatial structures of objects and processes. Note, however, that in *LKIF-Core* situation is not viewed as diachronic, but includes histories (and even future states), i.e. situation is equivalent to an instantiated world.

The distinction between a concept and its occurrence (instance) [14] is particularly relevant for mental concepts like action, plan, norm and role and their execution (respectively, their observation) may show divergence that can be marked as 'bad' or even illegal. A divergence can only be identified if a mental plan, norm or role still exists so that it can be compared with actual behaviour (or its memory or recording).

The category of occurrences in *LRI-Core* captures those strictly temporal aspects related to the *execution* of scenarios involving objects and processes. This means that events are occurrences, but processes are not. Where processes contain

[14] In OWL there is no distinction between individual and instance; the terms are used indiscriminately. For some reason, the distinction is lost in knowledge representation research [Brachman, 1979]. Conceptually, individuals have an identity and a life-cycle, while instances are occurrences of individuals at a particular time and a particular place, i.e. situation dependent.

the explanation of the changes they cause, events only describe a discrete difference between the situation before and after the event took place: they describe the input-output of the execution of a process, and happen 'in' time. All this does not reduce the need for terms to talk about occurrences in general. For instance, above we have used terms like situation, event, history and entity. These terms refer to occurrences in an abstract sense that can legitimately be part of an ontology that defines concepts. Therefore, *LRI-Core* has a category of 'occurrences'.

Physical Entities The physical world evolves around two main classes: *physical objects* and *processes*. Objects are bits of matter, which in turn is typed by its substance. Objects have mass, extension, viz. form and aggregation state (limiting form). The existence of objects expresses the notion that matter (in particular solid matter) is what renders the physical world relatively stable and observable. Physical situations are usually described by the arrangement of individuals of physical objects.

This intuition does not exist for the second major class that governs the physical world: process. Processes consume energy to change objects, or parts of objects. Though energy is a naively problematic concept [Hayes, 1985], its use has become widespread to such an extent, that it has conquered its place in common sense. Processes are described by the changes they bring about. Change is an inherently temporal concept, belonging to the realm of occurrences. Through interaction, processes can cause one another, leading to series of events that only stop at some equilibrium: in general conceived as that there are no interactions at all. In *LRI-Core*, processes are typed according to two views: (1) formal change (transformation, transduction and transfer) and (2) the kinds of (properties of) objects involved. (e.g. movements are the change of position of objects; chemical processes change the substance of objects, etc.). A third property is whether a process produces or consumes energy; the default is the latter.

The concept of process is often used as synonymous to action and activity. LRI-Core defines actions as processes that are initiated by an agent acting as actor. Notwithstanding the intricacies of mental (or agent) causation [Kim, 1998], the action itself is strictly physical: i.e. some muscle movement. The mental perspective implied by agent-causation is that actions are intended: they are preceded by some kind of intentional decision to act.

Mental Entities Conceptions of the mental world have a strong analogy to the physical world. We conceive the *mind* as containing (mental) objects, like concepts and memories, which are processed by mental processes that transform or transfer these objects. Memories are retrieved; concepts are formed. Moreover, these mental objects may be aggregations of more elementary objects. Memories consists of multimedia representations of situations experienced; thoughts – '*propositions*' – are made of more elementary parts like concepts.

The contents (substance) of these objects are representations. The conceptual content of thoughts is intended by propositional attitudes, like belief, desire, norm etc. Mental objects are processed or stored in containers (such as the mind) which in turn can have parts, e.g. memories. Mental processes like thinking, memorising, imaging are operations on mental objects. The equivalent of physical energy in mental processing is the concept of emotion: the forces that make us focus our mental energies.

There is however, an important difference between the mental and the physical world. Where physical processes are governed by causation, mental processes are controlled by intention. For processes governed by intention (free will) we rather use the term 'action'. Thinking is an action, as we assume that we have full control over our thoughts and can decide about what we are thinking. However, where our mind escapes our conscious intentions, as e.g. in getting in uncontrollable emotional states, or in forgetting an appointment, we rather take a physical than an intentional stance. Despite this subtle difference, we keep the term mental process to cover both, as we want to reserve the term action for agent-caused processes. Note that actions may affect the mental or the physical world. A special type of actions are speech acts: actions that have physical effects (speaking/writing) but are intended to affect the mind of a hearer/reader. The role of mental conceptualisations is extremely important in understanding and communicating with other people. Their primary use lies in their role as building-stones of models of the minds of other people: user-models. The intentional stance means that we attribute intentions, and intention directed mental processing and belief to other people, and to some extent to most animals (or even computers, [Dennett, 1987]).

Roles Roles cover functional views on physical objects (devices), on agent behaviour or on mental processes. In particular, social behaviour and social organisations are explained as (consisting of) roles. Typical mental roles are epistemological ones. For instance conclusion, evidence and hypothesis are roles in problem solving processes and can therefore also be categorised under mental classes [Breuker, 1994]. From a role perspective, functions are roles of physical objects, e.g. we may use objects for non-intended functions.

Roles are entities in the mind, they do not 'really' exist. Roles are idealisations: models. We may not play a role correctly. An important distinction should be made between playing a role and the role itself: "agents can act, and roles cannot" [Pacheco and Carmo, 2003]. Correcting incorrect role playing does not mean that we change the role: we change our behaviour. Like plans and processes, roles in ontologies are often confounded with their execution, in the same way as the execution of a symphony may be confounded with the symphony itself. The original meaning of the term role refers to a role of paper that contained the text of an actor in a play. Also the role-taker (some agent) and the role are often confounded, which may become obvious when we identify a role with a person. These kinds of confusions have made conceptual modellers aware of the tricky issues about roles [Steimann, 2000].

Roles are often viewed as relationships [Sowa, 2000, Steimann, 2000, Masolo et al., 2004]. Indeed, social roles have mutuality and complementarity. No students without teachers; no parents without children; no speakers without hearers, etc. In theory of law, a related view exists about the mutuality of legal positions: i.e. rights and duties [Hohfeld, 1919, Kelsen, 1991]. For instance, if citizens have the obligation to vote, the government has the duty to enable this voting. Nevertheless, this complementarity of roles might not be of enough importance to grant their representation as relationships in an ontology. The ontology may specify such relationships, but the primary notion of role is as a concept [Brachman et al., 1991].

This becomes clear when we look at roles as concepts, Roles are behavioural requirements on role execution and on qualifications of role taking. These requirements are prescriptions, i.e. they are normative. In modern society many roles have formal requirements enforced by *law*. Legislation addresses actors by the roles they play. [15] If actual behaviour deviates from the norms attached to these roles we violate the law. Violations are based upon the distinction between the prescription (role) and role performance. Therefore, in court, it is the actor of the role who is made responsible: as a person; not as a role. Even the fictitious concept of legal-person for social organisations turns into concrete responsibilities of the liable persons who have mis-performed their roles.

Abstract Entities As all concepts are abstractions, one may argue that a separate abstract world is difficult to see. However, common-sense knows about a (small) number of proto-mathematical concepts, such as collections, sequences and count-numbers (positive integers). We know about geometric simplifications such as line, circle, square, cube, etc. [Lakoff and Núñez, 2000] even argue that these common sense notions are the real roots of our mathematics. Nonetheless, these kind of semi-formal abstractions do not play a very central role in law, and therefore *LRI-Core* is thinly populated with abstract classes.

4.3 Evaluation

The purpose of the study outlined in Section 4.1.1 is more ambitious than only the selection of the most basic terms for describing law, but time and effort constraints make it that we can only report here the results for selection. The list of terms will be/are submitted to 1) a group of legal professionals (taking courses in legal drafting), and 2) law students. These empirical studies are planned in the sideline of Estrella. By applying statistical cluster analysis, we may be able to identify properties of the scales used (e.g. are they independent?) and whether the statistical clusters have some resemblance to the clusters we have identified in the modelling a number of central, basic legal concepts.

The results of this analysis will be used to evaluate the *LKIF-Core* ontology compared to the requirements we identified in the previous chapters. Furthermore, the *LKIF-Core* ontology vocabulary will be adopted for expressing the LKIF vendor models that will be developed within ESTRELLA.

[15] An exception to this rule is in criminal law.

Chapter 5

Modules

5.1 Modules and Dependencies

The LKIF core ontology consists of 15 modules, each of which describes a set of closely related concepts from both legal and commonsense domains. In that sense, the LKIF core ontology is rather a library of ontologies than a monolithic body of definitions. A glossary of the concepts and properties included in these modules can be found in Section 5.2. Appendix A.3 provides the definition of each concept and property in the Manchester OWL syntax[1]. Section 5.3 describes a candidate framework for argumentation, which will serve as an extension to the LKIF core ontology.

Type	Local	Imported	Inferred	Total
owl:AnnotationProperty	0	2	0	2
owl:Class	0	205	0	205
owl:DatatypeProperty	0	1	0	1
owl:FunctionalProperty	0	2	0	2
owl:InverseFunctionalProperty	0	2	0	2
owl:ObjectProperty	0	98	0	98
owl:Ontology	1	14	0	15
owl:SymmetricProperty	0	7	0	7
owl:TransitiveProperty	0	12	0	12

Table 5.1: Statistics for the core ontology module.

The most abstract concepts are defined in five closely related modules: top, place, mereology, time and spacetime.

top The LKIF top ontology is largely based on the top-level of LRI-Core but has less ontological commitment in the sense that it imposes less restrictions on subclasses of the top categories.

place The place module partially implements the theory of relative places ([Donnelly, 2005]) in OWL DL.

[1] See `http://www.co-ode.org/resources/reference/manchester_syntax/`

mereology The mereology module defines mereological concepts such as parts and wholes, and typical mereological relations such as part of, component of, containment, membership etc.

time The time module provides an OWL DL implementation of the theory of time by [Allen, 1984].

spacetime The space-time module is a placeholder for the place and time modules.

Basic-level concepts are distributed across four modules: process, role, action and expression.

process The process module extends the LKIF top ontology module with a definition of changes, processes (being causal changes) and physical objects. It introduces a limited set of properties for describing participant roles of processes.

role The role module defines a typology of roles (epistemic roles, functions, person roles, organisation roles) and the plays-property for relating a role filler to a role.

action The action module describes the vocabulary for representing actions in general. Actions are processes which are performed by some agent (the actor of the action). This module does not commit itself to a particular theory on thematic roles.

expression The expression module describes a vocabulary for describing, propositions and propositional attitudes (belief, intention), qualifications, statements and media. It furthermore extends the role module with a number or epistemic roles, and is the basis for the definition of norms.

These basic clustered are extended by three modules to cover legal concepts: legal action, legal role and norm.

legal-action The legal action module extends the action module with a number of legal concepts related to action and agent, such as public acts, public bodies, legal person, natural person etc.

legal-role The legal role module extends the role module with a small number of legal concepts related to roles, legal professions etc.

norm The norm module is an extension primarily on the expression module where norms are defined as qualifications. Please refer to [Boer et al., 2007] for a more in-depth description of the underlying theory. It furthermore defines a number of legal sources, e.g. legal documents, customary law etc., and a typology of rights and powers, cf. [Sartor, 2006, Rubino et al., 2006]

In addition to these legal clusters, two modules are provided that cover the basic vocabulary of two frameworks: modification and rules.

modification The modification module is both an extension of the time module and the legal action module. The time module is extended with numerous intervals and moments describing the efficacy and being in force of legal documents. The action module is extended with a typology of modifications. These concepts are described in further detail in [Palmirani et al., 2007].

rules The rules & argumentation module defines roles central to argumentation, and describes the vocabulary for LKIF rules as defined in [Boer et al., 2007], chapter 5. The module leaves room for further extension to complex argumentation frameworks (AIF, Carneades).

Finally, these fourteen modules are integrated in the *LKIF-Core* ontology module. This module does not provide any additional definitions, but functions as an entry-point for users of the ontology library. Figure 5.1 gives an overview of the dependencies between the different modules. Table 5.1 shows statistics for the core ontology module.

5.2 Glossary of Classes and Properties

This section lists all classes and properties defined in the LKIF core ontology, sorted alphabetically and grouped by module. Every module, class and property is described shortly. For a full description, graphical representation and formal definition of each term, we refer to the online documentation available at `http://www.estrellaproject.org/lkif-core/doc`. The ontology itself can be downloaded from `http://www.estrellaproject.org/lkif-core`.

5.2.1 Action Module

Direct Imports
expression module, see Section 5.2.2 *process* module, see Section 5.2.9
Indirect Imports
role module, see Section 5.2.10 *top* module, see Section 5.2.14 *time* module, see Section 5.2.13 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *space-time* module, see Section 5.2.12
Short description
The action module describes the vocabulary for representing actions in general. Actions are processes which are performed by some agent (the actor of the action). This module does not commit itself to a particular theory on thematic roles.

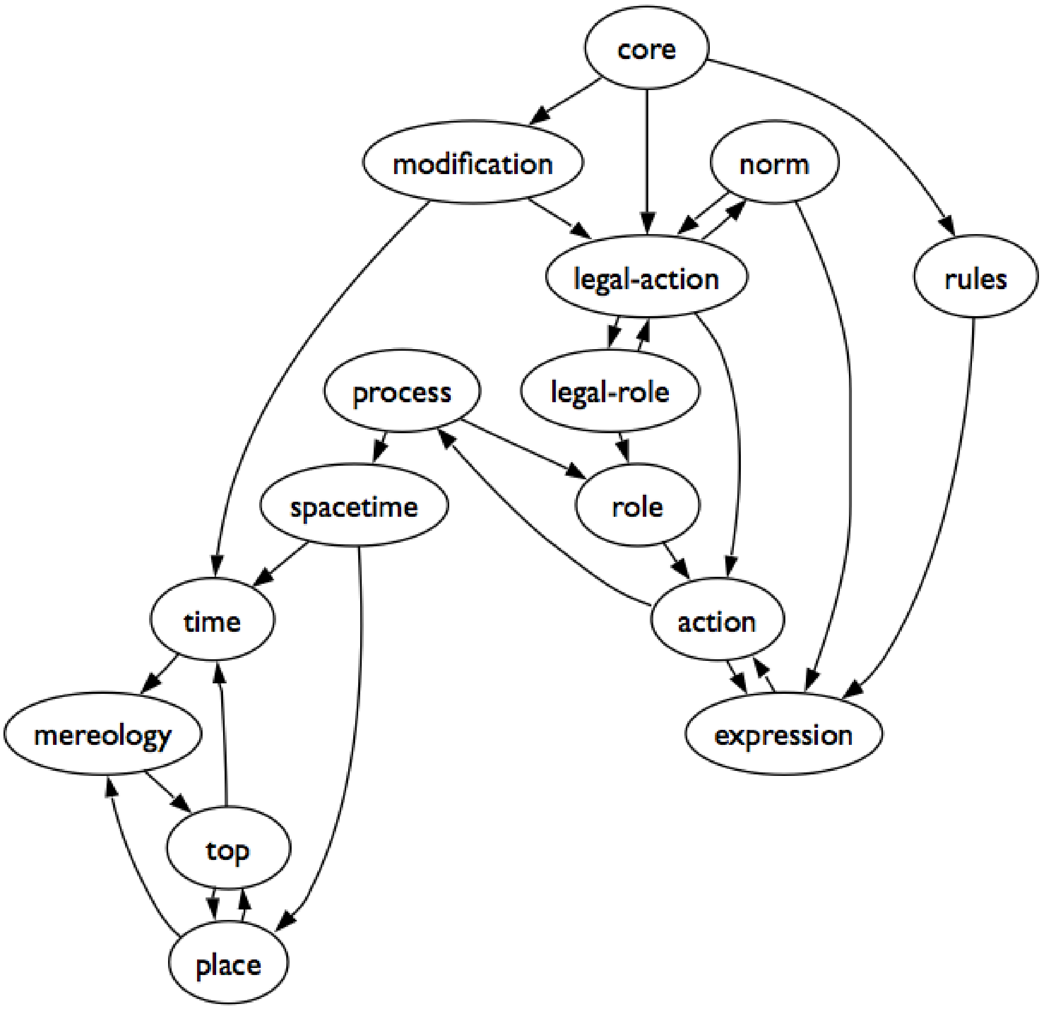

Figure 5.1: Dependencies between LKIF-Core modules.

Class action:Action An action is a change which is brought about by a single agent, playing a role. The agent is the holder of some indended outcome of the action: an action is always intentional. The intention of the agent has usually corresponds with an expectation the intended outcome to be brought about: the agent believes in some expectation. Note that the intention might not correspond with the *actual* outcome of the action.

ObjectProperty action:actor Specifies that some participant is an actor in the action.

ObjectProperty action:actor_in Specifies that the participant is an actor in some action.

Class action:Agent An agent is any owl:Thing which can act, i.e. play the ‘actor’ role wrt. an action

Class action:Artifact A physical object created by some person to fulfil a particular purpose

Class action:Collaborative_Plan NOTE: this definition cannot currently be expressed in OWL

Class action:Continuation The continuation of an action, once initiated

Class action:Creation An act which results in the creation of some entity/individual

Class action:Initiation The initation of an action

ObjectProperty action:intended_by Specifies that some intention is held by some agent

ObjectProperty action:intends Specifies that the agent holds some intention

Class action:Natural_Object A natural object is an object not created by man.

Class action:Organisation An organisation is a group of other organisations or persons which acts ‘as one’. An organisation can be both formal (i.e. created by law or decree) or informal.

Class action:Person A person is an individual agent. Usually associated with ‘human being’.

Class action:Personal_Plan A personal plan is a plan which is held (and can be executed) by at most one agent.

Class action:Plan A plan is a structure of multiple other plans or actions. These can be both sequential or concurrent. Usually a plan is referred to in the context of the intention to act of some agent, however when executed the plan itself comes into effect.

Class action:Reaction A reaction is an action that is (at least) performed by an agent that has observed some other action (this is an intensional view): Action ¡-¿ Reaction

Class action:Speech_Act A speech act (or illocutionary act) creates some propositional attitude which qualifies an expression (which by default is mediated through some medium). The actor of the speech act is the utterer of the atitude (NB cannot be expressed in OWL DL).

Class action:Termination The termination of an action.

Class action:Trade A trade is an exchange of some things between two agents a and b. Consists of two actions A and B where a is the actor of action A and the recipient of action B, and vice versa. Usually these actions are performed concurrently, but these may also be done consequently. The ownership of the things being traded is the requirement/result pair for each action.

Class action:Transaction A transaction is an exchange or interaction between at least two agents a and b. Consists of two actions A and B where a is the actor of action A and the recipient of action B, and vice versa. Usually these actions are performed concurrently, but these may also be done consequently.

5.2.2 Expression Module

Direct Imports
action module, see Section 5.2.1 *top* module, see Section 5.2.14
Indirect Imports
role module, see Section 5.2.10 *time* module, see Section 5.2.13 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *space-time* module, see Section 5.2.12
Short description
The expression module describes a vocabulary for describing, propositions and propositional attitudes (belief, intention), qualifications, statements and media. It furthermore extends the role module with a number or epistemic roles, and is the basis for the definition of norms

ObjectProperty expression:addressee Allows for expressing the relation between a communicated attitude and the Agent to which the communication act is addressed

Class expression:Argument An argument is a reason that is expressed through some medium

ObjectProperty expression:asserted_by Relates an expression being asserted to the assertion

Class expression:Assertion The assertion is subject to a fit between words and world, in Searle's terms. It's propositional content can be true of false. If it is inconsistent with other assertions, beliefs, observations, it may come to be considered false.

ObjectProperty expression:asserts Relates an assertion to the expression being asserted

Class expression:Assumption proposes something that usually is the case, although there is no specific evidence that it is true in this particular case

ObjectProperty expression:attitude Relates a proposition to the attitude held towards it

ObjectProperty expression:author Relates an expression to the author of the expression (e.g. for expressions contained in documents)

ObjectProperty expression:bears A Medium 'bears' or carries expressions.

Class expression:Belief Someting an agent 'believes in', i.e. holds as true

ObjectProperty expression:believed_by Relates a belief to the agent holding the belief

ObjectProperty expression:believes Relates an agent to the belief(s) it holds

Class expression:Cause A cause is an epistemic role played by something which is the outcome of a (chain) of processes

Class expression:Communicated_Attitude A communictated attitude is a propositional attitude involved in an act of communication.

Class expression:Declaration Searle: the successful performance of a declaration is sufficient to bring about the fit between words and world, to make the propositional content true. In other words, if there is an inconsistency between the declaration and assertions, beliefs, or observations, it is not the declaration that is false. True of definitions and norms, and several other performative statements by legislators.

ObjectProperty expression:declared_by Relates a declared expression to the attitude to the declaration

ObjectProperty expression:declares Relates a declaration to the expression being declared

Class expression:Desire A feeling of wanting

Class expression:Document A Document bears some (and only) expression(s) stated by some statement in writing.

Class expression:Evaluative_Attitude An evaluative attitude expresses e.g. a judgement. The proposition qualified by the evaluative attitude is comparable to something else.

Class expression:Evaluative_Proposition Some thing which is evaluatively qualified, i.e. an evaluation applies to the proposition, the proposition is judged. The proposition is comparable to some other proposition.

ObjectProperty expression:evaluatively_comparable Expresses whether some thing is evaluatively comparable to some other thing.

Class expression:Evidence Observation and/or statement, used as a backing for a belief. Evidence is the role of observation which is qualified by some belief.

Class expression:Exception An exception is something that is excluded from a general statement or does not follow a rule. In LKIF rules, an exception is a rule which has an exception relation to another rule (cf. [Boer et al., 2007])

Class expression:Expectation An expectation is a predictive belief held on purely logical grounds, i.e. not based on direct external evidence, like an observation or a statement by another agent. It is also often characterised as a "belief about the future", but this isn't very helpful since one may also hold expectations about the (yet unknown) past. The theory of evolution is for instance a fertile ground for predictions about the existence of past species, and therefore at the same a predictor of future observations.

Class expression:Expression An expression is a proposition borne by some medium, e.g. a document.

Class expression:Fact A fact is a proposition (about something) which is (commonly) agreed upon to hold as true: it has some backing in evidence

ObjectProperty expression:held_by Relates a propositional attitude to the agent holding the attitude

ObjectProperty expression:holds Relates an agent to the propositional attitude it holds

Class expression:Intention Intention is where the agent expects certain consequences of his or her actions and desires those consequences to occur. This concept must be related to qualification and preference in some way.

Class expression:Lie An assertion that is inconsistent with the beliefs of the speaker. It is *intentionally* false.

Class expression:Medium A medium is a bearer of expressions, i.e. externalised propositions. Propositions become expressions once they are externalised through some medium.

ObjectProperty expression:medium Relates an expression to the medium it is borne in (i.e. for extentional propositions)

Class expression:Observation An observation is the role played by some proposition believed to be true or false through observation of an agent.

ObjectProperty expression:observer Relates a believed observation to the agent observing it

ObjectProperty expression:observes Relates an agent to the thing it beliefs it observes

Class expression:Problem NOTE: This currently cannot be expressed: OWL DL does not support negation of properties. The definition will be further elaborated once OWL 1.1 is supported by development tools.

Class expression:Promise A promise is a communicated attitude about some future action or state

ObjectProperty expression:promised_by Relates an expression to the promise over the expression

ObjectProperty expression:promises Relates a promise to the expression being promised

Class expression:Proposition NOTE: In previous versions of this ontology, this Class was called 'Qualified'. In lieu of the separation between Proposition and Propositional_Attitude, the intended meaning of that Class has been moved to 'Evaluative_Proposition'

Class expression:Propositional_Attitude A propositional attitude connects a person (the holder of the attitude) to some proposition, in fact it expresses some qualification over the proposition. Distinguishing a proposition from the propositional content expressed by it is necessary when properties relating to the thing expressed and properties of the expression itself must be distinguished. For LKIF the distinction between Belief, Intention, Qualification, and Observation is relevant. The distinction between belief/expectation, intention, and observation is relevant for i.a. establishing mens rea (guilty mind). The distinction between beliefs (expressing the content of the mind of an agent) and statements (expressing the content of an act of communication by an agent) is Class ical.

Class expression:Qualification A qualification expresses e.g. a judgement. The thing qualified by the qualification is comparable to something else.

Class expression:Qualified Something that is qualified by some qualification

ObjectProperty expression:qualified_by Relates something which is qualified to the attitude or qualification qualifying it

ObjectProperty expression:qualifies Relates an evaluative attitude or qualification to the proposition or thing being qualified

Class expression:Reason Teleological counterpart of cause

ObjectProperty expression:stated_by Relates a statement to its author

Class expression:Statement_In_Writing Not to be confused with the actual writing/document itself, which is the medium of the statement.

ObjectProperty expression:states Relates an author to its statements

Class expression:Surprise NOTE: This currently cannot be expressed: OWL DL does not support negation of properties. The definition will be further elaborated once OWL 1.1 is supported by development tools.

ObjectProperty expression:towards A towards relation between a propositional attitude and a proposition expresses that the attitude is held towards that proposition. Qualification of the proposition can be either true or false, i.e. the attitude denotes whether the proposition is either true or false.

ObjectProperty expression:utterer Relates an utterance (communicated propositional attitude) to its utterer

ObjectProperty expression:utters Relates an agent to its utterance(s)

5.2.3 Legal Action Module

Direct Imports
action module, see Section 5.2.1 *norm* module, see Section 5.2.7 *legal role* module, see Section 5.2.4
Indirect Imports
expression module, see Section 5.2.2 *process* module, see Section 5.2.13 *top* module, see Section 5.2.14 *time* module, see Section 5.2.13 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *space-time* module, see Section 5.2.12
Short description
The legal action module extends the action module with a number of legal concepts related to action and agent, such as public acts, public bodies, legal person, natural person etc.

Class legal-action:Act_of_Law Act of law: a public act by a legislative body which creates an expression with legal status; the legal status depends on the jurisdiction of the legislative body.

Class legal-action:Assignment A public act that attributes a power to perform a public act to a public body.

Class legal-action:Association A voluntary association (also sometimes called an unincorporated association, or just an association) is a group of individuals who voluntarily enter into an agreement to form a body (or organization) to accomplish a purpose.

Class legal-action:Co-operative An autonomous association of persons united voluntarily to meet their common economic, social, and cultural needs and aspirations through a jointly-owned and democratically-controlled enterprise.

Class legal-action:Company A company refers to a legal entity formed which has a separate legal identity from its members, and is ordinarily incorporated to undertake commercial business. Although some jurisdictions refer to unincorporated entities as companies, in most jurisdictions the term refers only to incorporated entities.

Class legal-action:Corporation A corporation is a legal entity which, while being composed of natural persons, exists completely separately from them. This separation gives the corporation unique powers which other legal entities lack.

Class legal-action:Decision NOTE: Cannot express that the 'promised' act corresponds to the act qualified by the statement in writing

Class legal-action:Delegation Delegate: entrust a task or responsibility to some other person

Class legal-action:Foundation A kind of (philanthropic) organization, set up as a legal entity either by individuals or institutions, with the purpose of distributing grants to support causes in line with the goals of the foundation

Class legal-action:Incorporated An organisation formed into a legal corporation

Class legal-action:Legal Person (Private Law) A legal person as defined in private law

Class legal-action:Legal_Person A legal entity is a natural person or a legal construct through which the law allows a group of natural persons to act as if it were a single composite individual for certain purposes. The most common purposes are lawsuits, property ownership, and contracts. Sometimes referred to as corporate personhood or legal personality, this concept allows for easy conduct of business by having ownership, lawsuits, and agreements under the name of the legal entity instead of the several names of the people making up the entity.

A legal entity is not necessarily distinct from the natural persons of which it is composed. Most legal entities are simply amalgamations of the persons that make it up for convenience's sake. A legal entity that does have a separate existence from its members is called a company or corporation. This distinction gives the corporation its unique perpetual succession privilege and is usually also the source of the limited liability of corporate members. Some other legal entities also enjoy limited liability of members, but not on account of separate existence (Source: Wikipedia.org)

Class legal-action:Legal_Speech_Act A legal speech act creates some propositional attitude which qualifies a legal expression (which by default is mediated through some medium).

Class legal-action:Legislative Body A legislature is a type of (representative) deliberative assembly with the power to adopt laws.

Class legal-action:Limited_Company A limited company is a corporation whose liability is limited by law

Class legal-action:Mandate Mandate: give (someone) authority to act in a certain way

Class legal-action:Natural_Person A natural person is a human being perceptible through the senses and subject to physical laws, as opposed to an artificial person, i.e., an organization that the law treats for some purposes as if it were a person distinct from its members or owner.

Class legal-action:PLC Similar to the US corporation, offers several advantages over trading as sole trader.

Class legal-action:Public_Act A public act is an act by some Person or Organisation which creates (at least) a communicated attitude (and thereby an expression)

Class legal-action:Public_Body A public body or body created by an act of law to serve a public interest

Class legal-action:Society An organized group of people associated together for religious, benevolent, cultural, scientific, political, patriotic, or other purposes.

Class legal-action:Unincorporated

5.2.4 Legal Role Module

Direct Imports
legal action module, see Section 5.2.3 *role* module, see Section 5.2.10
Indirect Imports
norm module, see Section 5.2.7 *action* module, see Section 5.2.1 *expression* module, see Section 5.2.2 *process* module, see Section 5.2.9 *top* module, see Section 5.2.14 *time* module, see Section 5.2.13 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *space-time* module, see Section 5.2.12
Short description
The legal role module extends the role module with a small number of legal concepts related to roles, legal professions etc.

Class legal-role:Legal_Role A legal role is a role played in a legal context. Legal role players can be both Agents and other 'things'

Class legal-role:Professional_Legal_Role A professional legal role is a legal profession of some person, examples: lawyer, judge etc.

Class legal-role:Social_Legal_Role A social legal role is played by some agent in the context of legal activities.

5.2.5 Mereology Module

Direct Imports
top module, see Section 5.2.14
Indirect Imports
time module, see Section 5.2.13 *place* module, see Section 5.2.8 **Short description**
The mereology module defines mereological concepts such as parts and wholes, and typical mereological relations such as part of, component of, containment, membership etc.

Class mereo:Atom An atom has no parts

ObjectProperty mereo:component Specifies that some thing is a (functional) component of some other thing

ObjectProperty mereo:component_of Specifies that some thing is a (functional) component of some other thing

ObjectProperty mereo:composed_of Specifies that some thing is composed_of (spatially) within some other thing

Class mereo:Composition A composition has multiple parts, the components should meet

ObjectProperty mereo:contained_in Specifies that some thing is contained (spatially) within some other thing

ObjectProperty mereo:direct_part The non-transitive part relation.

ObjectProperty mereo:direct_part_of The non-transitive part_of relation

ObjectProperty mereo:member Specifies membership of a set or group

ObjectProperty mereo:member_of Specifies membership of a set or group

Class mereo:Pair A composition of exactly two parts

ObjectProperty mereo:part Transitive part relation

Class mereo:Part A part is a part_of some whole

ObjectProperty mereo:part_of Transitive part_of relation

Class mereo:Whole A whole as several parts

5.2.6 Modification Module

Direct Imports
legal action module, see Section 5.2.3 *time* module, see Section 5.2.13
Indirect Imports
norm module, see Section 5.2.7 *role* module, see Section 5.2.10 *action* module, see Section 5.2.1 *expression* module, see Section 5.2.2 *process* module, see Section 5.2.9 *top* module, see Section 5.2.14 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *space-time* module, see Section 5.2.12
Short description
The modification module is both an extension of the time module and the legal action module. The time module is extended with numerous intervals and moments describing the efficacy and being in force of legal documents. The action module is extended with a typology of modifications.

Class modification:Annulment

Class modification:Application When a local act applies in the local legal system normative supra-ordered such as when a region applies an EU directive or a decree it applies the normative contant in some act

ObjectProperty modification:application

Class modification:Application_Date The date when a modification is applied to the destination legislative document. The date can be instantaneous, in the future or in the past. Normally this date coincides with the date of efficacy of the text in which the modification takes place.

Class modification:Application_Interval This is the interval during which a normative document, or fragment, produces the consequences that the normative provision establishes.

DatatypeProperty modification:date

Class modification:Delivery_Date The date when the competent authorities finalise the document by affixing their signatures to it (e.g. promulgation by the president, signature by the queen etc.) This is the date shown on the document itself: it is an objective element and clearly identifyable.

Class modification:Deregulation When a part of some primary legislative source is delegated to some secondary legislative source (e.g. from act to regulation).

ObjectProperty modification:duration

Class modification:Dynamic_Temporal_Entity A dynamic temporal entity represents the norm in its evolution over time.

ObjectProperty modification:efficacy

Class modification:Efficacy_Interval The period during which a normative fragment may be either operative or inoperative, by explicit provision of the document itself. The document is said to be in a period of operation, or enforceability or efficacy, when it may or must be applied or enforced.

Class modification:Efficacy_Modification Modifies the efficacy period of some norm

Class modification:End_efficacy

Class modification:End_in_Force When an act states the end of the period of enactment, and includes an implicit repealing of the text.

Class modification:Enter_in_Force_Date The date when a dnormative document becomes law and enters the legal system for the first time. This is the moment in the doucment's history starting from which the document can be amended, its applicability assessed, and its manner of producing an effect determined.

Class modification:Exception Restricts the scope of normative content to some particular jurisdiction

Class modification:Existence_Date The date when the formal act by which a legislative body freezes the document into its final form, it is the time when the document can be said to have begun its existence.

Class modification:Extension Extends the scope of normative content to some particular jurisdiction

ObjectProperty modification:final_date

ObjectProperty modification:final_date_of

ObjectProperty modification:in_force

Class modification:In_Force_Interval The period during which a normative fragment belongs to the normative system. The period of force for each fragment may change over time as a function of the modifications the document goes through.

Class modification:In_Force_Modification Modifies the period of being in force of some norm

ObjectProperty modification:initial_date

ObjectProperty modification:initial_date_of

Class modification:Integration

Class modification:Interpretation Modification of the interpretation of normative content by the author or some superior actor

Class modification:Modification See [Palmirani et al., 2007]

Class modification:Modification_of_Meaning A modification that changes the meaning of some provision without changing its literal text

Class modification:Modification_of_Scope See [Palmirani et al., 2007]

Class modification:Modification_of_System See [Palmirani et al., 2007]

Class modification:Modification_of_Term Modification of the term specified by some procedural deadline.

ObjectProperty modification:produce_efficacy_modification

ObjectProperty modification:produce_inforce_modification

ObjectProperty modification:produce_textual_modification

Class modification:Prorogation_Efficacy Prorogation of entry into force, but applied to the efficacy of some norm

Class modification:Prorogation_in_Force Prorogation of the date of enactment of some normative content. The 'life' of the act is lengthened with respect to some previous end of enactment modification.

Class modification:Publication_Date The date when the normative document is published in the official journal designated as the source for making all such documents public and legal.

Class modification:Ratification When an international treaty is ratified by local parliament in order to include it in the local legal system. The same thing happens when each local parliament ratifies bilateral or multilateral agreements.

Class modification:Relocation Moving some part of the text to some other place within the same document

Class modification:Remaking When an act is completely rewritten in a new way but the topic remains the same.

Class modification:Renewal

Class modification:Repeal Removal of some part of text or the entire act from the legal system.

Class modification:Retroactivity When the efficacy or a partial efficacy starts before the entry into force of the document

Class modification:Semantic_Annotation See [Palmirani et al., 2007]

Class modification:Start_Efficacy

Class modification:Start_in_Force

Class modification:Static_Temporal_Entity A static temporal entity captures a single moment in the norm's life that is fixed over time.

Class modification:Substitution Substitution of a part or entire act or annex

Class modification:Suspension When a part of an act or some provisions are suspended in coming to be operative for a period or indefinately

Class modification:Temporal_Modification Modifies the efficacy or being in force of some norm

Class modification:Textual_Modification See [Palmirani et al., 2007]

Class modification:Transposition Interpretation in context of the European Union

Class modification:Ultractivity

Class modification:Variation Modification by paraphrasing

5.2.7 Norm Module

Direct Imports
legal action module, see Section 5.2.3 *expression* module, see Section 5.2.2
Indirect Imports
role module, see Section 5.2.10 *action* module, see Section 5.2.1 *expression* module, see Section 5.2.2 *process* module, see Section 5.2.9 *top* module, see Section 5.2.14 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *time* module, see Section 5.2.13 *space-time* module, see Section 5.2.12
Short description
This module is an extension primarily on the expression module where norms are defined as qualifications. Please refer to [Boer et al., 2007] for a more in-depth description of the underlying theory. It furthermore defines a number of legal sources, e.g. legal documents, customary law etc., and a typology of rights and powers, cf. [Sartor, 2006, Rubino et al., 2006]

Class norm:Action_Power An action-power consists in a generic power to produce a legal effect through an action determining it. Also called Subjections

Class norm:Allowed See normatively qualified. Alternative labels: permitted, sanctioned, let, licensed.

Class norm:Allowed_And_Disallowed Something which is both allowed and disallowed through some norm

ObjectProperty norm:allowed_by Relates a qualified Allowed to the norm allowing it

ObjectProperty norm:allows Relates a norm to the thing it allows

Class norm:Belief_In_Violation A belief in some violation

Class norm:Code A legal code bears one or more norms, all of which are uttered by some legislative body. It cannot bear expressions which are not uttered by a legislative body.

Class norm:Code_of_Conduct a code outlining the responsibilities of or best practice for an individual or organisation, such as a set of principles of good corporate behaviour adopted by a business

ObjectProperty norm:commanded_by

ObjectProperty norm:commands

Class norm:Contract A contract bears one or more norms, all of which are uttered by some natural person or legal person. It cannot bear expressions which are uttered by a different kind of agent.

Class norm:Custom The collective memory of some group of agents, i.e. propositions borne by custom are shared (i.e. held) by all members of the group

Class norm:Customary_Law In law, custom, or customary law consists of established patterns of behaviour that can be objectively verified within a particular social setting.

Class norm:Declarative_Power We say that j has the declarative power to realize A to mean that if j declares A, then it is legally valid that A. For example, if x has the declarative power to terminate y's obligation toward x to do then if x declares that y's obligation toward x finishes, then it is legally valid that this obligation finishes.

Class norm:Decree The word decree is often used as a derogative term for any authoritarian decision

Class norm:Definitional_Expression A definition in a legal context (for example 'x means y' or 'by x it is meant y');

Class norm:Directive Examples are European Union directive, a legislative act of the European Union and Directives, used by United States Government agencies (particularly the Department of Defense) to convey policies, responsibilities, and procedures.

Class norm:Disallowed See Qualified. Alternative labels: violation, prohibited, forbidden

ObjectProperty norm:disallowed_by Relates a qualified Disallowed to the norm disallowing it

Class norm:Disallowed_Intention The propositional content (i.e. that which is intended) of the intention is disallowed. Interesting concept for establishing mens rea (guilty mind), although 'disallowed' is a weaker qualification than criminal. Note the distinction between purposely committing a crime and knowingly committing a crime. The crime committed does not have to match the crime intended for establishing intent!

ObjectProperty norm:disallows Relates a norm to the thing it disallows

Class norm:Enabling_Power An enabling power is used in cases where the law aims at enabling some agent to produce an effect in a particular way.

Class norm:Evaluative_Expression A legal evaluative expression asserts that something is good or bad, is a value to be optimised or an evil to be minimised (for example 'human dignity is value', 'participation ought to be promoted');

Class norm:Exclusionary_Right An exclusionary right concerns the prohibition against performing certain inferences (against reasoning in certain ways), or against using certain kinds of premises for certain purposes, in the interest of a particular person. This is especially the case with anti-discrimination rules.

Class norm:Existential_Expression Establishes or terminates the existence of a legal entity (for example 'the company ceases to exist');

Class norm:Hohfeldian_Power A hohfeldian power covers any action which determines a legal effect.

Class norm:Immunity Also called Disabilities

Class norm:International_Agreement An agreement entered into by actors in international law, namely states and international organizations.

Class norm:International_Arbitration International Arbitration is the established method today for resolving disputes between parties to international commercial agreements. As with arbitration generally, it is a creature of contract, i.e., the parties' decision to submit any disputes to private adjudication by one or more arbitrators appointed in accordance with rules the parties themselves have agreed to adopt, usually by including a provision for the same in their contract. The practice of international arbitration has developed so as to allow parties from different legal and cultural backgrounds to resolve their disputes, generally without the formalities of their underlying legal systems.

Class norm:Legal_Doctrine Legal doctrine is a framework, set of rules, procedural steps, or test, often established through precedent in the common law, through which judgements can be determined in a given legal case.

Class norm:Legal_Document SubClasses of legal source can be distinguished through the authority of the agent creating the expression (i.e. the agent holding the norm).

Class norm:Legal_Expression Legal expressions are created by some legal speech act and qualified by a communicated attitude

Class norm:Legal_Source A legal source is a source for legal statements, both norms and legal expressions. In a sense it is literally a ‘source’ of law

Class norm:Liability_Right A liability right expresses some right to act that brings with it some liability to perform another (compensating) act. For instance, if some k performs the permitted action A, then k will have to perform another action B for the benefit of j.

Class norm:Liberty_Right When, for the benefit of a person, this person is both permitted to perform and to omit an action that is, when the action is facultative we can say that he or she has a liberty right with regard to that action.

Class norm:Mandatory_Precedent In law, a binding precedent (also mandatory precedent or binding authority) is a precedent which must be followed by all lower courts. It is usually created by the decision of a higher court, such as the House of Lords in the United Kingdom. Binding precedent relies on the legal principle of stare decisis.

Class norm:Non-binding_International_Agreement Some international agreement that is not binding under international law

Class norm:Norm A norm is a kind of Qualification. A qualification which normatively qualifies some thing (i.e. some normatively qualified): i.e. a qualification which allows or disallows some thing.

ObjectProperty norm:normatively_comparable

ObjectProperty norm:normatively_equivalent_or_better

ObjectProperty norm:normatively_equivalent_or_worse

ObjectProperty norm:normatively_not_equivalent

Class norm:Normatively_Qualified Some thing which is qualified (allowed, disallowed) by a norm, i.e. a norm applies to the thing. Taking the principle of deontic choice to mean that the utterer of a normative statement intends to influence choices made by the addressee of the statement, the qualified thing should is comparable to some alternative. Note that Qualified does not partition into Allowed and Disallowed. Firstly, things can be Allowed by one Norm and Disallowed by another one. Secondly, the logical complement of the thing allowed by a permission is qualified (as worse or equal than the thing allowed), but neither allowed nor disallowed.

ObjectProperty norm:normatively_strictly_better

ObjectProperty norm:normatively_strictly_worse

Class norm:Obligation See prohibition

Class norm:Obligative_Right Directed obligative ought-to-do are also called obligative rights. k has the obligative right that j does A iff it is obligatory, toward k, that j does A. An example of obligative right is 'it is obligatory, toward Mary, that Tom pays 1,000 euro to John'

Class norm:Obliged Is allowed by an Obligation, i.e. the alternative is strictly_worse according to the Obligation. Alternative labels: directed, commanded.

Class norm:Observation_of_Violation An observation of a violation, i.e. of something that is Disallowed.

Class norm:Permission Permission allows propositions, to which therefore the predication Allowed applies. The thing(s) allowed is/are a subset of the thing(s) qualified by the permission. If S then P(A) means that S is qualified and S and A is allowed.

Class norm:Permissive_Right The negation of a directed obligation is a directed permission. However, it counts as a right, namely, a permissive right, only when such negation is aimed at benefitting the author of the permitted action.

Class norm:Persuasive_Precedent In law, a persuasive precedent or advisory precedent is a precedent that need not be followed under the legal principle of stare decisis, but is nevertheless followed. Sources of persuasive precedent include the obiter dicta in the judgement of a court whose judgement would otherwise be binding, and the judgements of courts in other jurisdictions where the facts and/or legal system are similar to the case at hand. If a superior court adopts a persuasive precedent, it may become binding in the jurisdiction.

Class norm:Potestative_Expression Attributes a power to some agent (for example 'a worker has the power to terminate his work contract');

Class norm:Potestative_Right A potestative right is an enabling power which is meant to benefit the holder of the power. For example, if some animal y does not belong to anybody, then x has the potestative-right to start x's ownership of the animal, by capturing y.

Class norm:Precedent In law, a precedent or authority is a legal case establishing a principle or rule that a court may need to adopt when deciding subsequent cases with similar issues or facts. The term may also refer to the collective body of case law that a court should consider when interpreting the law. When a precedent establishes an important legal principle, or represents new or changed law on a particular issue, that precedent is often known as a landmark decision.

Class norm:Proclamation A proclamation (Lat. proclamare, to make public by announcement) is an official declaration.

Class norm:Prohibition Prohibition obliges/allows thing(s), to which therefore the predication Obliged applies, and disallows thing(s), to which therefore the predication Disallowed applies. The union of the thing(s) allowed and the thing(s) disallowed is qualified by the prohibition. The things allowed and disallowed by the prohibition are disjoint (and completely partition the space of things qualified). If S then O(A) means that S is qualified, S and A is obliged/allowed, and S and not A is disallowed. If S then F(A) means that S is qualified, S and A is disallowed, and S and not A is obliged/allowed. The difference between Obligation and Prohibition is in which part of the partition of the qualified space is explicitly described by the normative statement. Alternative label: directive

Class norm:Qualificatory_Expression Ascribes a legal role to a person or an object (for example, ‘x is a citizen’, ‘x is an intellectual work’, ‘x is a technical invention’);

Class norm:Regulation A regulation bears one or more norms, all of which are uttered by some legislative body. It cannot bear expressions which are not uttered by a legislative body.

Class norm:Resolution A resolution is a written motion adopted by a deliberative body. The substance of the resolution can be anything that can normally be proposed as a motion.

Class norm:Right A right is the legal or moral entitlement to do or refrain from doing something or to obtain or refrain from obtaining an action, thing or recogition in civil society.

Class norm:Soft_Law The term ‘soft law’ refers to quasi-legal instruments which do not have any binding force, or whose binding force is somewhat ‘weaker’ than the binding force of traditional law, often referred to as ‘hard law’, in this context. The term ‘soft law’ initially appeared in the area of international law, but more recently it has been transferred to other branches of law.

Class norm:Statute A statute bears one or more norms, all of which are uttered by some legal person. It cannot bear expressions which are uttered by a different kind of agent.

Class norm:Strictly_Allowed Loose end.

Class norm:Strictly_Disallowed Loose end.

ObjectProperty norm:strictly_equivalent

Class norm:Treaty A treaty is a binding agreement under international law entered into by actors in international law, namely states and international organizations. Treaties are called by several names: treaties, international agreements, protocols, covenants, conventions, exchanges of letters, exchanges of notes, etc.

5.2.8 Place Module

Direct Imports
top module, see Section 5.2.14
Indirect Imports
mereology module, see Section 5.2.5 *time* module, see Section 5.2.13 *space-time* module, see Section 5.2.12
Short description
This module partially[2] implements the theory of relative places ([Donnelly, 2005]) in OWL.

Class place:Absolute_Place An absolute place is defined without reference to other places.

ObjectProperty place:abut Two places abut if they meet but do not partially coincide. NOTE: OWL 1.1: 'abut' is irreflexive and disjoint with 'partially-coincide'

Class place:Comprehensive_Place A place is comprehensive if it always covers every place (and thus also every location-complex) and every object.

ObjectProperty place:connect Two places are connected if they are both relatively fixed and when they meet

ObjectProperty place:cover Specifies that two places cover each other. Note: this cover-relation is the inverse of the COV relation described by Donnelly (2005)

ObjectProperty place:covered_by Specifies that two places cover each other. To add in OWL 1.1: 'covered-by' is reflexive, 'covered-by' is transitive

ObjectProperty place:exactly_coincide To add in OWL 1.1: 'exactly-coincide' is reflexive, 'exactly-coincide' is symmetric, 'exactly-coincide' is transitive

ObjectProperty place:externally_connect Two places are externally connected if they are relatively fixed and abut, but do not overlap (OWL 1.1)

ObjectProperty place:in The ‘in’ property is used to express that one place is located (i.e. contained) within another place. It is therefore a mereological relation as well.

Class place:Location_Complex A location complex is a relatively stable complex of places referred to as one: it is the maximal collection of places with the same reference object.

ObjectProperty place:location_complex

ObjectProperty place:location_complex_for

ObjectProperty place:meet Specifies that two places meet, but need not overlap or cover (reflexive)

ObjectProperty place:overlap Two places overlap if they are relatively fixed and partially coincide

ObjectProperty place:partially_coincide To add in OWL 1.1: ‘partially-coincide’ is reflexive, ‘partially-coincide’ is disjoint with ‘abut’

Class place:Place A place is a spatio-temporal-occurrence

Class place:Relative_Place A relative place is defined by some reference to some object/thing

ObjectProperty place:relatively_fixed Two places are relatively fixed if and only if either x and y have a common reference object or neither x nor y has a reference object

ObjectProperty place:spatial_reference The reference to an object determines the relative place.

ObjectProperty place:spatial_relation A spatial relation is a relation between two places

5.2.9 Process Module

Direct Imports
role module, see Section 5.2.10 *space-time* module, see Section 5.2.12
Indirect Imports
action module, see Section 5.2.1 *expression* module, see Section 5.2.2 *process* module, see Section 5.2.9 *top* module, see Section 5.2.14 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *time* module, see Section 5.2.13
Short description
The process module extends the LKIF top ontology module with a definition of changes, processes (being causal changes) and physical objects. It introduces a limited set of properties for describing participant roles of processes.

Class process:Change A change is a difference between the situation before and after the change occurs (the event of the change). A change can be instantaneous

ObjectProperty process:created_by Specifies that some thing is created (i.e. a result of) by a process, and exists because of the process taking place.

ObjectProperty process:creation Specifies that some thing is created (i.e. a result of) by a process, and exists because of the process taking place.

ObjectProperty process:participant A participant is someone or something that participates in a change, i.e. is involved in a change

ObjectProperty process:participant_in Specifies that some thing participates in a process

Class process:Physical_Object A physical object is a physical concept consisting of matter, it has a spatio-temporal extension.

Class process:Process A process is a 'causal' change: any change which can be explained through some known or understood causal structure. Every process has some Time_Period as duration.

ObjectProperty process:requirement A requirement relation relates the process with a prerequisite for that process: without it the process cannot occur

ObjectProperty process:requirement_of Specifies that some participant is a requirement for a process

ObjectProperty process:resource A resource is some quantity of something used to perform the action: i.e. time, energy

ObjectProperty process:resource_for Specifies that some participant is a resource for a process

ObjectProperty process:result Specifies that some participant is the result of the process, it might have existed before the process took place, but is in some way altered

ObjectProperty process:result_of Specifies that some participant is the result of a process, it might have existed before the process took place, but is in some way altered (an 'inanimate' goal of an act)

5.2.10 Role Module

Direct Imports
action module, see Section 5.2.1
Indirect Imports
expression module, see Section 5.2.2 *process* module, see Section 5.2.9 *top* module, see Section 5.2.14 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *time* module, see Section 5.2.13 *space-time* module, see Section 5.2.12
Short description
The role module defines a typology of roles (epistemic roles, functions, person roles, organisation roles) and the plays-property for relating a role filler to a role.

Class role:Epistemic_Role The role of something used in a reasoning/inference process

Class role:Function The use or purpose of some object as used in some context.

Class role:Organisation_Role A role which has a meaning in the context of an organisation: i.e. the role defines a 'position' within sthe structure of roles within an organisation

Class role:Person_Role A role played by some person (i.e. not an organisation)

ObjectProperty role:played_by Specifies that some role is played by some thing

ObjectProperty role:plays Specifies that some thing plays a role

Class role:Role A role is a specification of default behaviour and accompanying expectations of the thing 'playing' the role. Similar to actors in a theater who play roles, but are not the roles. Example: student.

Class role:Social_Role A social role is played by some agent in the context of social activities. The social role brings with it some expectation of 'default' behaviour of the role-filler.

5.2.11 Rules & Argumentation Module

Direct Imports
expression module, see Section 5.2.2
Indirect Imports
action module, see Section 5.2.1 *process* module, see Section 5.2.9 *role* module, see Section 5.2.10 *top* module, see Section 5.2.14 *mereology* module, see Section 5.2.5 *place* module, see Section 5.2.8 *time* module, see Section 5.2.13 *space-time* module, see Section 5.2.12
Short description
The argumentation rules module describes the vocabulary for LKIF rules as defined in [Boer et al., 2007], chapter 5. Note that the classes defined in this module are not extensions of SWRL rules, but rather define roles central to argumentation and LKIF rules. As the module does not commit to a particular technical representation of rules, the module leaves room for further extension to complex argumentation frameworks (cf. the proposal described in Section 5.3).

ObjectProperty rules:applies Specifies whether some rule applies to some atom

Class rules:Argument An argument is some rule used in argumentation (cf. [Boer et al., 2007])

Class rules:Assumption An assumption is some atom held to be true, without proof, in the head of a rule (defeasibly), cf. [Boer et al., 2007]

Class rules:Atom An atom is the most basic (undivisible) part of an LKIF rule (cf. [Boer et al., 2007])

Class rules:Exception An exception states an exception to the head of the rule (defeasibly), cf. [Boer et al., 2007]

ObjectProperty rules:excluded Specifies an exclusion relation between a rule and an atom

Class rules:Negated_Atom A negated atom is the negation of some other atom (cf. [Boer et al., 2007])

ObjectProperty rules:prior Specifies a prior relation between two rules

ObjectProperty rules:rebuts Specifies whether some rule abuts another rule

Class rules:Rule An LKIF rule.

ObjectProperty rules:rule_predicate Some predicate in or over some rule

Class rules:Valid_Rule A valid LKIF rule

5.2.12 Space-Time Module

Direct Imports
place module, see Section 5.2.8 *time* module, see Section 5.2.13
Indirect Imports
top module, see Section 5.2.14 *mereology* module, see Section 5.2.5
Short description
The Space-Time module is a mere placeholder for the place and time modules.

5.2.13 Time Module

Direct Imports
mereology module, see Section 5.2.5
Indirect Imports
top module, see Section 5.2.14 *place* module, see Section 5.2.8
Short description
The time module provides an OWL implementation of the theory of time by [Allen, 1984].

ObjectProperty time:after Specifies that a time period occurs after another period, but might overlap with it.

ObjectProperty time:before Specifies that a time period occurs before another period, but might overlap with it.

ObjectProperty time:between Specifies that a time period occurs between two other periods

ObjectProperty time:during Specifies that a time period occurs during another period

ObjectProperty time:finish Specifies that a time period finishes another period, i.e. the other period starts before, but ends at the same moment

ObjectProperty time:immediatly_after Specifies that a time period occurs immediately after another period, i.e. it starts where the other period ends

ObjectProperty time:immediatly_before Specifies that a time period occurs immediately before another period, i.e. it ends where the other period starts

Class time:Interval An interval is a composition of multiple periods of time.

Class time:Moment A moment is an atomic period of time, i.e. it cannot be divided into other parts

ObjectProperty time:overlap Specifies that a time period overlaps with another time period (in any way)

Class time:Pair_Of_Periods A pair of two time periods

ObjectProperty time:preceeds Specifies that a time period preceeds another period, the periods do not overlap

ObjectProperty time:start Specifies that a time period starts another period, i.e. the other period starts at the same time, but ends at a later time.

Class time:Temporal_Occurrence A period of time, has a duration

ObjectProperty time:temporal_relation A relation between two time periods

5.2.14 Top Module

Direct Imports
place module, see Section 5.2.8 *time* module, see Section 5.2.13
Indirect Imports
mereology module, see Section 5.2.5
Short description
This module defines the top ontology for LKIF Core, it is largely based on the top-level of LRI-Core but has less ontological commitment in the sense that it imposes less restrictions on subclasses of the top categories.

Class top:Abstract_Concept An abstract concept is a concept which does not (necessarily) have a referent in either the physical or the mental world.

Abstract entities are (proto)mathematical, formal entities, entities which have a purely formal, logical or mathematical meaning.

Class top:Mental_Concept Mental concepts are those concepts that reside ‘in our heads’. These can be thoughts, attitudes, memories etc. But also the meaning of the written word is something that does not exist outside our minds. Mental concepts are metaphors for physical concepts.

Class top:Mental_Object Metaphor of physical object, i.e. the things we mentally manipulate, either in thought or memory.

Class top:Occurrence Occurrences happen against the canvas of space and time. This canvas allows us to index instances of objects and processes. Time and space provide positions in time and space for objects. Positions are in general modelled in some mereological framework.

Class top:Physical_Concept In LKIF core, the physical world evolves around two main Classes: physical objects and processes. Objects are pieces of matter, while matter is typed by substances. Objects have mass, extension, viz. form and aggregation state (which limits form). Objects are the specification of the notion that matter, and in particular solid objects, is what makes the physical world relatively stable and observable. We define (physical) situations in the first place by the arrangement of physical objects (entities).

Class top:Spatio_Temporal_Occurrence An occurrence in space-time.

5.3 Towards an argumentation framework: the AIF-project

5.3.1 Introduction

In this section, we recommend the *Argument Interchange Format* (AIF), as proposed in Chesnevar et. al. [Chesnevar et al., 2006] and Rahwan et. al. [Rahwan and Zablith, 2007], as a starting point for an argument interchange format and ontology. We outline the AIF project, some of the key elements and extensions, their relevance to the Estrella Project, and the reason for our recommendation.

5.3.2 Outline of AIF Project

Chesnevar et. al. [Chesnevar et al., 2006] is a recently published proposal for an AIF. The origin of the paper is an AgentLink Technical Forum Group meeting in Hungary, September 2005, of researchers interested in argumentation and computation. Subsequently, the proposal was presented at several conferences; the journal publication represents the latest version available. The proposal, which is a draft, represents a consensus 'abstract model' among spectrum of researchers in argumentation, AI, and multi-agent systems. While the abstract model is not specifically designed to be a *legal* argumentation format, there is already an *extension* to the basic model which express argumentation schemes such as proposed in Walton [Walton, 1996] (cf. Rahwan et. al. [Rahwan and Zablith, 2007]); Walton's schemes are clearly relevant to legal argumentation (cf. Gordon and Walton [Gordon and Walton, 2006]).

The main aims are:

1. to facilitate the development of (closed or open) multi-agent systems capable of argumentation-based reasoning and interaction using a shared formalism;

2. to facilitate data interchange among tools for argument manipulation and argument visualisation.

The AIF has been designed in reaction to Argument Markup Languages, which, the authors claim, do not have rich enough semantics to support processing of arguments.[3] The AIF is also intended to *consolidate* elements of argumentation theory, which have been spread across a range of fields and approaches.

5.3.3 Key Elements of AIF and Extensions

We discuss some of the main elements of AIF, uses and reifications, as well as some extensions.

[3] The claim that an interchange language has a semantics is, in and of itself, rather interesting in our view and deserves greater examination. The authors are explicit that the AIF is just another data representation, which requires a syntax and semantics.

Overall Approach

We give some of the key points of the AIF, leaving further details to the interested reader. The AIF has several design points: it ought to be *machine readable syntax* and have a *machine processable semantics*; there ought to be a unified abstract model which can be specified for further concrete cases; the core concepts ought to allow for extensions with other concepts.

The AIF is presented as a *core ontology* of concepts related to argumentation. It is extensible and abstract. There are claimed to be three main ontological subgroups: arguments and argument networks; communication; and context. We only touch very briefly on the first, as the second two are not under active research in the Estrella Project.

The fundamental notion is that arguments are *nodes* in a directed graph; a set of nodes in a graph is an *argument network*. Some of the nodes represent *information* such as claims, premises, data, while so-called *scheme* nodes represent reasoning patterns. The nodes can have a range of attributes specified for them, which can vary from ontology to ontology. *Edges*, which are relations between nodes, represent relations between information and reasoning patterns: for example, an edge from an information node to an inference node indicates that the information is input to the inference. Another reasoning pattern is *attack*, the attacker being linked to the attacked argument via the attack node. A range of relations and restrictions on relations between the nodes are specified. Using such fundamental notions, the authors claim to be able to represent a variety of argumentation structures such as Toulmin's scheme.

Use Cases and Reifications for AIF

Two use cases are described and three reifications are described using AIF. We briefly mention those which are relevant to ESTRELLA: the ASPIC inference component, and the RDF schema.

ASPIC Argumentation Service Platform with Integrated Components (ASPIC) seeks to develop theoretical models of argumentation suitable for inference, decision-making, protocols for dialectical argumentation, and machine learning. A brief description of the components of the inference engine is provided, where structures for arguments are provided, arguments are valuated, attack and defeat relations are defined, and the dialectical status of arguments is determined along the lines of Dung [Dung, 1995].

RDF The AIF is presented in terms of the resource description framework (RDF) (cf. Rahwan and Sakeer [Rahwan and Sakeer, 2006]), using the Protege ontology development environment. In the ontology, the general *node* has several subclasses, among them information and scheme notes, which in turn each have a range of subclasses. Restrictions can be placed on the numbers of nodes in a particular representation. For example, the Toulmin scheme has but one claim. The edges, which are relationships between nodes, can also be structured and restricted in the ontology. They provide an RDF graph for a sample Toulmin argument. Furthermore, the structures support queries such that, given the argument, one can query the conclusion.

Extensions

Rahwan et. al. [Rahwan and Zablith, 2007] extend the AIF presented in Chesnevar et. al. [Chesnevar et al., 2006] to argument schemes, particularly those of Walton [Walton, 1996]. The fundamental idea appears to be to further articulate the information nodes, reasoning nodes, and their relationships so as to allow for representations of schemes such as the *expert opinion argumentation scheme*.

Critique

There are components of the AIF which may be relevant to the Estrella Project, but which have not, to this point, been discussed (to our knowledge). In particular, the AIF takes into consideration *multi-agent systems*, where *communication* among the participants is a key notion. There is, then, a significant discussion of speech acts, communication protocols, context, and participants. While these are not prominent topics in the Estrella Project, it is recognised that they are nonetheless significant in argumentation theory (cf. Gordon [Gordon, 1995]).

5.3.4 Relevance to the Estrella Project

The AIF is clearly highly relevant to the Estrella Project, particular that part which is addressing argumentation. AIF and Estrella share many high-level aims, develop the same conceptual space (e.g. argumentation schemes), are concerned with argumentation and inference, and want their expressions to function in the Semantic Web.

5.3.5 Recommendation to the Estrella Project

With these points in mind, we recommend the AIF as a starting point for further development, with particular reference to legal argumentation. Broadly, the means and goals of the AIF are similar to the Estrella Project, namely, to provide an abstract interchange format for argumentation which has an ontology and is suitable for representation on the semantic web using OWL and RDF. There is little reason to reproduce what work has already been done, but rather an advantage to take it as a starting point, which can be critiqued and developed for the Estrella Project. As a published proposal, the AIF has established itself as the *defacto* benchmark against which further developments in the same vein are measured. The AIF attempts to form a consensus approach to argumentation, drawing from a wide spectrum of research and applications. Moreover, using the AIF proposal links the research in the Estrella Project with a wider community of researchers interested in very similar issues and around a relevant proposal. Indeed, examination and development of the AIF proposal with specific application to the materials in the Estrella Project would test the claims of the AIF. Our recommendation does not imply that we agree with AIF as it currently stands, just that it is a suitable starting point for further discussion. Where members of the Estrella Project argue for an alternative, we can do so by showing how the our proposal is different and better than AIF.

Appendix A

Appendix

A.1 Scores obtained in assessing legal concepts

The tables below contain the mean scores from five partners in Estrella of the most important, ‘top 50’ terms on the original list of 243 terms.

Term	Importance for the Ontology
law	1,00
right	1,00
jurisdiction	1,20
permission	1,20
prohibition	1,20
rule	1,20
sanction	1,20
violation	1,20
power	1,25
duty	1,40
legal position	1,40
norm	1,40
obligation	1,40
permissive right	1,40
argument	1,60
authority	1,60
damage	1,60
legal person	1,60
legal procedure	1,60
liability	1,60
penalty	1,60
responsibility	1,60
natural person	1,75
agreement	1,80
contract	1,80
deontic operator	1,80
fact	1,80
legal fact	1,80
legislation	1,80
source of law	1,80
tort	1,80
citizen	2,00
competence	2,00
crime	2,00
defence	2,00
equity, fairness	2,00
lawyer	2,00
legal domain	2,00
negligence	2,00
penalty	2,00
permit	2,00
potestative right	2,00
privilege	2,00
proof	2,00
punishment	2,00
requirement	2,00
beneficiary	2,20
court	2,20
defendant	2,20

Table A.1: Top 50 of most important concepts

Term	Level of Abstraction
deontic operator	1,25
law	1,25
norm	1,25
obligative right	1,25
permissive right	1,25
power	1,25
right	1,25
rule	1,25
time	1,25
anancastic rule	1,33
existential initiation	1,33
existential termination	1,33
potestative right	1,33
productive characterisation = brings	1,33
absolute obligative right	1,50
absolute permissive right	1,50
absolute right	1,50
deformation	1,50
deontic emergence	1,50
deontic initiation	1,50
deontic termination	1,50
duty	1,50
enabling power	1,50
equity, fairness	1,50
permission	1,50
position	1,50
premise	1,50
prohibition	1,50
reified relationships	1,50
responsibility	1,50
facultativeness = freedom	1,67
other directed norm	1,67
qualificatory emergence	1,67
qualificatory initiation	1,67
qualificatory negative initiation	1,67
argument	1,75
assumption	1,75
conduct	1,75
constitutive rule	1,75
criteria	1,75
innocence	1,75
legal domain	1,75
negative deontic initiation	1,75
presumption	1,75
presupposition	1,75
proclamative power	1,75
proof	1,75
requirement	1,75
action power	2,00

Table A.2: Top 50 of most abstract concepts

Term	Relevance for Legal Domains
civil law	1,00
law	1,00
legal consequence	1,00
legislation	1,00
obligation	1,00
right	1,00
authority	1,25
deontic operator	1,25
duty	1,25
jurisdiction	1,25
legal fact	1,25
legal person	1,25
legal position	1,25
legal procedure	1,25
liability	1,25
negligence	1,25
permission	1,25
prohibition	1,25
rule	1,25
to judge	1,25
to prosecute	1,25
tort	1,25
violation	1,25
action	1,50
agreement	1,50
argument	1,50
citizen	1,50
claim	1,50
compensation	1,50
competence	1,50
Constitution	1,50
crime	1,50
criminal	1,50
criminal law	1,50
damage	1,50
defendant	1,50
evidence	1,50
guilt	1,50
high court	1,50
immunity	1,50
judge	1,50
jurisprudence	1,50
lawyer	1,50
norm	1,50
penalty	1,50
permissive right	1,50
plaintiff	1,50
power	1,50
privilege	1,50

Table A.3: Top 50 of most legally relevant concepts

A.2 Initial Definitions

This section lists the formal specification of the concepts defined in the initial modules outlined in Section 4.1.3.

A.2.1 Place

$$
\begin{aligned}
\bot &\sqsubseteq \textsc{Absolute-Place} \sqcap \textsc{Relative-Place}\\
\textsc{Location-Complex} &\equiv \exists\textsf{is-location-complex-of}.\textsc{Place}\\
\textsc{Place} &\equiv \textsc{Absolute-Place} \sqcup \textsc{Relative-Place}\\
\textsc{Relative-Place} &\equiv \exists\textsf{reference}.\textsc{Place}\\
\textsf{covered-by} &\equiv \textsf{cover}^{-1}\\
\textsf{relatively-fixed} &\equiv \textsf{relatively-fixed}^{-1}\\
\textsf{location-complex} &\equiv \textsf{is-location-complex-of}^{-1}\\
\textsf{cover} &\equiv \textsf{covered-by}^{-1}\\
\textsf{abut} &\equiv \textsf{abut}^{-1}\\
\textsf{is-location-complex-of} &\equiv \textsf{location-complex}^{-1}\\
\textsf{meet} &\equiv \textsf{meet}^{-1}\\
\textsf{partially-coincide} &\equiv \textsf{partially-coincide}^{-1}\\
\textsf{exactly-coincide} &\equiv \textsf{exactly-coincide}^{-1}\\
\textsf{covered-by} &\sqsubseteq \textsf{partially-coincide}\\
\textsf{cover} &\sqsubseteq \textsf{partially-coincide}\\
\textsf{abut} &\sqsubseteq \textsf{meet}\\
\textsf{meet} &\sqsubseteq \textsf{spatial-relation}\\
\textsf{partially-coincide} &\sqsubseteq \textsf{meet}\\
\textsf{exactly-coincide} &\sqsubseteq \textsf{cover}\\
\textsf{exactly-coincide} &\sqsubseteq \textsf{covered-by}\\
\textsc{Comprehensive} &\sqsubseteq \textsc{Location-Complex}\\
\textsc{Location-Complex} &\sqsubseteq \textsc{Place}\\
\textsc{Absolute-Place} &\sqsubseteq \textsc{Place}\\
\textsc{Place} &\sqsubseteq \exists\textsf{location-complex}.\textsc{Location-Complex}\\
\textsc{Relative-Place} &\sqsubseteq \textsc{Place}\\
&\hspace{-8em}\textsc{Trans}(\textsf{covered-by})\\
&\hspace{-8em}\textsc{Trans}(\textsf{relatively-fixed})\\
\textsf{relatively-fixed} &\equiv \textsf{relatively-fixed}^{-1}\\
\top &\sqsubseteq\ \leq 1\ \textsf{location-complex}.\top\\
&\hspace{-8em}\textsc{Trans}(\textsf{cover})\\
\textsf{abut} &\equiv \textsf{abut}^{-1}\\
\top &\sqsubseteq\ \leq 1\ \textsf{is-location-complex-of}^{-1}.\top\\
\textsf{meet} &\equiv \textsf{meet}^{-1}\\
\textsf{partially-coincide} &\equiv \textsf{partially-coincide}^{-1}\\
&\hspace{-8em}\textsc{Trans}(\textsf{exactly-coincide})\\
\textsf{exactly-coincide} &\equiv \textsf{exactly-coincide}^{-1}\\
\top &\sqsubseteq \forall\textsf{covered-by}^{-1}.\textsc{Place}\\
\top &\sqsubseteq \forall\textsf{cover}^{-1}.\textsc{Place}\\
\top &\sqsubseteq \forall\textsf{abut}^{-1}.\textsc{Place}
\end{aligned}
$$

$$\begin{aligned}
\top &\sqsubseteq \forall \text{meet}^{-1}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{partially-coincide}^{-1}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{exactly-coincide}^{-1}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{covered-by}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{cover}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{abut}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{meet}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{partially-coincide}.\textsc{Place}\\
\top &\sqsubseteq \forall \text{exactly-coincide}.\textsc{Place}
\end{aligned}$$

A.2.2 Time

$$\begin{aligned}
\textsc{Interval} &\equiv \textsc{Time\_Period} \sqcap \neg\textsc{Moment}\\
\textsc{Moment} &\equiv \textsc{Time\_Period} \sqcap \neg\textsc{Interval}\\
\textsc{Time\_Period} &\equiv \exists \text{immediatly\_after}.\textsc{Time\_Period} \sqcap \exists \text{immediatly\_before}.\textsc{Time\_Period}\\
\text{immediatly\_before} &\equiv \text{immediatly\_after}^{-1}\\
\text{before} &\equiv \text{after}^{-1}\\
\text{after} &\equiv \text{before}^{-1}\\
\text{immediatly\_after} &\equiv \text{immediatly\_before}^{-1}\\
\text{immediatly\_before} &\sqsubseteq \text{before}\\
\text{overlap} &\sqsubseteq \text{temporal\_relation}\\
\text{between} &\sqsubseteq \text{temporal\_relation}\\
\text{during} &\sqsubseteq \text{temporal\_relation}\\
\text{preceeds} &\sqsubseteq \text{temporal\_relation}\\
\text{before} &\sqsubseteq \text{temporal\_relation}\\
\text{transitionpoint} &\sqsubseteq \text{temporal\_relation}\\
\text{span} &\sqsubseteq \text{temporal\_relation}\\
\text{after} &\sqsubseteq \text{temporal\_relation}\\
\text{immediatly\_after} &\sqsubseteq \text{after}\\
\text{finish} &\sqsubseteq \text{temporal\_relation}\\
\text{start} &\sqsubseteq \text{temporal\_relation}\\
\textsc{Interval} &\sqsubseteq \textsc{Time\_Period}\\
\textsc{Interval} &\sqsubseteq \textsc{Composition}\\
\textsc{Moment} &\sqsubseteq \textsc{Time\_Period}\\
\textsc{Moment} &\sqsubseteq \textsc{Atom}\\
\textsc{Event} &\sqsubseteq \top\\
\textsc{Time\_Period} &\sqsubseteq \exists \text{immediatly\_after}.\textsc{Time\_Period}\\
\textsc{Time\_Period} &\sqsubseteq \exists \text{immediatly\_before}.\textsc{Time\_Period}\\
\textsc{Pair\_Of\_Periods} &\sqsubseteq \textsc{Pair}\\
\textsc{Pair\_Of\_Periods} &\sqsubseteq \forall \text{component}.\textsc{Time\_Period}\\
\textsc{Occured-Event} &\sqsubseteq \textsc{Event}\\
&\textsc{Trans}(\text{before})\\
&\textsc{Trans}(\text{after})\\
\top &\sqsubseteq \forall \text{immediatly\_before}^{-1}.\textsc{Time\_Period}\\
\top &\sqsubseteq \forall \text{between}^{-1}.\textsc{Time\_Period}\\
\top &\sqsubseteq \forall \text{before}^{-1}.\textsc{Time\_Period}
\end{aligned}$$

$$\begin{array}{rcl}
\top & \sqsubseteq & \forall \mathsf{after}^{-1}.\textsc{Time\_Period} \\
\top & \sqsubseteq & \forall \mathsf{immediatly\_after}^{-1}.\textsc{Time\_Period} \\
\top & \sqsubseteq & \forall \mathsf{immediatly\_before}.\textsc{Time\_Period} \\
\top & \sqsubseteq & \forall \mathsf{between}.\textsc{Pair\_Of\_Periods} \\
\top & \sqsubseteq & \forall \mathsf{before}.\textsc{Time\_Period} \\
\top & \sqsubseteq & \forall \mathsf{after}.\textsc{Time\_Period} \\
\top & \sqsubseteq & \forall \mathsf{immediatly\_after}.\textsc{Time\_Period}
\end{array}$$

A.2.3 Process

$$\begin{array}{rcl}
\bot & \sqsubseteq & \textsc{Agent} \sqcap \textsc{Process} \\
\bot & \sqsubseteq & \textsc{Physical\_Object} \sqcap \textsc{Process} \\
\bot & \sqsubseteq & \textsc{Physical\_Object} \sqcap \textsc{Role} \\
\bot & \sqsubseteq & \textsc{Action} \sqcap \textsc{Physical\_Object} \\
\bot & \sqsubseteq & \textsc{Role} \sqcap \textsc{Change} \\
\textsc{Change} & \equiv & \exists \mathsf{result}.\top \sqcap \exists \mathsf{requirement}.\top \\
\textsc{Process} & \equiv & \exists \mathsf{resource}.\textsc{Time\_Period} \\
\mathsf{result\_of} & \equiv & \mathsf{result}^{-1} \\
\mathsf{resource\_for} & \equiv & \mathsf{resource}^{-1} \\
\mathsf{participant} & \equiv & \mathsf{participant\_in}^{-1} \\
\mathsf{participant\_in} & \equiv & \mathsf{participant}^{-1} \\
\mathsf{requirement} & \equiv & \mathsf{requirement\_of}^{-1} \\
\mathsf{result} & \equiv & \mathsf{result\_of}^{-1} \\
\mathsf{requirement\_of} & \equiv & \mathsf{requirement}^{-1} \\
\mathsf{resource} & \equiv & \mathsf{resource\_for}^{-1} \\
\mathsf{result\_of} & \sqsubseteq & \mathsf{participant\_in} \\
\mathsf{resource\_for} & \sqsubseteq & \mathsf{participant\_in} \\
\mathsf{requirement} & \sqsubseteq & \mathsf{participant} \\
\mathsf{result} & \sqsubseteq & \mathsf{participant} \\
\mathsf{requirement\_of} & \sqsubseteq & \mathsf{participant\_in} \\
\mathsf{resource} & \sqsubseteq & \mathsf{participant} \\
\textsc{Change} & \sqsubseteq & \forall \mathsf{part}.\textsc{Change} \\
\textsc{Change} & \sqsubseteq & \top \\
\textsc{Process} & \sqsubseteq & \leq 2147483647\ \mathsf{actor}.\top \\
\textsc{Process} & \sqsubseteq & \textsc{Change} \\
\textsc{Agent} & \sqsubseteq & \forall \mathsf{participant\_in}.\textsc{Action} \\
\textsc{Physical\_Object} & \sqsubseteq & \top \\
\textsc{Action} & \sqsubseteq & \top \\
\textsc{Action} & \sqsubseteq & \textsc{Process} \\
\top & \sqsubseteq & \forall \mathsf{participant}^{-1}.\top
\end{array}$$

A.2.4 Action

$$\begin{array}{rcl}
\bot & \sqsubseteq & \textsc{Person} \sqcap \textsc{Organisation} \\
\bot & \sqsubseteq & \textsc{Person} \sqcap \textsc{Artifact} \\
\bot & \sqsubseteq & \textsc{Agent} \sqcap \textsc{Action}
\end{array}$$

$$
\begin{aligned}
\textsc{Agent} &\equiv \textsc{Agent}\\
\textsc{Artifact} &\equiv \exists \mathsf{plays}.\textsc{Function}\\
\textsc{Action} &\equiv \exists \mathsf{actor}.\textsc{Agent} \sqcap \exists \mathsf{plays}.\textsc{Role} \sqcap \exists \mathsf{intends}.\textsc{Intention} \sqcap \exists \mathsf{believes}.\textsc{Expectation}\\
\mathsf{actor} &\equiv \mathsf{actor\_in}^{-1}\\
\mathsf{actor\_in} &\equiv \mathsf{actor}^{-1}\\
\mathsf{intends} &\sqsubseteq \mathsf{inverse\_of\_medium}\\
\mathsf{actor} &\sqsubseteq \mathsf{participant}\\
\mathsf{actor\_in} &\sqsubseteq \mathsf{participant\_in}\\
\textsc{Person} &\sqsubseteq \textsc{Physical\_Object}\\
\textsc{Person} &\sqsubseteq \textsc{Agent}\\
\textsc{Reaction} &\sqsubseteq \exists \mathsf{actor}.\textsc{Agent} \sqcap \exists \mathsf{observes}.\textsc{Action}\\
\textsc{Reaction} &\sqsubseteq \textsc{Action}\\
\textsc{Transaction} &\sqsubseteq \forall \mathsf{actor}.\textsc{Agent}\\
\textsc{Transaction} &\sqsubseteq \geq 2\ \mathsf{actor}.\top\\
\textsc{Transaction} &\sqsubseteq \textsc{Action}\\
\textsc{Agent} &\sqsubseteq \top\\
\textsc{Organisation} &\sqsubseteq \textsc{Agent}\\
\textsc{Artifact} &\sqsubseteq \textsc{Physical\_Object}\\
\textsc{Artifact} &\sqsubseteq \forall \mathsf{plays}.\textsc{Role}\\
\textsc{Creation} &\sqsubseteq \textsc{Action}\\
\textsc{Action} &\sqsubseteq \forall \mathsf{actor}.\textsc{Agent}\\
\textsc{Trade} &\sqsubseteq \textsc{Transaction}
\end{aligned}
$$

A.2.5 Role

$$
\begin{aligned}
\textsc{Person-Role} &\equiv \exists \mathsf{played\_by}.\textsc{Person}\\
\textsc{Function} &\equiv \exists \mathsf{played\_by}.\textsc{Physical\_Object} \sqcap \neg\textsc{Agent}\\
\textsc{Role} &\equiv \exists \mathsf{played\_by}.\top\\
\textsc{Organisation-Role} &\equiv \exists \mathsf{played\_by}.\textsc{Organisation}\\
\mathsf{plays} &\equiv \mathsf{played\_by}^{-1}\\
\mathsf{played\_by} &\equiv \mathsf{plays}^{-1}\\
\top &\sqsubseteq \forall \mathsf{plays}^{-1}.\top\\
\top &\sqsubseteq \forall \mathsf{played\_by}^{-1}.\textsc{Role}\\
\top &\sqsubseteq \forall \mathsf{plays}.\textsc{Role}\\
\top &\sqsubseteq \forall \mathsf{played\_by}.\top\\
\textsc{Person-Role} &\sqsubseteq \textsc{Role}\\
\textsc{Agent} &\sqsubseteq \forall \mathsf{plays}.\textsc{Role}\\
\textsc{Function} &\sqsubseteq \textsc{Role}\\
\textsc{Role} &\sqsubseteq \top\\
\textsc{Organisation-Role} &\sqsubseteq \textsc{Role}\\
\bot &\sqsubseteq \textsc{Person-Role} \sqcap \textsc{Function}\\
\bot &\sqsubseteq \textsc{Person-Role} \sqcap \textsc{Organisation-Role}\\
\bot &\sqsubseteq \textsc{Agent} \sqcap \textsc{Role}\\
\bot &\sqsubseteq \textsc{Function} \sqcap \textsc{Organisation-Role}\\
\bot &\sqsubseteq \textsc{Action} \sqcap \textsc{Role}
\end{aligned}
$$

A.2.6 Expression

$$
\begin{aligned}
\textsc{Statement} &\equiv \exists\mathsf{states}.\top \sqcap \exists\mathsf{utterer}.\textsc{Agent} \\
\textsc{Belief} &\equiv \exists\mathsf{believed\_by}.\textsc{Agent} \\
\textsc{Assertion} &\equiv \exists\mathsf{asserts}.\top \\
\textsc{Observation} &\equiv \exists\mathsf{observer}.\textsc{Agent} \\
\textsc{Qualified} &\equiv \exists\mathsf{qualified\_by}.\textsc{Qualification} \\
\textsc{Expression} &\equiv \exists\mathsf{expresses}.\top \\
\textsc{Qualification} &\equiv \exists\mathsf{qualifies}.\textsc{Qualified} \\
\textsc{Declaration} &\equiv \exists\mathsf{declares}.\top \\
\textsc{Statement\_In\_Writing} &\equiv \textsc{Statement} \sqcap \exists\mathsf{author}.\textsc{Agent} \\
\bot &\sqsubseteq \textsc{Expectation} \sqcap \textsc{Observation} \\
\textsc{Statement} &\sqsubseteq \textsc{Expression} \\
\textsc{Belief} &\sqsubseteq \textsc{Expression} \\
\textsc{Intention} &\sqsubseteq \textsc{Expression} \\
\textsc{Expectation} &\sqsubseteq \textsc{Belief} \\
\textsc{Assertion} &\sqsubseteq \textsc{Statement} \\
\textsc{Fact} &\sqsubseteq \top \\
\textsc{Assumption} &\sqsubseteq \textsc{Belief} \\
\textsc{Observation} &\sqsubseteq \textsc{Expression} \\
\textsc{Qualified} &\sqsubseteq \top \\
\textsc{Document} &\sqsubseteq \textsc{Medium} \\
\textsc{Expression} &\sqsubseteq \exists\mathsf{medium}.\textsc{Medium} \\
\textsc{Expression} &\sqsubseteq \top \\
\textsc{Qualification} &\sqsubseteq \textsc{Expression} \\
\textsc{Lie} &\sqsubseteq \textsc{Assertion} \\
\textsc{Surprise} &\sqsubseteq \textsc{Observation} \\
\textsc{Medium} &\sqsubseteq \exists\mathsf{inverse\_of\_medium}.\textsc{Expression} \\
\textsc{Medium} &\sqsubseteq \top \\
\textsc{Declaration} &\sqsubseteq \textsc{Statement} \\
\textsc{Statement\_In\_Writing} &\sqsubseteq \textsc{Statement} \\
\textsc{Agent} &\sqsubseteq \textsc{Medium} \\
\textsc{Problem} &\sqsubseteq \textsc{Observation} \\
\mathsf{expressed\_by} &\equiv \mathsf{expresses}^{-1} \\
\mathsf{utters} &\equiv \mathsf{utterer}^{-1} \\
\mathsf{declares} &\equiv \mathsf{declared\_by}^{-1} \\
\mathsf{medium} &\equiv \mathsf{inverse\_of\_medium}^{-1} \\
\mathsf{states} &\equiv \mathsf{stated\_by}^{-1} \\
\mathsf{declared\_by} &\equiv \mathsf{declares}^{-1} \\
\mathsf{believes} &\equiv \mathsf{believed\_by}^{-1} \\
\mathsf{stated\_by} &\equiv \mathsf{states}^{-1} \\
\mathsf{utterer} &\equiv \mathsf{utters}^{-1} \\
\mathsf{expresses} &\equiv \mathsf{expressed\_by}^{-1} \\
\mathsf{qualifies} &\equiv \mathsf{qualified\_by}^{-1} \\
\mathsf{observer} &\equiv \mathsf{observes}^{-1} \\
\mathsf{believed\_by} &\equiv \mathsf{believes}^{-1}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{observes} &\equiv \mathsf{observer}^{-1}\\
\mathsf{asserted\_by} &\equiv \mathsf{asserts}^{-1}\\
\mathsf{inverse\_of\_medium} &\equiv \mathsf{medium}^{-1}\\
\mathsf{asserts} &\equiv \mathsf{asserted\_by}^{-1}\\
\mathsf{qualified\_by} &\equiv \mathsf{qualifies}^{-1}\\
\mathsf{utters} &\sqsubseteq \mathsf{inverse\_of\_medium}\\
\mathsf{declares} &\sqsubseteq \mathsf{states}\\
\mathsf{author} &\sqsubseteq \mathsf{utterer}\\
\mathsf{states} &\sqsubseteq \mathsf{expresses}\\
\mathsf{declared\_by} &\sqsubseteq \mathsf{stated\_by}\\
\mathsf{believes} &\sqsubseteq \mathsf{inverse\_of\_medium}\\
\mathsf{stated\_by} &\sqsubseteq \mathsf{expressed\_by}\\
\mathsf{utterer} &\sqsubseteq \mathsf{medium}\\
\mathsf{qualifies} &\sqsubseteq \mathsf{expresses}\\
\mathsf{observer} &\sqsubseteq \mathsf{medium}\\
\mathsf{believed\_by} &\sqsubseteq \mathsf{medium}\\
\mathsf{observes} &\sqsubseteq \mathsf{inverse\_of\_medium}\\
\mathsf{asserted\_by} &\sqsubseteq \mathsf{stated\_by}\\
\mathsf{asserts} &\sqsubseteq \mathsf{states}\\
\mathsf{qualified\_by} &\sqsubseteq \mathsf{expressed\_by}\\
\top &\sqsubseteq \forall \mathsf{utters}^{-1}.\textsc{Agent}\\
\top &\sqsubseteq \forall \mathsf{declares}^{-1}.\textsc{Declaration}\\
\top &\sqsubseteq \forall \mathsf{medium}^{-1}.\textsc{Expression}\\
\top &\sqsubseteq \forall \mathsf{states}^{-1}.\textsc{Statement}\\
\top &\sqsubseteq \forall \mathsf{believes}^{-1}.\textsc{Agent}\\
\top &\sqsubseteq \forall \mathsf{utterer}^{-1}.\textsc{Statement}\\
\top &\sqsubseteq \forall \mathsf{expresses}^{-1}.\textsc{Expression}\\
\top &\sqsubseteq \forall \mathsf{qualifies}^{-1}.\textsc{Qualification}\\
\top &\sqsubseteq \forall \mathsf{observer}^{-1}.\textsc{Observation}\\
\top &\sqsubseteq \forall \mathsf{believed\_by}^{-1}.\textsc{Belief}\\
\top &\sqsubseteq \forall \mathsf{observes}^{-1}.\textsc{Agent}\\
\top &\sqsubseteq \forall \mathsf{addressee}^{-1}.\textsc{Statement}\\
\top &\sqsubseteq \forall \mathsf{inverse\_of\_medium}^{-1}.\textsc{Medium}\\
\top &\sqsubseteq \forall \mathsf{asserts}^{-1}.\textsc{Assertion}\\
\top &\sqsubseteq \forall \mathsf{qualified\_by}^{-1}.\textsc{Qualified}\\
\top &\sqsubseteq \forall \mathsf{utters}.\textsc{Statement}\\
\top &\sqsubseteq \forall \mathsf{medium}.\textsc{Medium}\\
\top &\sqsubseteq \forall \mathsf{declared\_by}.\textsc{Declaration}\\
\top &\sqsubseteq \forall \mathsf{believes}.\textsc{Belief}\\
\top &\sqsubseteq \forall \mathsf{stated\_by}.\textsc{Statement}\\
\top &\sqsubseteq \forall \mathsf{utterer}.\textsc{Agent}\\
\top &\sqsubseteq \forall \mathsf{qualifies}.\textsc{Qualified}\\
\top &\sqsubseteq \forall \mathsf{observer}.\textsc{Agent}\\
\top &\sqsubseteq \forall \mathsf{believed\_by}.\textsc{Agent}\\
\top &\sqsubseteq \forall \mathsf{observes}.\textsc{Observation}\\
\top &\sqsubseteq \forall \mathsf{addressee}.\textsc{Agent}
\end{aligned}
$$

$$\begin{aligned}
\top &\sqsubseteq \forall \mathsf{asserted\_by}.\textsc{Assertion}\\
\top &\sqsubseteq \forall \mathsf{inverse\_of\_medium}.\textsc{Expression}\\
\top &\sqsubseteq \forall \mathsf{qualified\_by}.\textsc{Qualification}
\end{aligned}$$

A.2.7 Norm

$$\begin{aligned}
\textsc{Normatively\_Qualified} &\equiv \exists \mathsf{normatively\_comparable}.\textsc{Normatively\_Qualified}\\
\textsc{Normatively\_Qualified} &\equiv \exists \mathsf{qualified\_by}.\textsc{Norm}\\
\textsc{Allowed\_And\_Disallowed} &\equiv \textsc{Disallowed} \sqcap \textsc{Allowed}\\
\textsc{Obliged} &\equiv \exists \mathsf{allowed\_by}.\textsc{Obligation}\\
\textsc{Obliged} &\equiv \exists \mathsf{normatively\_strictly\_worse}.\textsc{Disallowed}\\
\textsc{Allowed} &\equiv \exists \mathsf{allowed\_by}.\textsc{Permission}\\
\textsc{Allowed} &\equiv \exists \mathsf{normatively\_equivalent\_or\_worse}.\textsc{Normatively\_Qualified}\\
\textsc{Disallowed} &\equiv \exists \mathsf{disallowed\_by}.\textsc{Prohibition}\\
\textsc{Disallowed} &\equiv \exists \mathsf{normatively\_strictly\_better}.\textsc{Allowed}\\
\textsc{Obligation} &\equiv \textsc{Prohibition}\\
\textsc{Norm} &\equiv \exists \mathsf{qualifies}.\textsc{Normatively\_Qualified}\\
\textsc{Prohibition} &\equiv \forall \mathsf{allows}.\textsc{Obliged} \sqcap \exists \mathsf{allows}.\textsc{Obliged} \sqcap \forall \mathsf{disallows}.\textsc{Disallowed} \sqcap\\
\textsc{Observation\_of\_Violation} &\equiv \textsc{Observation} \sqcap \exists \mathsf{expresses}.\textsc{Disallowed}\\
\textsc{Belief\_In\_Violation} &\equiv \textsc{Belief} \sqcap \exists \mathsf{expresses}.\textsc{Disallowed}\\
\textsc{Permission} &\equiv \exists \mathsf{allows}.\textsc{Allowed} \sqcap \forall \mathsf{allows}.\textsc{Allowed}\\
\textsc{Disallowed\_Intention} &\equiv \textsc{Intention} \sqcap \exists \mathsf{expresses}.\textsc{Disallowed}\\
\textsc{Normatively\_Qualified} &\sqsubseteq \textsc{Qualified}\\
\textsc{Strictly\_Disallowed} &\sqsubseteq \textsc{Disallowed}\\
\textsc{Allowed\_And\_Disallowed} &\sqsubseteq \top\\
\textsc{Obliged} &\sqsubseteq \textsc{Allowed}\\
\textsc{Allowed} &\sqsubseteq \textsc{Normatively\_Qualified}\\
\textsc{Disallowed} &\sqsubseteq \textsc{Normatively\_Qualified}\\
\textsc{Strictly\_Allowed} &\sqsubseteq \textsc{Allowed}\\
\textsc{Obligation} &\sqsubseteq \textsc{Permission}\\
\textsc{Obligation} &\sqsubseteq \top\\
\textsc{Norm} &\sqsubseteq \textsc{Declaration}\\
\textsc{Norm} &\sqsubseteq \textsc{Qualification}\\
\textsc{Prohibition} &\sqsubseteq \textsc{Permission}\\
\textsc{Observation\_of\_Violation} &\sqsubseteq \textsc{Problem}\\
\textsc{Belief\_In\_Violation} &\sqsubseteq \textsc{Belief}\\
\textsc{Permission} &\sqsubseteq \textsc{Norm}\\
\textsc{Disallowed\_Intention} &\sqsubseteq \top\\
\mathsf{allowed\_by} &\equiv \mathsf{allows}^{-1}\\
\mathsf{normatively\_strictly\_better} &\equiv \mathsf{normatively\_strictly\_worse}^{-1}\\
\mathsf{normatively\_equivalent\_or\_worse} &\equiv \mathsf{normatively\_equivalent\_or\_better}^{-1}\\
\mathsf{disallowed\_by} &\equiv \mathsf{disallows}^{-1}\\
\mathsf{normatively\_strictly\_worse} &\equiv \mathsf{normatively\_strictly\_better}^{-1}\\
\mathsf{disallows} &\equiv \mathsf{disallowed\_by}^{-1}\\
\mathsf{allows} &\equiv \mathsf{allowed\_by}^{-1}\\
\mathsf{normatively\_equivalent\_or\_better} &\equiv \mathsf{normatively\_equivalent\_or\_worse}^{-1}
\end{aligned}$$

$$\begin{aligned}
\text{strictly\_equivalent} &\equiv \text{strictly\_equivalent}^{-1}\\
\text{normatively\_not\_equivalent} &\equiv \text{normatively\_not\_equivalent}^{-1}\\
\text{commands} &\equiv \text{commanded\_by}^{-1}\\
\text{commanded\_by} &\equiv \text{commands}^{-1}\\
\text{allowed\_by} &\sqsubseteq \text{qualified\_by}\\
\text{normatively\_strictly\_better} &\sqsubseteq \text{normatively\_equivalent\_or\_better}\\
\text{normatively\_strictly\_better} &\sqsubseteq \text{normatively\_not\_equivalent}\\
\text{normatively\_equivalent\_or\_worse} &\sqsubseteq \text{normatively\_comparable}\\
\text{disallowed\_by} &\sqsubseteq \text{qualified\_by}\\
\text{normatively\_strictly\_worse} &\sqsubseteq \text{normatively\_equivalent\_or\_worse}\\
\text{normatively\_strictly\_worse} &\sqsubseteq \text{normatively\_not\_equivalent}\\
\text{disallows} &\sqsubseteq \text{qualifies}\\
\text{allows} &\sqsubseteq \text{qualifies}\\
\text{normatively\_equivalent\_or\_better} &\sqsubseteq \text{normatively\_comparable}\\
\text{strictly\_equivalent} &\sqsubseteq \text{normatively\_equivalent\_or\_better}\\
\text{strictly\_equivalent} &\sqsubseteq \text{normatively\_equivalent\_or\_worse}\\
\text{normatively\_not\_equivalent} &\sqsubseteq \text{normatively\_comparable}\\
\text{commands} &\sqsubseteq \text{allows}\\
\text{commanded\_by} &\sqsubseteq \text{allowed\_by}\\
\top &\sqsubseteq \forall \text{allowed\_by}^{-1}.\textsc{Allowed}\\
\top &\sqsubseteq \forall \text{normatively\_strictly\_better}^{-1}.\textsc{Disallowed}\\
\top &\sqsubseteq \forall \text{normatively\_comparable}^{-1}.\textsc{Normatively\_Qualified}\\
\top &\sqsubseteq \forall \text{normatively\_equivalent\_or\_worse}^{-1}.\textsc{Allowed}\\
\top &\sqsubseteq \forall \text{disallowed\_by}^{-1}.\textsc{Disallowed}\\
\top &\sqsubseteq \forall \text{normatively\_strictly\_worse}^{-1}.\textsc{Obliged}\\
\top &\sqsubseteq \forall \text{strictly\_equivalent}^{-1}.\textsc{Allowed}\\
\top &\sqsubseteq \forall \text{commanded\_by}^{-1}.\textsc{Obliged}
\end{aligned}$$

$$\begin{aligned}
&\textsc{Trans}(\text{normatively\_strictly\_better})\\
&\textsc{Trans}(\text{normatively\_strictly\_worse})\\
&\textsc{Trans}(\text{normatively\_equivalent\_or\_better})\\
&\textsc{Trans}(\text{strictly\_equivalent})
\end{aligned}$$

$$\begin{aligned}
\text{strictly\_equivalent} &\equiv \text{strictly\_equivalent}^{-1}\\
\text{normatively\_not\_equivalent} &\equiv \text{normatively\_not\_equivalent}^{-1}\\
\top &\sqsubseteq \forall \text{normatively\_strictly\_better}.\textsc{Obliged}\\
\top &\sqsubseteq \forall \text{normatively\_comparable}.\textsc{Normatively\_Qualified}\\
\top &\sqsubseteq \forall \text{normatively\_strictly\_worse}.\textsc{Disallowed}\\
\top &\sqsubseteq \forall \text{disallows}.\textsc{Disallowed}\\
\top &\sqsubseteq \forall \text{allows}.\textsc{Allowed}\\
\top &\sqsubseteq \forall \text{normatively\_equivalent\_or\_better}.\textsc{Allowed}\\
\top &\sqsubseteq \forall \text{strictly\_equivalent}.\textsc{Allowed}\\
\top &\sqsubseteq \forall \text{commands}.\textsc{Obliged}
\end{aligned}$$

A.3 Final Definitions

```
Class action:Action
       equivalentClass: action:actor some (action:Agent and
                             (role:plays some role:Role) and
               (action:intends some expression:Intention) and
             (expression:believes some (expression:Belief and
                  (role:plays some expression:Expectation))))
        subClassOf: action:actor all action:Agent
        subClassOf: action:actor exactly 1
        subClassOf: process:Process

Class action:Agent
        subClassOf: action:actor_in all action:Action
        subClassOf: expression:holds all top:Mental_Concept
        subClassOf: owl:Thing
        subClassOf: role:plays all role:Role
        subClassOf: process:participant_in all process:Process

Class action:Artifact
        equivalentClass: (role:plays some role:Function) and
                    (process:result_of some action:Creation)
        subClassOf: process:Physical_Object
        subClassOf: role:plays all role:Role

Class action:Collaborative_Plan
        subClassOf: action:Plan

Class action:Continuation
        equivalentClass: process:requirement some action:Initiation
        equivalentClass: process:result some action:Termination
        subClassOf: mereo:direct_part_of some action:Action
        subClassOf: action:Action

Class action:Creation
        equivalentClass: process:creation some owl:Thing
        subClassOf: action:Action
        subClassOf: process:creation all owl:Thing

Class action:Initiation
        equivalentClass: process:result some action:Continuation
        subClassOf: action:Action
        subClassOf: mereo:direct_part_of some action:Action

Class action:Natural_Object
        subClassOf: process:Physical_Object

Class action:Organisation
        equivalentClass: mereo:member some action:Person
        subClassOf: mereo:member all ((action:Organisation or
                                         action:Person) and
                    (role:plays some role:Organisation_Role))
        subClassOf: action:Agent
```

```
Class action:Person
       equivalentClass: laction:Natural_Person
       subClassOf: action:Agent
       subClassOf: action:Natural_Object

Class action:Personal_Plan
       subClassOf: action:Plan

Class action:Plan
       equivalentClass: mereo:part some action:Action
       subClassOf: top:Mental_Object
       subClassOf: mereo:part all (action:Action or action:Plan)

Class action:Reaction
       equivalentClass: action:actor some (action:Agent and
           (expression:observes some (expression:Belief and
                (expression:qualifies some (role:plays some
                                (expression:Observation and
                  (role:played_by some action:Action))))))
       subClassOf: action:Action

Class action:Speech_Act
       equivalentClass: process:creation some
                          expression:Communicated_Attitude
       subClassOf: action:Creation

Class action:Termination
       equivalentClass: process:requirement some
                                       action:Continuation
       subClassOf: mereo:direct_part_of some action:Action
       subClassOf: action:Action

Class action:Trade
       subClassOf: action:Transaction

Class action:Transaction
       equivalentClass: mereo:direct_part some action:Action
       subClassOf: action:Collaborative_Plan
       subClassOf: mereo:direct_part exactly 2

Class expression:Argument
       equivalentClass: role:played_by some (expression:Expression
                 and (expression:attitude some expression:Belief))
       subClassOf: expression:Reason

Class expression:Assertion
       equivalentClass: expression:asserts some
                          expression:Expression
       subClassOf: expression:Communicated_Attitude
```

```
Class expression:Assumption
       equivalentClass: role:played_by some
                                       (expression:Proposition and
                     (expression:attitude some expression:Belief))
       subClassOf: role:Epistemic_Role

Class expression:Belief
       equivalentClass: expression:believed_by some action:Agent
       subClassOf: expression:Propositional_Attitude

Class expression:Cause
       subClassOf: role:Epistemic_Role

Class expression:Communicated_Attitude
       equivalentClass: (expression:utterer some action:Agent) and
                (expression:states some expression:Expression) and
                          (expression:addressee some action:Agent)
       subClassOf: expression:states all expression:Expression
       subClassOf: expression:addressee all action:Agent
       subClassOf: expression:Propositional_Attitude

Class expression:Declaration
       equivalentClass: expression:declares some
                              expression:Expression
       subClassOf: expression:Communicated_Attitude

Class expression:Desire
       subClassOf: expression:Propositional_Attitude

Class expression:Document
       equivalentClass: expression:bears some
             (expression:Expression and (expression:stated_by some
                                 expression:Statement_In_Writing))
       subClassOf: expression:bears all (expression:Expression and
      (expression:stated_by some expression:Statement_In_Writing))
       subClassOf: expression:Medium

Class expression:Evaluative_Attitude
       equivalentClass: expression:qualifies some
                           expression:Proposition
       subClassOf: expression:Propositional_Attitude

Class expression:Evaluative_Proposition
       equivalentClass: expression:qualified_by some
                      expression:Evaluative_Attitude
       equivalentClass: expression:evaluatively_comparable some
                              expression:Evaluative_Proposition
       subClassOf: expression:Proposition
```

```
Class expression:Evidence
       equivalentClass: role:played_by some
             (expression:Proposition and (expression:attitude some
                                               expression:Belief) and
                         (role:plays some expression:Observation))
       subClassOf: role:Epistemic_Role

Class expression:Exception
       equivalentClass: role:played_by some expression:Proposition
       subClassOf: role:Epistemic_Role

Class expression:Expectation
       equivalentClass: role:played_by some
             (expression:Proposition and (expression:attitude some
                                               expression:Belief) and
                      (role:plays all not expression:Observation))
       subClassOf: role:Epistemic_Role

Class expression:Expression
       equivalentClass: (expression:stated_by some
                             expression:Communicated_Attitude) and
                        (expression:medium some expression:Medium)
       subClassOf: expression:medium all expression:Medium
       subClassOf: expression:Proposition

Class expression:Fact
       equivalentClass: role:played_by some
             (expression:Proposition and (expression:attitude some
                                               expression:Belief) and
                         (role:plays some expression:Observation))
       subClassOf: role:Epistemic_Role

Class expression:Intention
       equivalentClass: action:intended_by some action:Agent
       subClassOf: expression:Propositional_Attitude

Class expression:Lie
       subClassOf: expression:Assertion

Class expression:Medium
       equivalentClass: expression:bears some expression:Expression
       subClassOf: expression:bears all expression:Expression
       subClassOf: owl:Thing

Class expression:Observation
       equivalentClass: role:played_by some (expression:attitude
                      some (expression:observer some action:Agent))
       subClassOf: role:Epistemic_Role
```

```
Class expression:Problem
       subClassOf: expression:Observation

Class expression:Promise
       equivalentClass: expression:promises some
                              expression:Expression
       subClassOf: expression:Communicated_Attitude

Class expression:Proposition
       equivalentClass: (expression:medium some expression:Medium)
                        and (expression:held_by some action:Agent)
       subClassOf: expression:attitude all
                                 expression:Propositional_Attitude
       subClassOf: top:Mental_Object

Class expression:Propositional_Attitude
       equivalentClass: (expression:towards some
                                     expression:Proposition) and
                          (expression:held_by some action:Agent)
       subClassOf: expression:towards all expression:Proposition
       subClassOf: top:Mental_Object

Class expression:Qualification
       equivalentClass: expression:qualifies some
                             expression:Qualified
       subClassOf: top:Mental_Object
       subClassOf: expression:qualifies all expression:Qualified

Class expression:Qualified
       subClassOf: owl:Thing

Class expression:Reason
       equivalentClass: role:played_by some
                     (expression:Proposition and
                     (expression:attitude some expression:Belief))
       subClassOf: role:Epistemic_Role

Class expression:Statement_In_Writing
       equivalentClass: expression:Communicated_Attitude and
                   (expression:author some action:Agent) and
          (expression:states some (expression:Expression and
              (expression:medium some expression:Document)))
       subClassOf: expression:Communicated_Attitude

Class expression:Surprise
       subClassOf: expression:Observation
```

```
Class laction:Act_of_Law
       equivalentClass: action:actor some laction:Legislative_Body
       subClassOf: action:actor all laction:Legislative_Body
       subClassOf: laction:Legal_Speech_Act

Class laction:Assignment
       equivalentClass: action:actor some laction:Public_Body
       equivalentClass: mereo:part_of some (action:Transaction and
                               (mereo:part some (action:Action and
                        (action:actor some laction:Public_Body))))
       subClassOf: action:actor all laction:Public_Body
       subClassOf: laction:Legal_Speech_Act
       subClassOf: laction:Public_Act

Class laction:Association
       equivalentClass: laction:Society
       subClassOf: laction:Legal Person (Private Law)

Class laction:Co-operative
       subClassOf: laction:Society

Class laction:Company
       equivalentClass: laction:Limited_Company or
                                    laction:Public_Limited_Company
       subClassOf: laction:Legal Person (Private Law)

Class laction:Corporation
       equivalentClass: laction:Unincorporated or
                        laction:Foundation or laction:Incorporated
       subClassOf: laction:Legal Person (Private Law)

Class laction:Decision
       equivalentClass: process:creation some
                              (expression:Statement_In_Writing and
                     (expression:towards some laction:Public_Act))
       equivalentClass: action:actor some laction:Public_Body
       equivalentClass: process:creation some (expression:Promise
                and (expression:towards some laction:Public_Act))
       subClassOf: action:actor all laction:Public_Body
       subClassOf: laction:Legal_Speech_Act

Class laction:Delegation
       equivalentClass: mereo:part_of some action:Transaction
       equivalentClass: action:actor some laction:Public_Body
       subClassOf: laction:Legal_Speech_Act
       subClassOf: action:actor all laction:Public_Body
       subClassOf: laction:Public_Act
```

```
Class laction:Foundation
       subClassOf: laction:Corporation

Class laction:Incorporated
       equivalentClass: laction:PLC
       subClassOf: laction:Corporation

Class laction:Legal Person (Private Law)
       equivalentClass: process:created_by some laction:Public_Act
       subClassOf: laction:Legal_Person

Class laction:Legal_Person
       equivalentClass: process:created_by some laction:Public_Act
       subClassOf: action:Organisation

Class laction:Legal_Speech_Act
       subClassOf: action:Speech_Act

Class laction:Legislative Body
       subClassOf: laction:Public_Body

Class laction:Limited_Company
       equivalentClass: laction:Unincorporated
       subClassOf: laction:Company

Class laction:Mandate
       equivalentClass: mereo:part_of some action:Transaction
       equivalentClass: action:actor some laction:Public_Body
       subClassOf: action:actor all laction:Public_Body
       subClassOf: laction:Public_Act

Class laction:Natural_Person
       equivalentClass: action:Person
       subClassOf: action:Person

Class laction:PLC
       equivalentClass: laction:Incorporated
       subClassOf: laction:Company

Class laction:Public_Act
       equivalentClass: (process:result some
                             expression:Communicated_Attitude) and
        (action:actor some (action:Person or action:Organisation))
       subClassOf: action:Action

Class laction:Public_Body
       equivalentClass: process:created_by some laction:Act_of_Law
       subClassOf: laction:Legal_Person
```

```
Class laction:Society
       equivalentClass: laction:Association
       subClassOf: laction:Legal Person (Private Law)

Class laction:Unincorporated
       equivalentClass: laction:Limited_Company
       subClassOf: laction:Corporation

Class lrole:Legal_Role
       subClassOf: role:Role

Class lrole:Professional_Legal_Role
       subClassOf: role:Organisation_Role
       subClassOf: lrole:Social_Legal_Role

Class lrole:Social_Legal_Role
       subClassOf: lrole:Legal_Role
       subClassOf: role:Social_Role

Class mereo:Atom
       subClassOf: top:Abstract_Concept

Class mereo:Composition
       equivalentClass: mereo:part some owl:Thing
       subClassOf: top:Abstract_Concept

Class mereo:Pair
       subClassOf: mereo:part exactly 2
       subClassOf: mereo:Composition

Class mereo:Part
       equivalentClass: mereo:part_of some mereo:Whole
       subClassOf: mereo:part_of all mereo:Whole
       subClassOf: top:Abstract_Concept

Class mereo:Whole
       equivalentClass: mereo:part some mereo:Part
       subClassOf: mereo:part all mereo:Part
       subClassOf: top:Abstract_Concept

Class modification:Annulment
       subClassOf: modification:In_Force_Modification

Class modification:Application
       subClassOf: modification:Modification_of_System

Class modification:Application_Date
       subClassOf: time:Moment
       subClassOf: modification:Dynamic_Temporal_Entity
```

```
Class modification:Application_Interval
       subClassOf: modification:Dynamic_Temporal_Entity
       subClassOf: time:Interval

Class modification:Delivery_Date
       subClassOf: time:Moment
       subClassOf: modification:Static_Temporal_Entity

Class modification:Deregulation
       subClassOf: modification:Modification_of_System

Class modification:Dynamic_Temporal_Entity
       subClassOf: time:Temporal_Occurrence

Class modification:Efficacy_Interval
       subClassOf: modification:Dynamic_Temporal_Entity
       subClassOf: time:Interval

Class modification:Efficacy_Modification
       subClassOf: modification:Temporal_Modification

Class modification:End_efficacy
       subClassOf: modification:Efficacy_Modification

Class modification:End_in_Force
       subClassOf: modification:In_Force_Modification

Class modification:Enter_in_Force_Date
       subClassOf: time:Moment
       subClassOf: modification:Static_Temporal_Entity

Class modification:Exception
       subClassOf: modification:Modification_of_Scope

Class modification:Existence_Date
       subClassOf: time:Moment

Class modification:Extension
       subClassOf: modification:Modification_of_Scope

Class modification:In_Force_Interval
       subClassOf: modification:Dynamic_Temporal_Entity
       subClassOf: time:Interval

Class modification:In_Force_Modification
       subClassOf: modification:Temporal_Modification

Class modification:Integration
       subClassOf: modification:Textual_Modification
```

```
Class modification:Interpretation
       subClassOf: modification:Modification_of_Meaning

Class modification:Modification
       subClassOf: laction:Public_Act

Class modification:Modification_of_Meaning
       subClassOf: modification:Semantic_Annotation

Class modification:Modification_of_Scope
       subClassOf: modification:Semantic_Annotation

Class modification:Modification_of_System
       subClassOf: modification:Semantic_Annotation

Class modification:Modification_of_Term
       subClassOf: modification:Modification_of_Meaning

Class modification:Prorogation_Efficacy
       subClassOf: modification:Efficacy_Modification

Class modification:Prorogation_in_Force
       subClassOf: modification:In_Force_Modification

Class modification:Publication_Date
       subClassOf: time:Moment
       subClassOf: modification:Static_Temporal_Entity

Class modification:Ratification
       subClassOf: modification:Modification_of_System

Class modification:Relocation
       subClassOf: modification:Textual_Modification

Class modification:Remaking
       subClassOf: modification:Modification_of_System

Class modification:Renewal
       subClassOf: modification:In_Force_Modification

Class modification:Repeal
       subClassOf: modification:Textual_Modification

Class modification:Retroactivity
       subClassOf: modification:Efficacy_Modification

Class modification:Semantic_Annotation
       subClassOf: modification:Modification
```

```
Class modification:Start_Efficacy
       subClassOf: modification:Efficacy_Modification

Class modification:Start_in_Force
       subClassOf: modification:In_Force_Modification

Class modification:Static_Temporal_Entity
       subClassOf: time:Temporal_Occurrence

Class modification:Substitution
       subClassOf: modification:Textual_Modification

Class modification:Suspension
       subClassOf: modification:Efficacy_Modification

Class modification:Temporal_Modification
       subClassOf: modification:Modification

Class modification:Textual_Modification
       subClassOf: modification:Modification

Class modification:Transposition
       subClassOf: modification:Modification_of_System

Class modification:Ultractivity
       subClassOf: modification:Efficacy_Modification

Class modification:Variation
       subClassOf: modification:Modification_of_Meaning

Class norm:Action_Power
       subClassOf: norm:Hohfeldian_Power

Class norm:Allowed
       equivalentClass: norm:allowed_by some norm:Permission
       equivalentClass: norm:normatively_equivalent_or_worse some
                                      norm:Normatively_Qualified
       subClassOf: norm:Normatively_Qualified

Class norm:Allowed_And_Disallowed
       equivalentClass: norm:Disallowed and norm:Allowed
       subClassOf: norm:Allowed
       subClassOf: norm:Disallowed

Class norm:Belief_In_Violation
       equivalentClass: expression:Belief and
                          (expression:towards some norm:Disallowed)
       subClassOf: expression:Belief
```

```
Class norm:Code
       equivalentClass: expression:bears some (norm:Norm and
               (expression:utterer some laction:Legislative_Body))
       subClassOf: expression:bears all (expression:utterer some
                                         laction:Legislative_Body)
       subClassOf: norm:Legal_Document

Class norm:Code_of_Conduct
       subClassOf: norm:Soft_Law
       subClassOf: norm:Legal_Document

Class norm:Contract
       equivalentClass: expression:bears some (norm:Norm and
               (expression:utterer some (laction:Natural_Person or
                                           laction:Legal_Person)))
       subClassOf: expression:bears all (expression:utterer some
                 (laction:Natural_Person or laction:Legal_Person))
       subClassOf: norm:Legal_Document

Class norm:Custom
       subClassOf: expression:Medium

Class norm:Customary_Law
       subClassOf: norm:Custom
       subClassOf: norm:Legal_Source

Class norm:Declarative_Power
       subClassOf: norm:Potestative_Expression

Class norm:Decree
       subClassOf: norm:Legal_Document
       subClassOf: norm:Proclamation

Class norm:Definitional_Expression
       subClassOf: norm:Legal_Expression

Class norm:Directive
       subClassOf: norm:Legal_Document
       subClassOf: norm:Proclamation

Class norm:Disallowed
       equivalentClass: norm:normatively_strictly_better some
                                                    norm:Allowed
       equivalentClass: norm:disallowed_by some norm:Prohibition
       subClassOf: norm:Normatively_Qualified

Class norm:Disallowed_Intention
       equivalentClass: expression:Intention and
                      (expression:towards some norm:Disallowed)
       subClassOf: expression:Intention
```

```
Class norm:Enabling_Power
       subClassOf: norm:Potestative_Expression

Class norm:Evaluative_Expression
       subClassOf: norm:Legal_Expression
       subClassOf: expression:Evaluative_Proposition

Class norm:Exclusionary_Right
       subClassOf: norm:Obligative_Right

Class norm:Existential_Expression
       subClassOf: norm:Legal_Expression

Class norm:Hohfeldian_Power
       subClassOf: norm:Potestative_Expression

Class norm:Immunity
       subClassOf: norm:Hohfeldian_Power

Class norm:International_Agreement
       subClassOf: norm:Legal_Source

Class norm:International_Arbitration
       subClassOf: norm:Soft_Law

Class norm:Legal_Doctrine
       subClassOf: norm:Legal_Source

Class norm:Legal_Document
       subClassOf: expression:Document
       subClassOf: norm:Legal_Source

Class norm:Legal_Expression
       equivalentClass: expression:attitude some
                              expression:Communicated_Attitude
       subClassOf: expression:Expression

Class norm:Legal_Source
       equivalentClass: expression:bears some (norm:Norm or
                                        norm:Legal_Expression)
       subClassOf: expression:Medium

Class norm:Liability_Right
       subClassOf: norm:Right

Class norm:Liberty_Right
       subClassOf: norm:Right
```

```
Class norm:Mandatory_Precedent
       subClassOf: norm:Precedent

Class norm:Non-binding_International_Agreement
       subClassOf: norm:Soft_Law
       subClassOf: norm:International_Agreement

Class norm:Norm
       equivalentClass: expression:qualifies some
                                    norm:Normatively_Qualified
       subClassOf: expression:Qualification

Class norm:Normatively_Qualified
       equivalentClass: norm:normatively_comparable some
                                    norm:Normatively_Qualified
       equivalentClass: expression:qualified_by some norm:Norm
       subClassOf: expression:Qualified

Class norm:Obligation
       equivalentClass: norm:Prohibition
       subClassOf: norm:Permission

Class norm:Obligative_Right
       subClassOf: norm:Right

Class norm:Obliged
       equivalentClass: norm:normatively_strictly_worse some
                                             norm:Disallowed
       equivalentClass: norm:allowed_by some norm:Obligation
       subClassOf: norm:Allowed

Class norm:Observation_of_Violation
       equivalentClass: expression:Observation and
                       (role:played_by some norm:Disallowed)
       subClassOf: expression:Problem

Class norm:Permission
       equivalentClass: (norm:allows some norm:Allowed) and
                              (norm:allows all norm:Allowed)
       subClassOf: norm:Norm

Class norm:Permissive_Right
       subClassOf: norm:Right

Class norm:Persuasive_Precedent
       subClassOf: norm:Precedent

Class norm:Potestative_Expression
       subClassOf: norm:Legal_Expression
```

```
Class norm:Potestative_Right
       subClassOf: norm:Enabling_Power

Class norm:Precedent
       subClassOf: norm:Legal_Source

Class norm:Proclamation
       subClassOf: norm:Legal_Source

Class norm:Prohibition
       equivalentClass: norm:Obligation
       equivalentClass: (norm:allows all norm:Obliged) and
                       (norm:allows some norm:Obliged) and
                  (norm:disallows all norm:Disallowed) and
                     (norm:disallows some norm:Disallowed)
       subClassOf: norm:Permission

Class norm:Qualificatory_Expression
       subClassOf: norm:Legal_Expression

Class norm:Regulation
       equivalentClass: expression:bears some (norm:Norm and
             (expression:utterer some laction:Legislative_Body))
       subClassOf: expression:bears all (expression:utterer some
                                       laction:Legislative_Body)
       subClassOf: norm:Legal_Document

Class norm:Resolution
       subClassOf: norm:Soft_Law
       subClassOf: norm:International_Agreement

Class norm:Right
       subClassOf: norm:Norm

Class norm:Soft_Law
       subClassOf: norm:Legal_Source

Class norm:Statute
       equivalentClass: expression:bears some (norm:Norm and
                 (expression:utterer some laction:Legal_Person))
       subClassOf: expression:bears all (expression:utterer some
                                           laction:Legal_Person)
       subClassOf: norm:Legal_Document

Class norm:Strictly_Allowed
       subClassOf: norm:Allowed

Class norm:Strictly_Disallowed
       subClassOf: norm:Disallowed
```

```
Class norm:Treaty
       subClassOf: norm:Legal_Document
       subClassOf: norm:International_Agreement

Class place:Absolute_Place
       subClassOf: place:Place

Class place:Comprehensive_Place
       subClassOf: place:Location_Complex

Class place:Location_Complex
       equivalentClass: place:location_complex_for some place:Place
       subClassOf: place:Place

Class place:Place
       equivalentClass: place:Absolute_Place or place:Relative_Place
       subClassOf: top:Spatio_Temporal_Occurrence
       subClassOf: place:location_complex some place:Location_Complex

Class place:Relative_Place
       equivalentClass: place:spatial_reference some owl:Thing
       subClassOf: place:Place

Class process:Change
       equivalentClass: (process:result some owl:Thing) and
                              (process:requirement some owl:Thing)
       subClassOf: mereo:part all process:Change
       subClassOf: owl:Thing

Class process:Physical_Object
       subClassOf: top:Physical_Concept

Class process:Process
       equivalentClass: process:resource some
                                     time:Temporal_Occurrence
       subClassOf: process:Change

Class role:Epistemic_Role
       equivalentClass: role:played_by some top:Mental_Object
       subClassOf: role:Role
       subClassOf: role:played_by all top:Mental_Object

Class role:Function
       equivalentClass: role:played_by some
                    (process:Physical_Object and not action:Agent)
       subClassOf: role:played_by all (process:Physical_Object and
                                                  not action:Agent)
       subClassOf: role:Role
```

```
Class role:Organisation_Role
       equivalentClass: role:played_by some (action:Agent and
                          (mereo:part_of some action:Organisation))
       subClassOf: role:Social_Role
       subClassOf: role:played_by all (action:Agent and
                          (mereo:part_of some action:Organisation))

Class role:Person_Role
       equivalentClass: role:played_by some action:Person
       subClassOf: role:played_by all action:Person
       subClassOf: role:Social_Role

Class role:Role
       equivalentClass: role:played_by some owl:Thing
       subClassOf: top:Mental_Concept

Class role:Social_Role
       equivalentClass: role:played_by some (action:Agent and
                      (action:actor_in some (action:Action and
                      (role:plays some expression:Expectation))))
       subClassOf: role:played_by all action:Agent
       subClassOf: role:Role

Class rules:Argument
       subClassOf: rules:Rule
       subClassOf: expression:Argument

Class rules:Assumption
       subClassOf: expression:Assumption
       subClassOf: rules:Atom

Class rules:Atom
       equivalentClass: role:played_by some
                             (expression:Expression and mereo:Atom)
       subClassOf: role:Epistemic_Role
       subClassOf: role:played_by all mereo:Atom

Class rules:Exception
       subClassOf: expression:Exception
       subClassOf: rules:Atom

Class rules:Negated_Atom
       equivalentClass: role:played_by some (expression:Expression
                                                    and mereo:Atom)
       subClassOf: role:Epistemic_Role
       subClassOf: role:played_by all mereo:Atom
```

```
Class rules:Rule
       equivalentClass: role:played_by some (expression:Expression
                                             and mereo:Composition)
       subClassOf: role:Epistemic_Role
       subClassOf: role:played_by all mereo:Atom

Class rules:Valid_Rule
       equivalentClass: role:played_by some (expression:Expression
                                             and mereo:Composition)
       subClassOf: rules:Rule
       subClassOf: role:played_by all mereo:Atom

Class time:Interval
       subClassOf: time:Temporal_Occurrence
       subClassOf: mereo:Composition

Class time:Moment
       subClassOf: mereo:Atom
       subClassOf: time:Temporal_Occurrence

Class time:Pair_Of_Periods
       subClassOf: mereo:component all time:Temporal_Occurrence
       subClassOf: mereo:Pair

Class time:Temporal_Occurrence
       equivalentClass: (time:immediatly_after some
                                      time:Temporal_Occurrence) and
            (time:immediatly_before some time:Temporal_Occurrence)
       subClassOf: top:Spatio_Temporal_Occurrence

Class top:Abstract_Concept
       subClassOf: owl:Thing

Class top:Mental_Concept
       subClassOf: owl:Thing

Class top:Mental_Object
       subClassOf: top:Mental_Concept
       subClassOf: expression:held_by all action:Agent

Class top:Occurrence
       subClassOf: owl:Thing

Class top:Physical_Concept
       subClassOf: owl:Thing

Class top:Spatio_Temporal_Occurrence
       subClassOf: top:Occurrence
```

```
Property action:actor
       subPropertyOf: process:participant
       domain: action:Action
       range: action:Agent
       inverseOf: action:actor_in

Property action:actor_in
       subPropertyOf: process:participant_in
       domain: action:Agent
       range: action:Action
       inverseOf: action:actor

Property action:intended_by
       subPropertyOf: expression:held_by
       domain: expression:Intention
       range: action:Agent
       inverseOf: action:intends

Property action:intends
       subPropertyOf: expression:holds
       domain: action:Agent
       range: expression:Intention
       inverseOf: action:intended_by

Property expression:addressee
       domain: expression:Communicated_Attitude
       range: action:Agent

Property expression:asserted_by
       subPropertyOf: expression:stated_by
       domain: expression:Expression
       range: expression:Assertion
       inverseOf: expression:asserts

Property expression:asserts
       subPropertyOf: expression:states
       domain: expression:Assertion
       range: expression:Expression
       inverseOf: expression:asserted_by

Property expression:attitude
       domain: expression:Qualified
       range: expression:Propositional_Attitude
       range: expression:Qualification
       inverseOf: expression:towards

Property expression:author
       subPropertyOf: expression:utterer
```

```
Property expression:bears
       domain: expression:Medium
       range: expression:Expression
       inverseOf: expression:medium

Property expression:believed_by
       subPropertyOf: expression:held_by
       domain: expression:Belief
       range: action:Agent
       inverseOf: expression:believes

Property expression:believes
       subPropertyOf: expression:holds
       domain: action:Agent
       range: expression:Belief
       inverseOf: expression:believed_by

Property expression:declared_by
       subPropertyOf: expression:stated_by
       domain: expression:Expression
       range: expression:Declaration
       inverseOf: expression:declares

Property expression:declares
       subPropertyOf: expression:states
       domain: expression:Declaration
       range: expression:Expression
       inverseOf: expression:declared_by

Property expression:evaluatively_comparable
       domain: expression:Evaluative_Proposition
       domain: expression:Qualified
       range: expression:Evaluative_Proposition
       range: expression:Qualified

Property expression:held_by
       domain: top:Mental_Object
       range: action:Agent
       inverseOf: expression:holds

Property expression:holds
       domain: action:Agent
       range: top:Mental_Object
       inverseOf: expression:held_by

Property expression:medium
       domain: expression:Expression
       range: expression:Medium
       inverseOf: expression:bears
```

```
Property expression:observer
       subPropertyOf: expression:believed_by
       domain: expression:Belief and (role:plays some
                               expression:Observation)
       range: action:Agent
       inverseOf: expression:observes

Property expression:observes
       subPropertyOf: expression:believes
       domain: action:Agent
       range: expression:Belief and (role:plays some
                              expression:Observation)
       inverseOf: expression:observer

Property expression:promised_by
       subPropertyOf: expression:stated_by
       domain: expression:Expression
       range: expression:Promise
       inverseOf: expression:promises

Property expression:promises
       subPropertyOf: expression:states
       domain: expression:Promise
       range: expression:Expression
       inverseOf: expression:promised_by

Property expression:qualified_by
       subPropertyOf: expression:attitude
       domain: expression:Qualified
       domain: expression:Evaluative_Proposition
       range: expression:Qualification
       range: expression:Evaluative_Attitude
       inverseOf: expression:qualifies

Property expression:qualifies
       subPropertyOf: expression:towards
       domain: expression:Qualification
       domain: expression:Evaluative_Attitude
       range: expression:Qualified
       range: expression:Evaluative_Proposition
       inverseOf: expression:qualified_by

Property expression:stated_by
       subPropertyOf: expression:attitude
       range: expression:Communicated_Attitude
       inverseOf: expression:states
```

```
Property expression:states
       subPropertyOf: expression:towards
       domain: expression:Communicated_Attitude
       inverseOf: expression:stated_by

Property expression:towards
       domain: expression:Qualification
       domain: expression:Propositional_Attitude
       range: expression:Qualified
       inverseOf: expression:attitude

Property expression:utterer
       subPropertyOf: expression:held_by
       domain: expression:Communicated_Attitude
       range: action:Agent
       inverseOf: expression:utters

Property expression:utters
       subPropertyOf: expression:holds
       domain: action:Agent
       range: expression:Communicated_Attitude
       inverseOf: expression:utterer

Property mereo:component
       subPropertyOf: mereo:part
       inverseOf: mereo:component_of

Property mereo:component_of
       subPropertyOf: mereo:part_of
       inverseOf: mereo:component

Property mereo:composed_of
       subPropertyOf: mereo:part

Property mereo:contained_in
       subPropertyOf: mereo:part_of

Property mereo:direct_part
       subPropertyOf: mereo:part
       inverseOf: mereo:direct_part_of

Property mereo:direct_part_of
       subPropertyOf: mereo:part_of
       inverseOf: mereo:direct_part

Property mereo:member
       subPropertyOf: mereo:part
       inverseOf: mereo:member_of
```

```
Property mereo:member_of
       subPropertyOf: mereo:part_of
       inverseOf: mereo:member

Property mereo:part
       inverseOf: mereo:part_of

Property mereo:part_of
       inverseOf: mereo:part

Property modification:application
       domain: modification:Modification
       range: modification:Application_Date

Property modification:date
       domain: time:Moment
       range: xsd:dateTime

Property modification:duration
       domain: modification:Suspension
       range: time:Interval

Property modification:efficacy
       domain: modification:Modification
       range: modification:Efficacy_Interval

Property modification:final_date
       subPropertyOf: time:temporal_relation
       domain: time:Interval
       range: time:Moment
       inverseOf: modification:final_date_of

Property modification:final_date_of
       subPropertyOf: time:finishes
       domain: time:Moment
       range: time:Interval
       inverseOf: modification:final_date

Property modification:in_force
       domain: modification:Modification
       range: modification:In_Force_Interval

Property modification:initial_date
       subPropertyOf: time:temporal_relation
       domain: time:Interval
       range: time:Moment
       inverseOf: modification:initial_date_of
```

```
Property modification:initial_date_of
       subPropertyOf: time:starts
       domain: time:Moment
       range: time:Interval
       inverseOf: modification:initial_date

Property modification:produce_efficacy_modification
       domain: modification:Repeal or modification:Integration or
                                         modification:Substitution

Property modification:produce_inforce_modification
       domain: modification:Repeal or modification:Integration or
                                         modification:Substitution
       range: modification:Renewal or modification:Annulment

Property modification:produce_textual_modification
       domain: modification:Annulment or modification:Renewal
       range: modification:End_in_Force

Property norm:allowed_by
       subPropertyOf: expression:qualified_by
       domain: norm:Allowed
       inverseOf: norm:allows

Property norm:allows
       subPropertyOf: expression:qualifies
       range: norm:Allowed
       inverseOf: norm:allowed_by

Property norm:commanded_by
       subPropertyOf: norm:allowed_by
       domain: norm:Obliged
       inverseOf: norm:commands

Property norm:commands
       subPropertyOf: norm:allows
       range: norm:Obliged
       inverseOf: norm:commanded_by

Property norm:disallowed_by
       subPropertyOf: expression:qualified_by
       domain: norm:Disallowed
       inverseOf: norm:disallows

Property norm:disallows
       subPropertyOf: expression:qualifies
       range: norm:Disallowed
       inverseOf: norm:disallowed_by
```

```
Property norm:normatively_comparable
       subPropertyOf: expression:evaluatively_comparable
       domain: norm:Normatively_Qualified
       range: norm:Normatively_Qualified

Property norm:normatively_equivalent_or_better
       subPropertyOf: norm:normatively_comparable
       range: norm:Allowed
       inverseOf: norm:normatively_equivalent_or_worse

Property norm:normatively_equivalent_or_worse
       subPropertyOf: norm:normatively_comparable
       domain: norm:Allowed
       inverseOf: norm:normatively_equivalent_or_better

Property norm:normatively_not_equivalent
       subPropertyOf: norm:normatively_comparable
       inverseOf: norm:normatively_not_equivalent

Property norm:normatively_strictly_better
       subPropertyOf: norm:normatively_not_equivalent
       subPropertyOf: norm:normatively_equivalent_or_better
       domain: norm:Disallowed
       range: norm:Obliged
       inverseOf: norm:normatively_strictly_worse

Property norm:normatively_strictly_worse
       subPropertyOf: norm:normatively_not_equivalent
       subPropertyOf: norm:normatively_equivalent_or_worse
       domain: norm:Obliged
       range: norm:Disallowed
       inverseOf: norm:normatively_strictly_better

Property norm:strictly_equivalent
       subPropertyOf: norm:normatively_equivalent_or_worse
       subPropertyOf: norm:normatively_equivalent_or_better
       domain: norm:Allowed
       range: norm:Allowed
       inverseOf: norm:strictly_equivalent

Property place:abut
       subPropertyOf: place:meet
       domain: place:Place
       range: place:Place
       inverseOf: place:abut
```

```
Property place:connect
       subPropertyOf: place:meet
       subPropertyOf: place:relatively_fixed
       domain: place:Place
       range: place:Place

Property place:cover
       subPropertyOf: place:meet
       domain: place:Place
       range: place:Place
       inverseOf: place:covered_by

Property place:covered_by
       subPropertyOf: place:partially_coincide
       subPropertyOf: place:meet
       domain: place:Place
       range: place:Place
       inverseOf: place:cover

Property place:exactly_coincide
       subPropertyOf: place:meet
       domain: place:Place
       range: place:Place
       inverseOf: place:exactly_coincide

Property place:externally_connect
       subPropertyOf: place:abut
       subPropertyOf: place:connect
       domain: place:Place
       range: place:Place

Property place:in
       subPropertyOf: mereo:part_of
       subPropertyOf: place:spatial_relation
       domain: place:Place
       range: place:Place

Property place:location_complex
       subPropertyOf: place:spatial_relation
       domain: place:Place
       range: place:Location_Complex
       inverseOf: place:location_complex_for

Property place:location_complex_for
       subPropertyOf: place:spatial_relation
       domain: place:Location_Complex
       range: place:Place
       inverseOf: place:location_complex
```

```
Property place:meet
       subPropertyOf: place:spatial_relation
       domain: place:Place
       range: place:Place
       inverseOf: place:meet

Property place:overlap
       subPropertyOf: place:partially_coincide
       subPropertyOf: place:relatively_fixed
       domain: place:Place
       range: place:Place

Property place:partially_coincide
       subPropertyOf: place:meet
       domain: place:Place
       range: place:Place
       inverseOf: place:partially_coincide

Property place:relatively_fixed
       subPropertyOf: place:spatial_relation
       domain: place:Place
       range: place:Place
       inverseOf: place:relatively_fixed

Property place:spatial_reference
       subPropertyOf: place:spatial_relation
       domain: place:Place
       range: owl:Thing

Property place:spatial_relation
       domain: place:Place
       range: place:Place

Property process:created_by
       subPropertyOf: process:result_of
       inverseOf: process:creation

Property process:creation
       subPropertyOf: process:result
       inverseOf: process:created_by

Property process:participant
       domain: owl:Thing
       range: process:Change
       inverseOf: process:participant_in
```

```
Property process:participant_in
       domain: process:Change
       range: owl:Thing
       inverseOf: process:participant

Property process:requirement
       subPropertyOf: process:participant
       inverseOf: process:requirement_of

Property process:requirement_of
       subPropertyOf: process:participant_in
       inverseOf: process:requirement

Property process:resource
       subPropertyOf: process:participant
       inverseOf: process:resource_for

Property process:resource_for
       subPropertyOf: process:participant_in
       inverseOf: process:resource

Property process:result
       subPropertyOf: process:participant
       inverseOf: process:result_of

Property process:result_of
       subPropertyOf: process:participant_in
       inverseOf: process:result

Property role:played_by
       domain: role:Role
       range: owl:Thing
       inverseOf: role:plays

Property role:plays
       domain: owl:Thing
       range: role:Role
       inverseOf: role:played_by

Property rules:applies
       subPropertyOf: rules:rule_predicate
       range: role:plays some rules:Atom

Property rules:excluded
       subPropertyOf: rules:rule_predicate
       range: role:plays some rules:Atom
```

```
Property rules:prior
       subPropertyOf: rules:rule_predicate
       range: role:plays some rules:Rule

Property rules:rebuts
       subPropertyOf: rules:rule_predicate
       range: role:plays some rules:Rule

Property rules:rule_predicate
       domain: role:plays some rules:Rule
       range: role:plays some (rules:Atom or rules:Rule)

Property time:after
       subPropertyOf: time:temporal_relation
       domain: time:Temporal_Occurrence
       range: time:Temporal_Occurrence
       inverseOf: time:before

Property time:before
       subPropertyOf: time:temporal_relation
       domain: time:Temporal_Occurrence
       range: time:Temporal_Occurrence
       inverseOf: time:after

Property time:between
       subPropertyOf: time:temporal_relation
       domain: time:Temporal_Occurrence
       range: time:Pair_Of_Periods

Property time:during
       subPropertyOf: time:temporal_relation

Property time:finish
       subPropertyOf: time:temporal_relation

Property time:immediatly_after
       subPropertyOf: time:after
       domain: time:Temporal_Occurrence
       range: time:Temporal_Occurrence
       inverseOf: time:immediatly_before

Property time:immediatly_before
       subPropertyOf: time:before
       domain: time:Temporal_Occurrence
       range: time:Temporal_Occurrence
       inverseOf: time:immediatly_after
```

```
Property time:overlap
       subPropertyOf: time:temporal_relation

Property time:preceeds
       subPropertyOf: time:temporal_relation

Property time:start
       subPropertyOf: time:temporal_relation

Property time:temporal_relation
       domain: time:Temporal_Occurrence
       range: time:Temporal_Occurrence
```

A.4 On reasoning about norms

Norms are propositional attitudes, expressed in deontic terms; the directive or expectancy attitudes are added by the addressees. Depending on the kind of logic and the type of problem considered, knowledge engineers choose for different conceptualisations of these deontic qualifiers. Norms, which themselves are qualified in deontic terms, are used to assess (qualify) cases. The simplest view is that if there is no discrepancy between what is in the description of a case, and what legal norms prescribe, the case is allowed (or the law is 'silent'). However, if there is one or more discrepancies –violations– the case is a disallowed or illegal one, and law may also prescribe a sanction. Note that the sanction is related to which norm is violated. This is all fine and dandy and common sense. However, the problem is that if more than one norm is applicable to a case, the total outcome of the case may be different than a simple 'additive' set, as in the grading of answers to questions in an examination. For instance, a violated prohibition may be cancelled out by a compliant permission. This is the case if for example the propositions ('normative statements') of one of the two applicable norms subsumes the other one, and 'lex specialis' is applied to solve the conflict (or: indicate the preferences). This complicates the view on norms as consisting of a qualifier about a proposition, where the proposition (normative statement) is reified to an object. It means that in assessing legal cases we cannot do with passing a case at the object level (matching normative statements with case descriptions) and then collect the set of norms that indicate 'violation': we also have to check whether violations are overruled by 'compliances'. In other words, in legal, normative reasoning we have to jump from object level to meta-level and back. Of course, we can also establish a priori which norms are exceptions to which other norms [Winkels et al., 1999], but the point is that the deontic qualifiers do not completely hide the object level. Exceptions spoil the game. [1] This is probably a major reason why approaches to legal normative reasoning flatten out the distinction between the meta-level qualifier and object level content of the normative statement.

[1] To be sure: permissions are (always) exceptions to either an obligation or a prohibition; an obligation may be an exception to a prohibition, and vice versa. Even an obligation may be an exception to another obligation, but a prohibition that overrules another prohibition does not make sense (unless negation is used in one of the norm statements).

Bibliography

[Allen, 1984] Allen, J. (1984). Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154.

[Allen and Ferguson, 1994] Allen, J. F. and Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579.

[Benjamins et al., 2004] Benjamins, V., Contreras, C., Casanovas, P., Ayuso, M., Becue, M., Lemus, L., and Urios, C. (2004). Ontologies of professional legal knowledge as the basis for intelligent it support for judges. *Artificial Intelligence and Law*, pages 359–378.

[Bodenreider et al., 2004] Bodenreider, O., Smith, B., and Burgun, A. (2004). The ontology-epistemology divide: a case study in medical terminology. In Varzi, A. and Vieu, L., editors, *Formal ontology in Information Systems*, pages 185–198. IOS-Press, Amsterdam.

[Boer, 2006a] Boer, A. (2006a). Note on production rules and the legal knowledge interchange format. Technical report, Leibniz Center for Law, Faculty of Law, University of Amsterdam.

[Boer, 2006b] Boer, A. (2006b). Notes on the legal knowledge interchange format. Technical report, Leibniz Center for Law, Faculty of Law, University of Amsterdam.

[Boer et al., 2007] Boer, A., Gordon, T. F., van den Berg, K., Di Bello, M., Förhécz, A., and Vas, R. (2007). Specification of the legal knowledge interchange format. Deliverable 1.1, Estrella.

[Boer et al., 2005] Boer, A., van Engers, T., and Winkels, R. (2005). Mixing legal and non-legal norms. In Moens, M.-F. and Spyns, P., editors, *Jurix 2005: The Eighteenth Annual Conference.*, Legal Knowledge and Information Systems, pages 25–36, Amsterdam. IOS Press.

[Brachman et al., 1991] Brachman, R., McGuinness, D., Patel-Schneider, P., Resnick, L., and Borgida, A. (1991). Living with classic: When and how to use a kl-one-like language. In Sowa, J., editor, *Principles of semantic networks: explorations in the representation of knowledge*, pages 401–456. Morgan Kaufmann. http://www.cs.man.ac.uk/ franconi/dl/course/articles/brachman91living.ps.gz.

[Brachman, 1979] Brachman, R. J. (1979). On the epistemological status of semantic networks. In Findler, N. V., editor, *Associative Networks*, New York. Academic Press.

[Breuker, 1994] Breuker, J. (1994). Components of problem solving. In Steels, L., Schreiber, G., and Velde, W. V. D., editors, *A Future for Knowledge Acquisition: proceedings of the EKAW-94, European Knowledge Acquisition Workshop*, pages 118 – 136, Berlin. Springer Verlag.

[Breuker et al., 2006] Breuker, J., Boer, A., Hoekstra, R., and van den Berg, K. (2006). Developing content for lkif: Ontologies and frameworks for legal reasoning. In van Engers, T. M., editor, *Legal Knowledge and Information Systems. JURIX 2006: The Nineteenth Annual Conference*, volume 152 of *Frontiers in Artificial Intelligence and Applications*.

[Breuker and Hoekstra, 2004a] Breuker, J. and Hoekstra, R. (2004a). Core concepts of law: taking common-sense seriously. In Varzi, A. and Vieu, L., editors, *Proceedings of Formal Ontologies in Information Systems (FOIS-2004*, pages 210–221. IOS-Press.

[Breuker and Hoekstra, 2004b] Breuker, J. and Hoekstra, R. (2004b). Epistemology and ontology in core ontologies: FOLaw and LRI-Core, two core ontologies for law. In *Proceedings of EKAW Workshop on Core ontologies*, Http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/. Ceur.

[Breuker et al., 2004a] Breuker, J., Valente, A., and Winkels, R. (2004a). Legal ontologies in knowledge engineering and information management. *Artificial Intelligence and Law*, 12:241–277. appeared in 2006.

[Breuker et al., 2004b] Breuker, J., Valente, A., and Winkels, R. (2004b). Use and reuse of legal ontologies in knowledge engineering and information management. *Artificial Intelligence and Law*, (to appear in special issue on Legal Ontologies).

[Breuker and Van De Velde, 1994] Breuker, J. and Van De Velde, W. (1994). *CommonKADS Library for Expertise Modeling: reusable problem solving components.* IOS-Press/Ohmsha, Amsterdam/Tokyo.

[Breuker and Van de Velde, 1994] Breuker, J. and Van de Velde, W., editors (1994). Common*KADS Library for Expertise Modelling.* IOS Pres.

[casanovas et al., 2006] casanovas, P., Casellas, N., Vallbe, J.-J., maria Poblet, benjamins, R., Blazquez, M., Pena, R., and Contreras, J. (2006). Semantic web: a legal case study. In Davies, J., Studer, R., and Warren, P., editors, *Semantic Web Technologies.* Wiley.

[Castillo-Colaux and Krief, 2006] Castillo-Colaux, C. and Krief, A. (2006). Protocols for building an organic chemical ontology. In *Proceedings of the Second ELeGI Conference*, Barcelona.

[Chandrasekaran et al., 1999] Chandrasekaran, B., Josephson, J., and Benjamins, V. (1999). What are ontologies and why do we need them. *IEEE Intelligent Systems*, January/February:20–26.

[Chesnevar et al., 2006] Chesnevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G., South, M., Vreeswijk, G., , and Willmott, S. (2006). Towards an argument interchange format. *The Knowledge Engineering Review*, 21(4):293–316.

[Dahllöf, 1995] Dahllöf, M. (1995). On the semantics of propositional attitude reports.

[Davis, 1984] Davis, R. (1984). Diagnostic reasoning based on structure and behavior. *Artificial Intelligence*, 24:347–410.

[Dennett, 1987] Dennett, D. (1987). *The Intentional Stance.* MIT-Press.

[Donnelly, 2005] Donnelly, M. (2005). Relative places. *Applied Ontology*, 1(1):55–75.

[Dubislav, 1937] Dubislav, W. (1937). Zur unbegründbarkeit der forderungsätze. *Theoria*, 3:330–342.

[Dung, 1995] Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358.

[Fernández et al., 1997] Fernández, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium Series on Ontological Engineering*, pages 33–40, Stanford, USA.

[Gangemi et al., 2003] Gangemi, A., Guarino, N., Masolo, C., and Oltramari, A. (2003). Sweetening WORDNET with DOLCE. *AI Magazine*, 24:13–24.

[Gangemi et al., 2002] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. (2002). Sweetening ontologies with DOLCE. In Gomez-Perez, A. and Benjamins, V., editors, *Proceedings of the EKAW-2002*, pages 166–181. Springer.

[Gangemi et al., 2005] Gangemi, A., Sagri, M., and Tiscornia, D. (2005). A constructive framework for legal ontologies. In Benjamins, V., Casanovas, P., Breuker, J., and Gangemi, A., editors, *Law and the Semantic Web*, pages 97–124. Springer Verlag.

[Gordon, 1995] Gordon, T. (1995). *The Pleadings Game. An Artificial Intelligence Model of Procedural Justice.* Kluwer, Dordrecht.

[Gordon and Walton, 2006] Gordon, T. and Walton, D. (2006). The carneades argumentation framework: Using presumptions and exceptions to model critical questions. In Dunne, P. E. and Bench-Capon, T., editors, *Computational Models of Argument: Proceedings of COMMA 2006*, pages 195–207, Amsterdam. IOS Press.

[Gruber, 1994] Gruber, T. (1994). Towards principles for the design of ontologies used for knowledge sharing. In Guarino, N. and Poli, R., editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, pages –. Kluwer Academic Publishers. also: Technical Report KSL 93-40, Knowledge Systems Laboratory, Stanford University.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220.

[Grüninger and Fox, 1995] Grüninger, M. and Fox, M. S. (1995). Methodology for the design and evaluation of ontologies. In *IJCAI'95, Workshop on Basic Ontological Issues in Knowledge Sharing*.

[Guarino and Welty, 2002] Guarino, N. and Welty, C. (2002). Evaluating ontological decisions with OntoClean. *Communications of the ACM*, 45(2):61–65.

[Hamscher et al., 1992] Hamscher, W. C., Console, L., and de Kleer, J., editors (1992). *Readings in Model-Based Diagnosis*. Morgan Kaufmann, San Mateo, CA.

[Hansson, 1969] Hansson, B. (1969). An analysis of some deontic logics. *Nous*, 3:373–398.

[Hayes, 1985] Hayes, P. J. (1985). The second naive physics manifesto. In Hobbs, J. R. and Moore, R. C., editors, *Formal Theories of the Common Sense World*, pages 1–36. Ablex Publishing Corporation, Norwood.

[Hobbs and Pan, 2006] Hobbs, J. R. and Pan, F. (2006). Time ontology in owl. Technical report, W3C.

[Hohfeld, 1919] Hohfeld, W. (1919). *Fundamental Legal Conceptions as Applied in Legal Reasoning*. Yale University Press. Edited by W.W. Cook, fourth printing, 1966.

[Jörgenson, 1937] Jörgenson, J. (1937). Imperatives and logic. *Erkenntnis*, 7:288–296.

[Kashyap and Borgida, 2004] Kashyap, V. and Borgida, A. (2004). Representing the UMLS semantic network using OWL. In Fensel, D., Sycara, K., and Mylopoulos, J., editors, *The Semantic Web - ISCW 2003*, pages 1–16.

[Kelsen, 1991] Kelsen, H. (1991). *General Theory of Norms*. Clarendon Press, Oxford.

[Kim, 1998] Kim, J. (1998). Mind in a physical world: An essay on the mind-body problem and mental causation. In *Representation and Mind*. MIT Press.

[Lakoff, 1987] Lakoff, G. (1987). *Women, Fire and Dangerous Things*. University of Chicago Press.

[Lakoff and Núñez, 2000] Lakoff, G. and Núñez, R. (2000). *Where Mathematics Comes From*. Basic Books.

[Lame, 2006] Lame, G. (2006). Using NLP techniques to identify legal ontology components: Concepts and relations. *Artificial Intelligence and Law.* this issue.

[Loebe, 2003] Loebe, F. (2003). An analysis of roles. Technical Report 6, Research Group Ontologies in Medicine, University of Leipzig. Onto-Med Report.

[Makinson, 1999] Makinson, D. (1999). On a fundamental problem of deontic logic. In McNamara, P. and Prakken, H., editors, *Norms, Logics and Information Systems. New Studies in Deontic Logic and Computer Science*, volume 49 of *Frontiers in Artificial Intelligence and Applications*, pages 29–53. IOS Press, Amsterdam.

[Masolo et al., 2004] Masolo, C., Vieu, L., Bottazzi, E., Catenacci, C., Ferrario, R., Gangemi, A., and Guarino, N. (2004). Social roles and their descriptions. In *Proceedings of Knowledge Representation Workshop.*

[Massolo et al., 2002] Massolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L. (2002). The WonderWeb foundational ontologies: preliminary report. Technical Report Deliverable D17, version 2, ISTC-CNR (Italy).

[McCarty, 1989] McCarty, T. (1989). A language for legal discourse I. basic structures. In *Proc. of the Second International Conference on AI and Law*, pages 180–189, Vancouver. Acm.

[Minsky, 1975] Minsky, M. (1975). A framework for representing knowledge. In Winston, P. H., editor, *The psychology of Computer Vision*, pages 211–277, New York. McGraw-Hill.

[Mottta, 1999] Mottta, E. (1999). *Reusable Components for Knowledge Modelling.* FAIA-Series. IOS Pres, Amsterdam NL.

[Nuyts et al., 2005] Nuyts, J., Byloo, P., and Diepeveen, J. (2005). On deontic modality, directivity, and mood.

[Pacheco and Carmo, 2003] Pacheco, O. and Carmo, J. (2003). A role based model for the normative specification of organized collective agency and agents interaction. *Journal of Autonomous Agents and Multi-Agent Systems*, 6:145–184.

[Palmirani et al., 2007] Palmirani, M., Sartor, G., Rubino, R., Boer, A., de Maat, E., Vitali, F., and Francesconi, E. (2007). Guidelines for applying the new format. Deliverable 3.2, Estrella.

[Rahwan and Zablith, 2007] Rahwan, Iyad, C. R. and Zablith, F. (2007). On building argumentation schemes using the argument interchange format. In *Working Notes of the 7th Workshop on Computational Models of Natural Argument (CMNA 2007), Hyderabad.*

[Rahwan and Sakeer, 2006] Rahwan, I. and Sakeer, P. (2006). Towards representing and querying arguments on the semantic web. In Dunne, P. and Bench-Capon, T., editors, *Computational Models of Argument*, Amsterdam. IOS Press.

[Rubino et al., 2006] Rubino, R., Rotolo, A., and Sartor, G. (2006). An owl ontology of fundamental legal concepts. In van Engers, T. M., editor, *Legal Knowledge and Information Systems. JURIX 2006: The Nineteenth Annual Conference*, volume 152 of *Frontiers of Artificial Intelligence and Applications*. IOS Press.

[Sartor, 2006] Sartor, G. (2006). Fundamental legal concepts: A formal and teleological characterisation. Technical report, European University Institute, Florence / Cirsfid, University of Bologna.

[Schank and Abelson, 1977] Schank, R. and Abelson, R. (1977). *Scripts, Plans Goals and Understanding*. Lawrence Erlbaum, New Jersey.

[Schreiber et al., 2000] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van den Velde, W., and Wielinga, B. (2000). *Knowledge Engineering and Managament: The CommonKADS Methodology*. MIT Press.

[Searle and Vanderveken, 1985] Searle, J. and Vanderveken, D. (1985). *Foundations of illocutionary logic*. Cambridge University Press, Cambridge, England.

[Sowa, 2000] Sowa, J. F. (2000). *Knowledge Representation: Logical Philosophical, and Computational Foundations*. Brooks Cole Publishing Co, Pacific Grove, CA.

[Staab and Studer, 2004] Staab, S. and Studer, R., editors (2004). *Handbook on Ontologies*. Springer Verlag.

[Steimann, 2000] Steimann, F. (2000). On the representation of roles in object-oriented and conceptual modelling. *Data and Knowledge Engineering*, 35:83–106.

[Stevens et al., 2004] Stevens, R., Wroe, C., Lord, P., and Goble, C. (2004). Ontologies in bioinformatics. In Staab, S. and Studer, R., editors, *Handbook of Ontologies*, pages 635–657. Springer.

[Uschold and Grüninger, 1996] Uschold, M. and Grüninger, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155.

[Uschold and King, 1995] Uschold, M. and King, M. (1995). Towards a methodology for building ontologies. In *Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95*, Montreal, Canada.

[Valente, 1995] Valente, A. (1995). *Legal knowledge engineering: A modelling approach*. IOS Press, Amsterdam, The Netherlands.

[Valente and Breuker, 1995] Valente, A. and Breuker, J. (1995). ON-LINE: An architecture for modelling legal information. In *International Conference on Artificial Intelligence and Law (ICAIL-1995)*, pages 307–315.

[Valente et al., 1999a] Valente, A., Breuker, J., and Brouwer, P. (1999a). Legal modelling and automated reasoning with ON-LINE. *International Journal of Human Computer Studies*, 51:1079–1126.

[Valente et al., 1999b] Valente, A., Russ, T., MacGregor, R., and Swartout, W. (1999b). Building, using, and reusing an ontology of air campiagn planning. *IEEE Intelligent Systems*, 14:27–36.

[Walton, 1996] Walton, D. (1996). *Argumentation Schemes for Presumptive Reasoning*. Erlbaum, Mahwah, N.J.

[Winkels et al., 1999] Winkels, R., Bosscher, D., Boer, A., and Breuker, J. (1999). Generating exception structures for legal information serving. In Gordon, T., editor, *Proceedings of the Seventh International Conference on Artificial Intelligence and Law (ICAIL-99)*, pages 182–189. ACM, New York.